



## Enterprise Architect User Guide

*Enterprise Architect is an intuitive, flexible and powerful UML analysis and design tool for building robust and maintainable software. From requirements gathering, through analysis, modeling, implementation and testing to deployment and maintenance, Enterprise Architect is a fast, feature-rich, multi-user UML modeling tool, driving the long-term success of your software project.*



Copyright © 1998-2014 Sparx Systems Pty Ltd

# Enterprise Architect User Guide

## Introduction

---

*by Geoffrey Sparks*

*Enterprise Architect is a complete UML-based solution for analysing, designing, managing, sharing and building software systems.*

# Enterprise Architect User Guide

© 1998-2014 Sparx Systems Pty Ltd

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: November 2014

## **Publisher**

*Sparx Systems*

## **Managing Editor**

*Geoffrey Sparks*

## **Technical Editors**

*Geoffrey Sparks*

*Dermot O'Bryan*

*Simon McNeilly*

*Neil Capey*

*John Redfern*

*Vimal Kumar*

*Howard Britten*

*Brad Maxwell*

*Steve Meagher*

## **Special thanks to:**

*All the people who have contributed suggestions, examples, bug reports and assistance in the development of Enterprise Architect. The task of developing and maintaining this tool has been greatly enhanced by their contribution.*

# Table of Contents

Foreword .....	1
<b>Part I Introduction .....</b>	<b>3</b>
Overview .....	5
What You Can Do .....	7
Key Benefits .....	9
Key Features .....	13
Enterprise Architect Editions .....	18
The Trial Version .....	18
Editions Available .....	20
The Read-only 'Lite' Edition .....	23
Formal Statements .....	25
Copyright Notice .....	25
End User License Agreement .....	26
Trademarks .....	29
Acknowledgements .....	32
Order Enterprise Architect .....	33
Installation .....	34
Register a Full License .....	36
Help and Support .....	38
Available Helpfile Formats .....	39
Support .....	40
<b>Part II Getting Started .....</b>	<b>42</b>
Basics .....	43
Common Tasks .....	45
A Quick Start Tutorial .....	47
Create a Project .....	47
Add a View to your Model .....	49
Add New Packages .....	50
Create a New Diagram .....	50
Save Changes .....	51
Create New Elements on a Diagram .....	52
Add Connectors .....	55
Modify Properties .....	56
Move Objects Around .....	57
Move Objects Within a Package.....	57
Move Objects Between Packages.....	58
Move Elements in a Diagram.....	59
Move Elements Between Diagrams.....	59
Connect to a Different Element on a Diagram.....	61
Delete a Connector .....	62
Delete an Element .....	64
Remove Elements From a Diagram.....	64
Delete Elements From Your Model.....	65
User Interface Guide .....	67
Start Page .....	68
Remove Recent Projects.....	73
Learning Center .....	74



Create Learning Center Pages.....	74
<b>Main Menu .....</b>	<b>77</b>
File Menu.....	79
Edit Menu.....	80
Paste Elements.....	82
View Menu.....	83
View Submenus .....	84
Project Menu.....	87
Documentation Submenu .....	89
Model Validation Submenu.....	89
Model Import/Export Submenu.....	90
Version Control Submenu.....	91
Security Submenu.....	92
Diagram Menu.....	93
Diagram Advanced Menu.....	95
Element Menu.....	97
Appearance Submenu.....	99
Advanced Submenu.....	100
Inline Features Submenu.....	101
Source Code Engineering Submenu.....	103
Tools Menu.....	104
Source Code Engineering Submenu.....	106
Database Engineering Submenu.....	107
Web Services Submenu.....	108
XML Schema Submenu.....	108
Model Transformation (MDA) Submenu.....	109
Spelling Tools.....	109
Data Management Submenu.....	110
Manage .EAP File Submenu .....	111
Analyzer Menu.....	112
Debug Menu.....	114
Record Menu.....	115
Extensions Menu.....	117
Settings Menu.....	119
Project Types Submenu.....	120
Window Menu.....	121
Help Menu.....	123
<b>Standard Windows .....</b>	<b>125</b>
Dock Windows.....	128
Autohide Windows.....	132
Code Editors.....	133
<b>Standard Toolbars .....</b>	<b>134</b>
Default Tools Toolbar .....	135
Project Toolbar .....	136
Diagram Toolbar .....	138
UML Elements Toolbar.....	140
Current Element Toolbar.....	141
Current Connector Toolbar.....	143
Code Generation Toolbar .....	144
Debug Toolbar.....	146
Build Toolbar.....	147
Record & Analyze Toolbar.....	148
Workspace Layouts Toolbar.....	149
Status Bar .....	150
<b>Customization .....</b>	<b>151</b>
The Customize Dialog.....	152
Customize Commands.....	153
Customize Toolbars .....	156

Add Custom Tools.....	157
Open External Tools.....	158
Pass Parameters to Applications.....	160
Customize Keyboard Shortcuts.....	161
Customize Menu.....	162
Customize Options.....	163
Manage Workspace Layout.....	163
Customize Command Sets.....	165
User Interface Customization.....	167
Visual Styles .....	168
<b>Other Windows .....</b>	<b>168</b>
The System Output Window .....	169
The Web Browser.....	170
<b>Keyboard Shortcuts .....</b>	<b>171</b>
Keyboard-Mouse Shortcuts.....	180
<b>Typical Project Roles .....</b>	<b>182</b>
Summary of Typical Tasks .....	183
Business Analysts .....	185
Software Architects .....	186
Software Engineers .....	187
Developers .....	188
Project Managers .....	190
Testers .....	191
Implementation Managers .....	192
Technology Developers .....	193
Database Developers .....	194

## Part III Projects and Teams

197

<b>Introduction .....</b>	<b>199</b>
<b>Projects Defined .....</b>	<b>200</b>
<b>Open a Project .....</b>	<b>202</b>
<b>Project Shortcuts .....</b>	<b>204</b>
Create Project Shortcut.....	206
Capture Current Work Environment.....	207
Encrypt Repository Password.....	209
<b>File Based Repositories .....</b>	<b>210</b>
Copy a Base Project .....	211
Copy Existing Project .....	212
<b>Server Based Repositories .....</b>	<b>214</b>
<b>Create a Project in a SQL Server Database .....</b>	<b>215</b>
Create a SQL Server Repository.....	217
Connect to a SQL Server Data Repository .....	218
<b>Create a Project in a MySQL Database .....</b>	<b>220</b>
Create a MySQL Repository.....	222
Set up a MySQL ODBC Driver.....	223
Connect to a MySQL Data Repository.....	225
<b>Create a Project in a PostgreSQL Database .....</b>	<b>227</b>
Create a PostgreSQL Repository.....	229
Set up a PostgreSQL ODBC Driver .....	230
Connect to a PostgreSQL Data Repository.....	233
<b>Create a Project in a Sybase ASA Database .....</b>	<b>234</b>
Create an Adaptive Server Anywhere Repository.....	237
Set up an ASA ODBC Driver.....	237
Connect to an ASA Data Repository.....	239
<b>Create a Project in an Access 2007 Database .....</b>	<b>241</b>
<b>Create a Project in a Progress OpenEdge Database .....</b>	<b>243</b>
Create a Progress OpenEdge Repository.....	245

Set up a Progress OpenEdge ODBC Driver .....	247
Connect to a Progress OpenEdge Data Repository .....	248
<b>Create a Project in an Oracle Database .....</b>	<b>250</b>
Create an Oracle Data Repository .....	253
Set up an Oracle ODBC Driver .....	254
Connect to an Oracle Data Repository (ODBC) .....	255
Connect to an Oracle Data Repository (OLE DB) .....	257
<b>The WAN Optimizer .....</b>	<b>259</b>
<b>Connecting to Projects Via the Cloud .....</b>	<b>262</b>
<b>Setting up the Cloud Server .....</b>	<b>262</b>
<b>Connect to a Project Via the Cloud .....</b>	<b>266</b>
<b>Additional Functionality Using the Cloud .....</b>	<b>267</b>
<b>OSLC Requirements Management .....</b>	<b>268</b>
Service Provider and Service Provider Resource .....	269
Resource Shape .....	271
Query Capability .....	273
WHERE Parameter .....	275
SELECT Parameter .....	276
Combine WHERE and SELECT Parameters .....	277
PROPERTIES Parameter .....	278
PREFIX Parameter .....	279
Creation Factory .....	280
<b>Reuseable Asset Service .....</b>	<b>282</b>
<b>Connect to the Asset Service .....</b>	<b>283</b>
<b>Browse Assets .....</b>	<b>284</b>
Asset Properties .....	286
Package Contents .....	286
Package Dependencies .....	288
Package Technologies .....	289
Storage Files .....	290
Compare an Asset to the Model .....	291
<b>Import an Asset into the Model .....</b>	<b>291</b>
<b>Set Up the Asset Service .....</b>	<b>293</b>
Register New Assets .....	298
Check Package Dependencies .....	302
Update an Asset .....	304
<b>Team Development .....</b>	<b>306</b>
<b>Project Sharing .....</b>	<b>307</b>
Share Enterprise Architect Projects .....	308
Refresh View of Shared Project .....	308
Share Projects on Network Drive .....	309
Distributed Development .....	309
Replication .....	310
Design Masters .....	311
Create Replicas .....	312
Synchronize Replicas .....	313
Remove Replication .....	313
Upgrade Replicas .....	315
Resolve Conflicts .....	315
<b>User Security .....</b>	<b>316</b>
Enable/Disable Security .....	318
Set Security Policy .....	319
Maintain Groups .....	320
Set Group Permissions .....	321
Maintain Users .....	323
Import User IDs From Active Directory .....	324
Assign User To Groups .....	326

Set Up Single Permissions.....	327
View All User Permissions.....	328
List of Available Permissions.....	329
View and Manage Locks.....	332
Password Encryption.....	333
Change Password.....	334
Lock Model Elements.....	336
Lock Objects Under User/Group Locking.....	336
Lock Packages Under User/Group Locking.....	337
Lock Objects Under Require User Lock to Edit.....	339
Locked Element Indicators.....	340
Identify Who Has Locked An Object.....	342
Manage Your Own Locks.....	342
<b>Team Review Tools .....</b>	<b>343</b>
Work on Team Review Items.....	346
Add a New Category.....	350
Add a New Topic.....	351
Add a New Document.....	353
Comment on a Document.....	355
Protection Against Editing.....	357
Edit an Item.....	358
Add Object Links.....	359
Team Review Resources.....	361
Search Team Review .....	362
Team Review Options.....	362
Team Review Connections .....	363
<b>Element Discussions .....</b>	<b>365</b>
<b>Workflow Scripts .....</b>	<b>367</b>
Workflow Script Functions.....	368
Functions - Validate and Control User Input.....	369
Functions - Create a Search With User Tasks .....	370
Filled Workflow Data Structures.....	371
Workflow Data Structures You Fill.....	372
Functions You Call.....	373
<b>Sharing Reference Data .....</b>	<b>374</b>
Link Reference Data to a Shared Repository.....	375
Export Reference Data.....	376
Import Reference Data.....	380
<b>Change Management .....</b>	<b>382</b>
<b>Version Control .....</b>	<b>383</b>
Introduction.....	383
Version Control Usage.....	385
Version Control of Model Data.....	386
Version Control and Reference Data.....	387
Version Control Basics.....	387
Add Connectors To Locked Elements.....	388
Applying Version Control in a Team Environment.....	389
Version Control Nested Packages.....	390
Project Browser Indicators.....	391
Offline Version Control.....	392
Version Control Branching.....	394
Version Control Product Setup.....	394
System Requirements.....	395
Create a Subversion Environment.....	397
Create a new Repository Sub-tree.....	398
Create a Local Working Copy .....	399
Verify the SVN Workspace.....	400
Subversion Under Wine-Crossover.....	401

Preparing a Subversion Environment Under Wine.....	402
TortoiseSVN.....	403
Create a CVS Environment.....	404
Prepare a CVS Local Workspace.....	405
Verify the CVS Workspace.....	406
TortoiseCVS.....	407
Create a TFS Environment.....	408
TFS Workspaces.....	409
TFS Exclusive Check Outs.....	411
Create an SCC Environment.....	411
Upgrade at Enterprise Architect Version 4.5, Under SCC Version Control .....	412
Version Control Setup.....	413
Re-use an Existing Configuration.....	414
Version Control Settings.....	415
SCC Settings.....	417
CVS Settings.....	419
SVN Settings.....	420
TFS Settings.....	421
Use Version Control.....	422
Configure Controlled Package.....	424
Apply Version Control To Branches.....	426
Package Version Control Options.....	427
Check Out a Package.....	431
Undo Check Out of a Package.....	432
Check In a Package.....	432
Check Out a Model Branch.....	433
Check In a Model Branch.....	434
Update to the Latest Revision of Selected Package.....	435
Update to the Latest Revision of All Packages.....	435
Include Other Users' Packages.....	436
Export Controlled Model Branch.....	437
Import Controlled Model Branch.....	438
Manually Locating Model Branch Files.....	439
Review Package History.....	440
Review Package History - SCC Client.....	441
Retrieve Prior Revision - SCC Client.....	442
Validate Package Configurations.....	443
Resynchronize the Status of Version Controlled Packages.....	444
<b>Tracking Changes .....</b>	<b>445</b>
Auditing.....	446
Auditing Quickstart.....	447
Auditing Settings.....	448
The Audit View .....	451
Audit View Controls.....	453
Audit History Tab.....	455
Auditing Performance Issues.....	456
Audit View Performance Issues.....	457
Package Baselines.....	457
Baselines .....	459
Manage Baselines.....	461
Create Baselines.....	462
The Compare Utility (Diff).....	463
Compare Options .....	464
Check Visual Changes to Diagrams.....	466
Example Comparison.....	468
Baseline Comparison Tab Options.....	470
<b>Model Transfer .....</b>	<b>473</b>

XML Import and Export.....	473
Export to XML.....	475
Publish Model Package.....	476
Import from XML.....	478
Import EMX/UML2 Files.....	480
Import a Rhapsody Model.....	482
Limitations of XML.....	482
The UML DTD.....	483
Controlled Packages.....	484
Controlled Package Menu.....	485
Configure Packages.....	488
Remove Package from Control.....	490
Save a Package.....	491
Load a Package.....	492
Batch XML Export.....	492
Batch XML Import.....	493
Manual Version Control with XML.....	494
Report Deletion of Cross Package References.....	496
CSV Import and Export.....	497
CSV Specifications.....	498
Using Preserve Hierarchy.....	500
CSV Export.....	501
CSV Import.....	503
Perform a Project Data Transfer.....	504
Copy Packages Between Projects.....	506
<b>Compare Projects .....</b>	<b>507</b>
<b>Project Management .....</b>	<b>509</b>
<b>The Project Management Window .....</b>	<b>510</b>
<b>Project Resources .....</b>	<b>512</b>
Resource Allocation.....	512
Assign Multiple Resources.....	515
Effort Management.....	518
Risk Management.....	519
Metrics.....	520
Effort Types.....	522
Metric Types.....	523
Risk Types.....	524
<b>The Project Status View .....</b>	<b>525</b>
Project Tasks.....	526
Add, Modify and Delete Tasks.....	527
Project Issues.....	528
Add, Delete and Modify Issues.....	529
Report From Project Issues Dialog.....	531
Report From Project Issues Tab.....	531
Report Output Sample.....	532
<b>Project Glossary .....</b>	<b>533</b>
Project Glossary View .....	534
The Glossary Dialog.....	535
Generate a Report.....	536
Glossary Report Output Sample.....	537
<b>Project Task Allocation .....</b>	<b>538</b>
Resource View .....	538
Element View .....	542
Report View .....	546
<b>Spell Checking .....</b>	<b>549</b>
Select Spell Checker Options.....	550
Use Languages Other Than English.....	552
Using the Spell Checker .....	553

Correcting Words.....	554
<b>Personal Tasks .....</b>	<b>554</b>
Review Allocated Work.....	555
Monitor Your Tasks.....	559
Monitor Workflow .....	560
Working Sets.....	561
<b>Model Mail .....</b>	<b>565</b>
Create a Message.....	570
<b>Project Calendar .....</b>	<b>571</b>
Calendar.....	574
Configure Event Subtypes.....	577
Allocated Resources.....	578
Project Tasks.....	581
<b>Use Case Estimation .....</b>	<b>584</b>
Technical Complexity Factors .....	585
Environment Complexity Factors.....	586
Default Hours .....	588
Estimating Project Size.....	588
<b>Update Package Status .....</b>	<b>590</b>
<b>Manage Bookmarks .....</b>	<b>591</b>
<b>Monitor Change Events .....</b>	<b>593</b>
<b>The Gantt View .....</b>	<b>594</b>
<b>Project Maintenance .....</b>	<b>596</b>
Check Project Data Integrity .....	597
Reset Table Auto Increment or Identity Columns .....	598
Upgrade a Project .....	599
Upgrade Replicas.....	600
Run SQL Patches .....	601
Rename a Project .....	601
Compact a Project .....	601
Repair a Project .....	602
<b>Local Options .....</b>	<b>604</b>
General Options .....	605
Diagram Options .....	608
Diagram Theme Options .....	611
Diagram Gradients and Backgrounds .....	613
Standard Colors .....	616
Create Custom Colors.....	619
Get/Set Custom Project Colors.....	621
Diagram Appearance Options.....	622
Set Default Fonts .....	624
Diagram Behavior Options.....	625
Sequence Diagram Options.....	629
Object Display Options .....	631
Connector Display Options .....	635
Communication Message Colors .....	638
XML Specifications .....	639

## **Part IV Navigate, Search and Trace 644**

<b>Navigate: Exploring Your Model .....</b>	<b>646</b>
<b>The Project Browser .....</b>	<b>646</b>
Project Browser Context Menus.....	648
Model (Root Node) Context Menu .....	648
Package Options in the Project Browser.....	651
Advanced Sub-Menu.....	654
Copy/Paste Submenu.....	654
Contents Sub-Menu.....	655

Documentation Sub-Menu.....	656
Code Engineering Options.....	657
Import/Export Sub-Menu.....	659
Element Options in the Project Browser .....	660
Add Sub Menu.....	662
Element Copy/Paste Submenu.....	663
Element Code Engineering Menu.....	664
Diagram Menu - Project Browser.....	665
Operation Menu - Project Browser.....	667
Project Browser Toolbar.....	669
Project Browser Icon Overlays.....	670
Order Package Contents.....	672
Set Default Behavior.....	672
<b>Package Browser .....</b>	<b>673</b>
List Header.....	677
Package Browser Options.....	679
<b>Diagram List .....</b>	<b>684</b>
<b>Model Views .....</b>	<b>686</b>
Model Views Toolbar .....	688
Model Views Context Menus.....	690
Model Views Operations.....	693
Diagram Slide Show .....	695
<b>The Pan &amp; Zoom Window .....</b>	<b>698</b>
<b>Search: Finding Information .....</b>	<b>700</b>
<b>Model Search .....</b>	<b>700</b>
Model Search Context Menu.....	703
Model Search Toolbar .....	705
Pre-defined Searches.....	706
Customizing the Search View .....	708
Create & Modify Searches.....	709
Create Search Definitions.....	711
Advanced Search Options .....	715
Add Filters .....	716
Fields and Conditions.....	718
<b>Diagram Filters .....</b>	<b>718</b>
Work With Diagram Filters .....	720
<b>Trace: Tracking Dependencies .....</b>	<b>723</b>
<b>Traceability Tools .....</b>	<b>724</b>
<b>The Traceability Window .....</b>	<b>725</b>
<b>Relationship Matrix .....</b>	<b>727</b>
Set Element Type.....	729
Set Connector Type and Direction.....	730
Set Source and Target Package.....	731
Matrix Overlays.....	732
Create a Matrix Overlay.....	733
Apply an Overlay to a Cell.....	734
Matrix Profiles.....	735
Creating and Deleting Relationships.....	737
Relationship Matrix Options.....	738
Review Source and Target Elements.....	741
<b>The Relationships Window .....</b>	<b>742</b>
<b>Example Traceability Diagram .....</b>	<b>743</b>
<b>Gap Analysis Matrix .....</b>	<b>745</b>
Edit Gap Notes.....	748

## Part V Modeling Basics

750

Modeling .....	752
----------------	-----



<b>Models .....</b>	<b>753</b>
<b>Model Wizard .....</b>	<b>753</b>
<b>Model Templates .....</b>	<b>755</b>
Business Process Model Template .....	758
Requirements Model Template .....	758
Use Case Model Template .....	759
Domain Model Template .....	760
Class Model Template .....	761
Database Model Template .....	762
Component Model Template .....	763
Deployment Model Template .....	764
Testing Model Template .....	765
Maintenance Model Template .....	766
Project Model Template .....	767
<b>Views .....</b>	<b>769</b>
<b>Add Views .....</b>	<b>770</b>
<b>Rename Views .....</b>	<b>771</b>
<b>Delete Views .....</b>	<b>771</b>
<b>Packages .....</b>	<b>772</b>
<b>Add a Package .....</b>	<b>772</b>
<b>Open Package in the Project Browser .....</b>	<b>774</b>
<b>Rename a Package .....</b>	<b>774</b>
<b>Copy a Package .....</b>	<b>775</b>
<b>Drag a Package Onto a Diagram .....</b>	<b>776</b>
Show or Hide Package Contents .....	777
<b>Delete a Package .....</b>	<b>777</b>
<b>Diagrams .....</b>	<b>778</b>
<b>Diagram Context Menu .....</b>	<b>778</b>
Insert Elements and Connectors .....	782
Advanced Diagram Submenu .....	783
<b>Print Preview .....</b>	<b>783</b>
<b>Diagram View .....</b>	<b>784</b>
Format Toolbar .....	785
Context Filter a Diagram .....	789
<b>Diagram Tabs .....</b>	<b>790</b>
Switch Between Views .....	791
<b>Diagram Toolbox .....</b>	<b>792</b>
Toolbox Appearance Options .....	797
Toolbox Shortcut Menu .....	799
Common Page .....	800
Use Case Toolbox .....	801
Class Toolbox .....	803
Object Toolbox .....	804
Composite Toolbox .....	804
Communication Toolbox .....	805
Interaction Toolbox .....	806
Timing Toolbox .....	806
State Toolbox .....	807
Activity Toolbox .....	808
Component Toolbox .....	809
Deployment Toolbox .....	809
Profile Toolbox .....	810
Metamodel Toolbox .....	812
Analysis Toolbox .....	812
Custom Toolbox .....	813
Requirements Toolbox .....	814
Maintenance Toolbox .....	815

User Interface Toolbox.....	816
WSDL Toolbox.....	817
XML Schema Toolbox.....	818
Data Modeling Toolbox.....	818
Test Domain Toolbox.....	819
<b>Diagram Tasks .....</b>	<b>820</b>
Add New Diagrams.....	822
Set Diagram Properties.....	823
General Tab.....	824
Configure Diagram Display.....	825
Define Element Characteristics.....	828
Features Tab.....	830
Visible Class Members.....	831
Connectors Tab.....	831
Paste from Project Browser.....	833
Paste Packages.....	835
Copy And Paste Diagram Element.....	836
Move/Copy Elements To & From Floating Diagrams.....	837
Place Related Elements on Diagram.....	838
Delete Diagram.....	839
Rename Diagram.....	839
Change Diagram Type.....	840
Diagram Navigation Hotkeys.....	841
Copy Image to Disk.....	841
Copy Image to Clipboard.....	842
Copy (Duplicate) Diagram.....	842
Z Order Elements.....	844
Set the Default Diagram.....	844
Open Package From Diagram.....	845
Feature Visibility.....	845
Insert Diagram Properties Note.....	848
Create Legends.....	849
Legend Properties.....	849
Style Options.....	850
Autosize Elements.....	851
Swimlanes.....	852
Swimlane Details.....	853
Swimlanes Matrix.....	854
Kanban Facilities.....	857
Using the Image Manager.....	860
Create Custom Diagram Background.....	863
Import Image Library.....	864
Show Realized Interfaces of Class.....	865
Manage Object Labels.....	866
Pan and Zoom a Diagram.....	868
Move Elements In Diagram Sections.....	869
View Last and Next Diagram.....	869
Set Up Diagram Page.....	870
Scale Image to Page Size.....	871
Lock Diagram.....	872
Undo Last Action.....	873
Redo Last Action.....	873
<b>Layout Diagrams .....</b>	<b>874</b>
Circular/Elliptical Layout.....	875
Box Layout.....	879
Per Page Layout.....	881
Digraph Layout.....	883
Spring Layout.....	883

Neaten Layout.....	885
Converge/Diverge Layout.....	886
Fan Relations Layout.....	888
Auto Route Layout.....	890
Lay Out a Diagram Automatically.....	891
Chain Diagram Layouts.....	894
<b>The Quick Linker .....</b>	<b>896</b>
Create New Elements.....	897
Create Connectors.....	898
<b>Elements .....</b>	<b>900</b>
<b>Element Tasks .....</b>	<b>900</b>
Create Elements.....	902
Add Elements Directly To Packages.....	903
Set Auto Naming and Auto Counters.....	904
Apply Auto Naming to Existing Elements.....	906
Set Element Parent.....	908
Show Element Use.....	910
Move Elements Within Diagrams.....	911
Copy Elements Between Diagrams.....	912
Move Elements Between Packages.....	913
Copy Elements Between Packages.....	915
Set Up Cross References.....	916
Change Element Type.....	918
Align Elements.....	918
Resize Elements.....	919
Delete Elements from Diagram and Model.....	921
Customize Visibility of Elements.....	922
Create Notes and Text.....	923
Link Note to Internal Documentation.....	924
Set an Element's Default Appearance.....	927
Set Element Templates Package.....	929
Highlight Context Element.....	930
Make Linked Element a Local Copy.....	931
Copy Features Between Elements.....	932
Move Features Between Elements.....	933
Insert Related Elements.....	933
Manage Structural Elements.....	935
Composite Elements.....	936
Show Embedded Elements In Composite.....	938
<b>Operations on Elements .....</b>	<b>939</b>
Finding Elements.....	944
Advanced Operations on Elements.....	944
Changing Element Appearance.....	948
Set Element Font.....	949
Operations on Multiple Elements.....	951
Position Elements.....	953
<b>Visual Representation .....</b>	<b>953</b>
Element Icons.....	954
Compartments.....	955
<b>Element Property Displays .....</b>	<b>956</b>
Properties Dialog.....	956
General Settings.....	958
Details.....	959
Advanced Properties.....	961
Templates.....	961
Requirements.....	961
External Requirements.....	963
Constraints.....	963

Links .....	964
Scenarios .....	965
Structured Specification Tab .....	967
Set Up Scenario Specification .....	968
Structured Specification Toolbar .....	970
Structured Specification Item Context Menu .....	972
Structured Specification Selected Text Context Menu .....	974
Structured Specification Entry Points Context Menu .....	976
Structured Specification Floating Toolbar .....	976
Generate Diagrams .....	976
Generated Activity Diagram .....	978
Generated RuleFlow Diagram .....	980
Generated State Machine Diagram .....	980
Generated Sequence Diagram .....	982
Generated Robustness Diagram .....	984
Generate Scenario From Activity Diagram .....	985
Generate Test Cases .....	986
Context References Tab .....	987
Scenario Constraints Tab .....	988
Associated Files .....	988
The Element Browser .....	989
Properties Window .....	992
Element Scenarios & Requirements .....	992
Select <Item> Dialog .....	994
Select Property Dialog .....	996
Set Feature Dialog .....	998
<b>Attributes .....</b>	<b>999</b>
General Properties of Attributes .....	1001
Attributes Dialog - Detail .....	1003
Attributes Dialog - Constraints .....	1004
Attribute Tagged Values .....	1005
Create Properties .....	1006
Display Inherited Attributes .....	1007
Create Object From Attribute .....	1008
<b>Classifiers and Instances .....</b>	<b>1009</b>
Classify an Object .....	1010
Add Property Slots to Instance .....	1011
Classify Object Via Classifier Associations .....	1012
<b>Behavior .....</b>	<b>1014</b>
Operations .....	1014
General Properties of Operations .....	1015
Operation Behavior .....	1018
Initial Code .....	1019
Advanced Properties of Operations .....	1020
Operation Tagged Values .....	1022
Operation Constraints .....	1023
Override Parent Operations .....	1023
Display Inherited Operations .....	1024
Interactions and Activities .....	1026
Behavior Calls .....	1026
Associate with Different Behaviors .....	1027
Synchronize Arguments .....	1027
Behavior Call Arguments .....	1027
Behavior Parameters .....	1028
Define Parameters .....	1028
Parameter Tagged Values .....	1030
Operation Parameters by Reference .....	1031
<b>In-place Editing Options .....</b>	<b>1032</b>

In-place Editing Tasks.....	1032
Edit Element Item Name.....	1034
Edit Feature Stereotype.....	1035
Edit Feature Scope.....	1036
Edit Attribute Keyw ord.....	1037
Edit Operation Parameter Keyw ord.....	1038
Edit Parameter Kind.....	1039
Edit Custom Compartment.....	1039
Insert New Feature.....	1040
Insert Operation Parameter.....	1041
Insert Maintenance Feature.....	1042
Insert Testing Features.....	1043
<b>Linked Documents .....</b>	<b>1044</b>
Create Document Artifact.....	1046
Create Linked Document on an Element.....	1047
Edit Linked Documents.....	1048
File Control.....	1050
Editor Tool Display Options.....	1052
Styles, Special Text & Table of Contents.....	1054
The Normal.rtf Style Template.....	1058
Scroll, Search and Select Text.....	1059
Format Text.....	1062
Format Paragraphs.....	1064
Set Tabs .....	1067
Define Document Sections.....	1069
Insert Headers, Footers, Footnotes and Endnotes.....	1071
Create Tables.....	1075
Apply User-Defined Section Numbering.....	1081
Insert Reference Links.....	1083
Insert Images, Objects and Frames.....	1085
Checking Text.....	1089
Track Changes.....	1091
Protect Document Contents.....	1093
Print Report Documents.....	1094
Hyperlink From Linked Document.....	1095
Create Element From Document.....	1096
Replace or Delete Documents.....	1097
Create Linked Document Templates.....	1098
Edit Linked Document Templates.....	1100
<b>Connectors .....</b>	<b>1102</b>
<b>Connector Management Options .....</b>	<b>1102</b>
Manage Connector Properties.....	1103
Connector Type-Specific Options.....	1103
Advanced Connector Management Options.....	1105
Connector Style Options.....	1106
Connector Appearance Options.....	1107
<b>Connector Tasks .....</b>	<b>1108</b>
Connect Elements.....	1109
Connect to Element Feature.....	1110
Add a Note to a Connector.....	1111
Arrange Connectors.....	1112
Change Connector Type.....	1113
Change the Source or Target Element.....	1113
Connector Styles.....	1114
Create Connector in Project Brow ser.....	1118
Relationship Visibility.....	1119
Delete Connectors.....	1119
Generalization Sets.....	1120

Hide/Show Connectors.....	1121
Hide/Show Labels.....	1122
Connector In-place Editing Options .....	1123
Reverse Connector.....	1123
Set Association Specializations.....	1123
Change Sequence Message Scope.....	1124
Show Uses Arrow Head.....	1125
Tree Style Hierarchy .....	1125
<b>Connector Properties .....</b>	<b>1126</b>
Connector Constraints .....	1128
Binding.....	1129
Source Role.....	1130
Target Role.....	1132
Connector Tagged Values .....	1132
Connector Advanced.....	1133
<b>Tagged Values .....</b>	<b>1134</b>
Quick Start - Add Tagged Value To Elements .....	1136
Assign a Tagged Value to an Item .....	1137
Assign Notes to a Tagged Value .....	1138
Modify Tagged Values .....	1138
Show Duplicate Tags .....	1139
Advanced Tag Management .....	1140
<b>Notes .....</b>	<b>1142</b>
Notes Toolbar .....	1143
<b>Reference Data .....</b>	<b>1146</b>
<b>UML Types .....</b>	<b>1146</b>
Stereotype Settings.....	1147
Shape Editor.....	1149
Tagged Value Types.....	1150
Cardinality.....	1151
<b>People .....</b>	<b>1152</b>
Project Authors .....	1153
Select Users .....	1154
Project Roles .....	1155
Project Resources.....	1156
Project Clients.....	1157
<b>General Types .....</b>	<b>1158</b>
Status Types.....	1159
Constraint Types.....	1160
Constraint Status Types.....	1161
Difficulty Types.....	1163
Priority Types .....	1164
Test Status Types.....	1165
Requirement Types .....	1166
Scenario Types.....	1167
<b>Metrics and Estimation .....</b>	<b>1169</b>
<b>Maintenance .....</b>	<b>1169</b>
Problem Types.....	1169
Testing Types.....	1170
<b>Data Types .....</b>	<b>1171</b>
<b>Resources .....</b>	<b>1173</b>
Favorites .....	1176

## Part VI Standard UML Models 1179

UML Diagrams .....	1181
UML Structural Models .....	1182

<b>Package Diagram .....</b>	<b>1182</b>
Example Package Diagram.....	1183
<b>Class Diagram .....</b>	<b>1184</b>
Example Class Diagram.....	1185
<b>Object Diagram .....</b>	<b>1186</b>
Example Object Diagram.....	1187
<b>Composite Structure Diagram .....</b>	<b>1188</b>
Example Composite Structure Diagram.....	1189
Properties.....	1189
<b>Deployment Diagram .....</b>	<b>1191</b>
Example Deployment Diagram.....	1193
<b>Component Diagram .....</b>	<b>1194</b>
Example Component Diagram.....	1195
<b>Profile Diagram .....</b>	<b>1196</b>
Example Profile Diagram.....	1197
<b>UML Behavioral Models .....</b>	<b>1198</b>
<b>Activity Diagram .....</b>	<b>1199</b>
Example Activity Diagram.....	1200
<b>Use Case Diagram .....</b>	<b>1201</b>
Example Use Case Diagram.....	1203
<b>State Machines .....</b>	<b>1203</b>
Example State Machine.....	1205
Pseudo-States.....	1208
Regions .....	1209
Create a Connection Point Reference.....	1210
<b>State Machine Table .....</b>	<b>1212</b>
State Machine Table Options.....	1213
State Machine Table Operations .....	1215
Change State Machine Table Position.....	1216
Change State Machine Table Size.....	1216
Insert New State.....	1217
Insert Trigger.....	1218
Insert/Change Transition.....	1218
Reposition State or Trigger Cells.....	1219
Add Legend.....	1219
Find Cell in State Machine Diagram.....	1220
State Machine Table Conventions.....	1221
Export State Table To CSV File.....	1222
Example State-Trigger Table.....	1222
Example State-Next State Table.....	1223
State Machine Table Simulation.....	1224
<b>Timing Diagram .....</b>	<b>1225</b>
Example Timing Diagram.....	1227
Create a Timing Diagram.....	1227
Set a Time Range.....	1228
Edit a Timing Diagram.....	1228
Add and Edit State Lifeline.....	1229
Add States to a State Lifeline .....	1230
Edit States in a State Lifeline.....	1231
Delete States in a State Lifeline.....	1231
Edit Transitions In State Lifeline.....	1232
Add and Move Transitions.....	1233
Add and Edit Value Lifeline.....	1234
Add States In Value Lifeline.....	1235
Edit Transitions In Value Lifeline.....	1235
Configure Timeline - States.....	1237
Numeric Range Generator .....	1239
Configure Timeline - Transitions.....	1239

Time Intervals .....	1241
Create Time Intervals .....	1241
Compress Time Intervals .....	1244
Select Time Intervals .....	1245
Time Interval Operations .....	1246
<b>Sequence Diagram .....</b>	<b>1249</b>
Example Sequence Diagram .....	1251
Denote Lifecycle of an Element .....	1251
Layout of Sequence Diagrams .....	1252
Sequence Elements .....	1253
Sequence Diagrams and Version Control .....	1254
Sequence Element Activation .....	1255
Lifeline Activation Levels .....	1256
Sequence Message Label Visibility .....	1258
Change the Top Margin .....	1258
Inline Sequence Elements .....	1258
<b>Communication Diagram .....</b>	<b>1259</b>
Example Communication Diagram .....	1260
Communication Diagrams in Color .....	1261
<b>Interaction Overview Diagram .....</b>	<b>1262</b>
Example Interaction Overview Diagram .....	1263
<b>UML Elements .....</b>	<b>1265</b>
<b>Behavioral Diagram Elements .....</b>	<b>1265</b>
Action .....	1266
Action Types .....	1268
Variable Actions .....	1273
Local Pre/Post Conditions .....	1274
Class Operations in Diagrams .....	1275
Action Pin .....	1277
Assign Action Pins .....	1278
Activity .....	1279
Activity Notation .....	1281
Activity Parameter Nodes .....	1281
Activity Partition .....	1283
Actor .....	1284
Central Buffer Node .....	1285
Choice .....	1286
Combined Fragment .....	1287
Create a Combined Fragment .....	1290
Interaction Operators .....	1290
Datastore .....	1293
Decision .....	1294
Diagram Frame .....	1296
Diagram Gate .....	1297
Endpoint .....	1298
Entry Point .....	1300
Exception .....	1300
Expansion Node .....	1301
Expansion Region .....	1301
Exit Point .....	1304
Final .....	1305
Flow Final .....	1306
Fork/Join .....	1307
Fork .....	1309
Join .....	1310
History .....	1311
Initial .....	1313
Interaction .....	1314



Interruptible Activity Region.....	1316
Interaction Occurrence.....	1317
Junction.....	1319
Lifeline.....	1321
Merge.....	1322
Message Endpoint.....	1322
Message Label.....	1323
Note.....	1324
Object Node.....	1325
Partition.....	1325
Receive.....	1327
Region.....	1328
Send.....	1328
State.....	1329
Composite State.....	1330
State/Continuation.....	1332
Continuation.....	1332
State Invariant.....	1334
State Lifeline.....	1335
State Machine.....	1338
Structured Activity.....	1338
Structured Node.....	1340
Sequential Node.....	1341
Loop Node.....	1341
Conditional Node.....	1345
Synch.....	1346
System Boundary.....	1347
System Boundary Properties.....	1348
Terminate.....	1350
Trigger.....	1350
Use Case.....	1352
Use Case Extension Points.....	1354
Rectangle Notation.....	1355
Value Lifeline.....	1355
<b>Structural Diagram Elements .....</b>	<b>1357</b>
Artifact.....	1358
Create File Artifacts.....	1362
Class.....	1363
Active Classes.....	1365
Parameterized Classes (Templates).....	1365
Collaboration.....	1367
Collaboration Use.....	1368
Component.....	1370
Data Type.....	1370
Deployment Spec.....	1371
Device.....	1372
Document Artifact.....	1373
Enumeration.....	1374
Execution Environment.....	1374
Expose Interface.....	1375
Information Item.....	1376
Interface.....	1377
Node.....	1378
Object.....	1379
Run-time State.....	1380
Define a Run-time Variable.....	1380
Remove a Defined Variable.....	1381
Object State.....	1381

Package.....	1382
Part.....	1383
Add Property Value.....	1383
Port.....	1384
Add a Port to an Element.....	1385
Inherited and Redefined Ports.....	1385
The Property Page.....	1386
Primitive.....	1386
Signal.....	1387
<b>UML Connectors .....</b>	<b>1389</b>
<b>Abstraction .....</b>	<b>1391</b>
<b>Aggregation .....</b>	<b>1392</b>
Change Aggregation Connector Form.....	1392
<b>Assembly .....</b>	<b>1393</b>
<b>Association .....</b>	<b>1393</b>
Qualifiers.....	1395
Qualifiers Dialog.....	1396
<b>Association Class .....</b>	<b>1398</b>
Connect New Class to Association.....	1400
<b>Communication Path .....</b>	<b>1400</b>
<b>Composition .....</b>	<b>1401</b>
<b>Connector .....</b>	<b>1402</b>
<b>Control Flow .....</b>	<b>1403</b>
<b>Delegate .....</b>	<b>1404</b>
<b>Dependency .....</b>	<b>1404</b>
Apply a Stereotype.....	1405
<b>Deployment .....</b>	<b>1406</b>
<b>Extend .....</b>	<b>1406</b>
<b>Generalization .....</b>	<b>1409</b>
<b>Include .....</b>	<b>1410</b>
<b>Information Flow .....</b>	<b>1410</b>
Using Information Flow s.....	1411
Convey Information on a Flow .....	1413
Realize an Information Flow .....	1414
<b>Interrupt Flow .....</b>	<b>1415</b>
<b>Manifest .....</b>	<b>1416</b>
<b>Message .....</b>	<b>1417</b>
Message (Sequence Diagram).....	1418
Self-Message.....	1421
Call .....	1422
Message Examples.....	1423
Change the Timing Details.....	1424
General Ordering.....	1426
Asynchronous Signal Message.....	1427
Co-Region Notation.....	1427
Message (Communication Diagram).....	1428
Create a Communication Message.....	1429
Re-Order Messages .....	1430
Message (Timing Diagram).....	1431
Create a Timing Message.....	1432
<b>Nesting .....</b>	<b>1434</b>
<b>Notelink .....</b>	<b>1435</b>
<b>Object Flow .....</b>	<b>1435</b>
Object Flow s in Activity Diagrams .....	1436
<b>Occurrence .....</b>	<b>1438</b>
<b>Package Import .....</b>	<b>1438</b>
<b>Package Merge .....</b>	<b>1439</b>
<b>Realization .....</b>	<b>1440</b>

Recursion .....	1441
Role Binding .....	1441
Represents .....	1442
Representation .....	1443
Substitution .....	1443
Template Binding .....	1444
Parameter Substitution .....	1445
Trace .....	1446
Transition .....	1446
Internal Transition .....	1449
Usage .....	1450
Use .....	1450
<b>UML Stereotypes .....</b>	<b>1452</b>
Apply Stereotypes .....	1453
Stereotype Selector .....	1455
Stereotype Visibility .....	1456
Standard Stereotypes .....	1457
Stereotypes with Alternative Images .....	1459
Custom Stereotypes .....	1461
<b>Design Patterns .....</b>	<b>1464</b>
Create a Pattern .....	1464
Import a Pattern .....	1466
Use a Pattern .....	1467
Add Pattern Dialog .....	1469

## Part VII Extending UML Models

1471

<b>Using UML Profiles .....</b>	<b>1472</b>
Add Profile Objects to a Diagram .....	1472
Tagged Values in Profiles .....	1473
Synchronize Tagged Values and Constraints .....	1473
<b>Using MDG Technologies .....</b>	<b>1475</b>
Work with MDG Technologies .....	1476
Manage MDG Technologies .....	1477
Access Remote MDG Technologies .....	1479
Import MDG Technologies to Model .....	1480
Extensions - MDG Technologies .....	1481
<b>MDG Technology SDK .....</b>	<b>1483</b>
Defining a Modeling Language .....	1483
Developing Profiles .....	1485
Create Stereotype Profiles .....	1485
Create a Profile Package .....	1487
Add Stereotypes and Metaclasses .....	1488
Create Stereotypes Extending non-UML Objects .....	1491
Define Stereotype Tagged Values .....	1492
Add an Enumeration to a Stereotype .....	1493
Define a Structured Tagged Value .....	1495
Use the Tagged Value Connector .....	1498
With Predefined Tag Types .....	1499
Define Stereotype Constraints .....	1500
Add Shape Scripts .....	1501
Set Default Appearance .....	1502
Special Attributes .....	1503
Define a Stereotype as a Metatype .....	1508
Define Multiple-Stereotype Level .....	1509
Define Creation of Instance .....	1510
Define Composite Elements .....	1511

Define Child Diagram Type.....	1512
Define Tag Groupings.....	1514
Quick Linker.....	1515
Quick Linker Definition Format.....	1516
Quick Linker Example.....	1519
Hide Default Quick Linker Settings.....	1521
Quick Linker Object Names.....	1521
Add Quick Linker Definition To Profile.....	1523
Export a Profile.....	1523
Save Profile Options.....	1525
UML Profiles in the Resources Window.....	1526
Import UML Profiles Into the Resources Window.....	1526
<b>MDG Technologies - Creating .....</b>	<b>1527</b>
Using the Profile Helpers.....	1528
Create Stereotype Profiles using Profile Helpers.....	1529
Add Stereotypes and Metaclasses using Profile Helpers.....	1531
Edit a Stereotype Element.....	1535
Create Diagram Profiles using the Profile Helpers.....	1536
Create Toolbox Profiles using the Profile Helpers.....	1538
Create Hidden Sub-Menus using the Profile Helpers.....	1543
Create MDG Technology File.....	1545
Add a Profile.....	1548
Add a Pattern.....	1549
Add a Diagram Profile.....	1550
Add a Toolbox Profile.....	1551
Add Tagged Value Types.....	1551
Add Code Modules.....	1552
Define Code Options.....	1553
Add MDA Transforms.....	1554
Add Document Report Templates.....	1555
Add Linked Document Templates.....	1555
Add Images.....	1556
Add Scripts.....	1556
Add Workspace Layouts.....	1557
Add Model View s.....	1558
Add Model Searches.....	1559
Working w ith MTS Files.....	1559
Create Toolbox Profiles.....	1560
Create Toolbox Profiles.....	1561
Toolbox Page Attributes.....	1563
Create Hidden Sub-Menus.....	1564
Assign Icons To Toolbox Items.....	1565
Override Default Toolboxes.....	1567
Elements Used in Toolboxes.....	1567
Connectors Used In Toolboxes.....	1569
Create Custom Diagram Profiles.....	1570
Built-In Diagram Types.....	1572
Attribute Values - styleex & pdata.....	1573
Set Up Technology Element Images.....	1574
Define Validation Configuration.....	1576
Incorporate Model Templates.....	1576
Add Import/Export Scripts.....	1578
Deploy An MDG Technology.....	1580
<b>Developing Programming Languages .....</b>	<b>1581</b>
<b>Shape Scripts .....</b>	<b>1582</b>
Getting Started With Shape Scripts.....	1583
Shape Editor.....	1584
Write Scripts.....	1585

Shape Attributes.....	1588
Draw ing Methods.....	1591
Color Queries.....	1598
Conditional Branching.....	1598
Query Methods.....	1599
Display Element/Connector Properties.....	1599
Sub-Shapes.....	1603
Add Custom Compartments to Element.....	1604
Show Composite Diagram.....	1608
Reserved Names.....	1611
Syntax Grammar.....	1613
Example Scripts.....	1614
<b>Tagged Value Types .....</b>	<b>1621</b>
Create Tagged Value Type from Predefined Types.....	1622
Predefined Structured Types.....	1622
Create Custom Masked Tagged Value Type.....	1626
Create Reference Data Tagged Values.....	1628
Predefined Reference Data Types.....	1629
<b>Code Template Framework .....</b>	<b>1631</b>
Code and Transform Templates.....	1632
Base Templates .....	1633
Export Code Generation and Transformation Templates.....	1637
Import Code Generation and Transformation Templates.....	1637
Synchronize Code.....	1638
Synchronize Existing Sections.....	1640
Add New Sections.....	1640
Add New Features and Elements.....	1641
The Code Template Editor.....	1641
Code Template Syntax.....	1643
Literal Text.....	1643
Variables .....	1644
Macros .....	1646
Template Substitution Macros.....	1647
Field Substitution Macros.....	1648
Substitution Examples.....	1649
Attribute Field Substitution Macros .....	1651
Class Field Substitution Macros.....	1652
Code Generation Option Field Substitution Macros.....	1655
Connector Field Substitution Macros .....	1660
Constraint Field Substitution Macros.....	1664
Effort Field Substitution Macros.....	1665
File Field Substitution Macros.....	1665
File Import Field Substitution Macros.....	1666
Link Field Substitution Macros.....	1667
Linked File Field Substitution Macros .....	1668
Metric Field Substitution Macros .....	1669
Operation Field Substitution Macros .....	1669
Package Field Substitution Macros .....	1671
Parameter Field Substitution Macros.....	1672
Problem Field Substitution Macros .....	1673
Requirement Field Substitution Macros.....	1673
Resource Field Substitution Macros .....	1674
Risk Field Substitution Macros .....	1675
Scenario Field Substitution Macros.....	1675
Tagged Value Substitution Macros.....	1676
Template Parameter Substitution Macros.....	1677
Test Field Substitution Macros.....	1677
Function Macros .....	1678

Control Macros.....	1683
List Macro.....	1683
Branching Macros.....	1684
Synchronization Macros.....	1686
The Processing Instruction (PI) Macro.....	1687
EASL Code Generation Macros.....	1688
EASL Collections.....	1690
EASL Properties.....	1693
Call Templates From Templates.....	1700
The Code Template Editor in MDG Development.....	1701
Create Custom Templates.....	1701
Customize Base Templates.....	1702
Add New Stereotyped Templates.....	1703
Override Default Templates.....	1704
<b>Grammar Framework .....</b>	<b>1705</b>
Grammar Syntax.....	1706
Grammar Instructions.....	1707
Grammar Rules.....	1708
Grammar Terms.....	1709
Grammar Commands.....	1709
AST Nodes.....	1711
Editing Grammars.....	1721
Parsing AST Results.....	1722
Profiling Grammar Parsing.....	1722
Example Grammars.....	1723

## Part VIII Requirement Models

**1726**

<b>Specification Manager .....</b>	<b>1728</b>
<b>Specification Manager - Overview .....</b>	<b>1729</b>
<b>Using the Specification Manager .....</b>	<b>1731</b>
Select Specification Type.....	1734
Select Specification Package.....	1735
Customize Columns.....	1738
Adding Elements.....	1738
Deleting Elements.....	1739
Editing Elements.....	1740
Indicator Columns.....	1743
<b>Traceability .....</b>	<b>1744</b>
Create Matrix Profile.....	1744
Open Relationship Matrix.....	1746
<b>Reviewing Elements .....</b>	<b>1747</b>
Create Review Document.....	1749
View Review Document.....	1751
<b>Managing Changes .....</b>	<b>1752</b>
<b>Reporting .....</b>	<b>1754</b>
<b>Specification Manager Configuration .....</b>	<b>1756</b>
Appearance Options.....	1757
Model Options.....	1758
Package Options.....	1759
<b>Requirements .....</b>	<b>1763</b>
<b>Model Requirements .....</b>	<b>1765</b>
<b>Requirements Diagram .....</b>	<b>1767</b>
Example Requirements Diagram.....	1768
<b>Feature .....</b>	<b>1769</b>
<b>Internal Requirements .....</b>	<b>1770</b>
Make Internal Requirement External.....	1771
<b>Create Requirements .....</b>	<b>1773</b>

Requirement Properties .....	1774
Color Code External Requirements .....	1776
Extend Requirement Properties .....	1777
Display Tagged Values On Diagrams .....	1778
Connect Requirements .....	1779
Import Requirements and Hierarchies in CSV .....	1780
<b>Manage Requirements .....</b>	<b>1782</b>
View Requirements .....	1783
Trace Use of Requirements .....	1784
Manage Requirement Changes .....	1785
Report on Requirements .....	1787

## **Part IX Domain Based Models 1789**

<b>Domain Based Diagrams .....</b>	<b>1793</b>
<b>Analysis Models .....</b>	<b>1794</b>
Mind Mapping .....	1794
Custom Diagram .....	1796
Data Flow Diagrams .....	1797
Analysis Stereotypes .....	1800
Analysis Diagram .....	1801
Example Analysis Diagram.....	1802
<b>Business Models .....</b>	<b>1804</b>
<b>Business Modeling/Interaction .....</b>	<b>1805</b>
Example Business Modeling Diagram.....	1805
<b>Business Models .....</b>	<b>1807</b>
Analysis Models.....	1809
Process Modeling Notation.....	1810
Inputs, Resources and Information.....	1811
Events .....	1812
Outputs .....	1812
Goals .....	1813
A Complete Business Process.....	1813
<b>Modeling Business Rules .....</b>	<b>1814</b>
Develop a Business Rules Model.....	1816
Generate a Business Rules Model.....	1818
Model Business Rules.....	1821
Create a Business Domain Model.....	1824
Create a Rule Flow Activity .....	1826
Pass Parameters to Rule Flow Activity.....	1829
Model Rules In an Operation.....	1830
Compose Business Rules .....	1833
Add and Remove Rules .....	1834
Define Rule Conditions.....	1836
Define Rule Actions .....	1837
Bind Rules to Conditions and Actions.....	1839
Define Computation Rules.....	1840
Validate Business Rules.....	1841
Export Composed Rules to CSV .....	1842
Code Generation For Business Rules.....	1843
<b>BPMN Models .....</b>	<b>1845</b>
BPMN 1.0 and 1.1 Toolbox Pages .....	1848
BPMN 2.0 Toolbox Pages .....	1851
BPMN 2.0 Business Process Toolbox Pages.....	1852
BPMN 2.0 Choreography Toolbox Pages.....	1854
BPMN 2.0 Collaboration Toolbox Page.....	1856
BPMN 2.0 Conversation Toolbox Pages.....	1859
BPMN 2.0 Type Toolbox Page.....	1861

Change BPMN Element Appearance.....	1862
Migrate BPMN 1.0 Model to BPMN 1.1.....	1866
Migrate BPMN 1.1 Model to BPMN 2.0.....	1868
BPMN 2.0 XML.....	1869
<b>BPEL Models .....</b>	<b>1870</b>
BPEL 1.1 Model.....	1871
Create BPEL 1.1 Model Structure.....	1873
Model a BPEL 1.1 Process.....	1876
Start Event.....	1877
Intermediate Event.....	1878
Activity .....	1881
Gateway.....	1884
End Event.....	1885
Sequence Flow .....	1887
Pool .....	1888
Assignment.....	1889
Create BPEL 1.1 Web Service.....	1890
Generate BPEL 1.1.....	1893
BPEL 2.0 Model.....	1894
Create BPEL 2.0 Model Structure.....	1897
Model a BPEL 2.0 Process.....	1898
Start Event.....	1900
Intermediate Event.....	1901
Activity .....	1904
Gateway.....	1906
End Event.....	1907
Data Object.....	1909
Property .....	1910
Sequence Flow .....	1910
Pool .....	1911
Assignment.....	1912
Create BPEL 2.0 Web Service Operation.....	1914
Generate BPEL 2.0.....	1916
BPEL Model Validation.....	1917
<b>SPEM .....</b>	<b>1919</b>
SPEM Toolbox Pages.....	1920
Example SPEM Diagram.....	1924
<b>ArchiMate .....</b>	<b>1926</b>
Migrate ArchiMate 1.0 to ArchiMate 2.0.....	1927
<b>Eriksson-Penker Extensions .....</b>	<b>1929</b>
<b>Decision Models .....</b>	<b>1931</b>
Decision Table Editor.....	1931
Code Generation from Decision Models.....	1935
<b>Data Models .....</b>	<b>1937</b>
Conceptual Data Model .....	1938
Logical Data Model .....	1939
Entity Relationship Diagrams (ERDs) .....	1939
Table .....	1942
Database Schema .....	1943
<b>Geodatabase Design for ArcGIS .....</b>	<b>1944</b>
ArcGIS UML Profile .....	1944
ArcGIS Toolbox Pages.....	1946
Connectivity Rule Examples.....	1952
Topology Example.....	1954
Relationship Rule Example.....	1956
Setting ArcGIS Coordinate Systems .....	1958
Export ArcGIS XML Workspace .....	1961
Export Modular ArcGIS Schemas .....	1962



Import ArcGIS XML Workspace .....	1968
Validate an ArcGIS Workspace .....	1970
<b>MDG Technology for ODM .....</b>	<b>1971</b>
ODM Toolbox Pages .....	1972
OWL Elements & Relationships .....	1974
RDF Elements & Relationships .....	1977
Example ODM Diagrams .....	1979
ODM Commands .....	1980
<b>MDG Technology For GML .....</b>	<b>1983</b>
UML Profile for GML .....	1983
GML Toolbox Pages .....	1984
Generate GML Application Schema .....	1987
<b>User Interface Models .....</b>	<b>1990</b>
Example User Interface Diagram .....	1991
Screen .....	1991
UI Control Elements .....	1992
Web Stereotypes .....	1995
Win32 UI Technology .....	1996
<b>Other Stereotypes .....</b>	<b>1997</b>
Boundary .....	1997
Create a Boundary .....	1998
Control .....	1999
Create a Control Element .....	1999
Entity .....	2000
Create an Entity .....	2001
Event .....	2001
Hyperlinks .....	2002
Hyperlinks To Files .....	2004
Script Hyperlinks .....	2004
Add Action As Hyperlink .....	2004
Hyperlinks Between Diagrams .....	2005
Image .....	2006
N-Ary Association .....	2007
Process .....	2008
Packaging Component .....	2008
Risk .....	2009
Task .....	2010
Test Case .....	2010

## Part X Model Transformation

2013

<b>Transform Elements .....</b>	<b>2017</b>
Chaining Transformations .....	2019
<b>Built-in Transformations .....</b>	<b>2020</b>
C# Transformation .....	2021
C++ Transformation .....	2023
Data Model To ERD Transformation .....	2024
DDL Transformation .....	2025
EJB Transformations .....	2029
ERD To Data Model Transformation .....	2032
Java Transformation .....	2034
JUnit Transformation .....	2037
NUnit Transformation .....	2038
PHP Transformation .....	2040
Sequence/Communication Diagram Transformations .....	2041
VB.Net Transformation .....	2042
WSDL Transformation .....	2043

XSD Transformation .....	2044
<b>Edit Transformation Templates .....</b>	<b>2048</b>
<b>Write Transformations .....</b>	<b>2051</b>
Default Transformation Templates .....	2052
Intermediary Language .....	2053
Intermediary Language Debugging .....	2053
Objects .....	2055
Connectors .....	2060
Transform Connectors .....	2063
Transform Foreign Keys .....	2065
Copy Information .....	2066
Convert Types .....	2066
Convert Names .....	2067
Cross References .....	2068
Transform Template Parameter Substitution .....	2069

## **Part XI Software Engineering 2073**

<b>Development Tools .....</b>	<b>2074</b>
Overview of Development .....	2076
Application Patterns (Model + Code) .....	2077
<b>MDG Integration and Code Engineering .....</b>	<b>2079</b>
<b>Modeling Conventions .....</b>	<b>2080</b>
ActionScript Conventions .....	2081
Ada 2005 Conventions .....	2082
C Conventions .....	2084
Object Oriented Programming In C.....	2086
C# Conventions .....	2088
C++ Conventions .....	2090
Managed C++ Conventions .....	2093
C++/CLI Conventions.....	2094
Delphi Conventions .....	2095
Java Conventions .....	2096
AspectJ Conventions.....	2098
PHP Conventions .....	2099
Python Conventions .....	2100
SystemC Conventions .....	2100
VB.NET Conventions .....	2102
Verilog Conventions .....	2105
VHDL Conventions .....	2106
Visual Basic Conventions .....	2109
<b>Generate Source Code .....</b>	<b>2111</b>
Generate a Single Class .....	2113
Generate a Group of Classes .....	2114
Generate a Package .....	2115
Update Package Contents .....	2117
Synchronize Model and Code.....	2118
Namespaces .....	2120
<b>Generate From Behavioral Models .....</b>	<b>2121</b>
Code Generation - State Machines .....	2122
Legacy State Machine Templates.....	2126
Java Code Generated From Legacy State Machine Template.....	2128
State Machine Modeling For HDLs .....	2131
Code Generation - Interaction Diagrams .....	2133
Code Generation - Activity Diagrams .....	2134
<b>Importing Source Code .....</b>	<b>2136</b>

Import Source Code .....	2138
Notes on Source Code Import .....	2139
Import Resource Script .....	2141
Import a Directory Structure .....	2142
Import Binary Module .....	2143
Classes Not Found During Import .....	2144
<b>Editing Source Code .....</b>	<b>2146</b>
Languages Supported .....	2147
Compare Editors .....	2148
Code Editor Toolbar .....	2149
Code Editor Context Menu .....	2152
Create Use Case for Method.....	2155
Code Editor Functions .....	2157
Function Details.....	2157
Intelli-sense.....	2160
Find and Replace.....	2161
Search in Files.....	2164
Search Intelli-sense .....	2166
Code Editor Key Bindings .....	2167
<b>Code Build &amp; Debug .....</b>	<b>2172</b>
Getting Started .....	2173
Prerequisites .....	2173
General Workflow .....	2173
Workspace Layout and Tools .....	2174
Analyzer Scripts .....	2175
Managing Analyzer Scripts.....	2175
Analyzer Script Editor .....	2179
Add Build Commands .....	2180
Clean .....	2181
Testing.....	2182
Add Testing Command.....	2182
Testpoints Output.....	2184
Set up a Debug script.....	2185
Operating System Specific Requirements .....	2186
UAC-Enabled Operating Systems.....	2187
WINE Debugging.....	2188
Microsoft C++ and Native (C, VB).....	2189
General Setup.....	2189
Debug Symbols.....	2191
Java .....	2191
General Setup for Java.....	2191
Advanced Techniques.....	2194
Attach to Virtual Machine.....	2194
Internet Browser Java Applets.....	2194
Working with Java Web Servers.....	2195
JBoss Server.....	2198
Apache Tomcat Server.....	2199
Apache Tomcat Windows Service.....	2199
.NET .....	2200
General Setup for .NET.....	2200
Debugging an Unmanaged Application .....	2201
Debug COM Interop.....	2202
Debug ASP .NET .....	2203
The PHP Debugger.....	2204
PHP Debugger - System Requirements.....	2206
PHP Debugger Checklist.....	2207
The GNU Debugger (GDB).....	2209
The Android Debugger.....	2210

Java JDWP Debugger .....	2213
Tracepoint Output .....	2215
Workbench Setup .....	2216
Add Run Command .....	2217
Add Deploy Command .....	2218
Configure Recording Detail .....	2219
<b>Build Application .....</b>	<b>2221</b>
Locate Compiler Errors in Code .....	2221
<b>Debugging .....</b>	<b>2222</b>
Before Starting .....	2222
Breakpoint and Marker Management .....	2224
Setting Code Breakpoints .....	2226
Breakpoint Properties .....	2226
Trace Statements .....	2227
Failure to Bind Breakpoint .....	2229
Break When a Variable Changes Value .....	2229
Trace When Variable Changes Value .....	2230
Debugger Facilities .....	2231
Run the Debugger .....	2231
View the Local Variables .....	2234
View Content Of Long Strings .....	2235
View Debug Variables in Code Editors .....	2236
Variable Snapshots .....	2236
View Variables in Other Scopes .....	2237
View Elements of Array .....	2238
View the Call Stack .....	2239
Create Sequence Diagram of Call Stack .....	2241
Inspect Process Memory .....	2242
Show Loaded Modules .....	2243
Debug Another Process .....	2243
Process First Chance Exceptions .....	2244
<b>Other Settings .....</b>	<b>2246</b>
<b>Source Code Engineering Options .....</b>	<b>2246</b>
Source Code Options .....	2247
Import Component Types .....	2249
Options - Code Editors .....	2250
Editor Language Properties .....	2251
Options - Object Lifetimes .....	2253
Options - Attribute/Operations .....	2253
Code Page for Source Editing .....	2255
<b>Local Paths .....</b>	<b>2255</b>
<b>Local Paths Dialog .....</b>	<b>2256</b>
<b>Language Macros .....</b>	<b>2257</b>
<b>Set Collection Classes .....</b>	<b>2259</b>
Example Use of Collection Classes .....	2260
<b>Language Options .....</b>	<b>2262</b>
ActionScript Options .....	2264
Ada 2005 Options .....	2264
ArcGIS Options .....	2265
C Options .....	2266
C# Options .....	2268
C++ Options .....	2269
Delphi Options .....	2271
Delphi Properties .....	2272
Java Options .....	2272
Python Options .....	2273
PHP Options .....	2274
SystemC Options .....	2275

VB.NET Options.....	2275
Verilog Options.....	2276
VHDL Options.....	2277
Visual Basic Options.....	2277
MDG Technology Language Options.....	2278
Reset Options.....	2279
<b>Code Template Framework Tool .....</b>	<b>2281</b>
<b>ICONIX .....</b>	<b>2282</b>
<b>GoF Patterns .....</b>	<b>2285</b>

## **Part XII Systems Engineering 2288**

<b>Systems Modeling Language (SysML) .....</b>	<b>2294</b>
<b>SysML Activity Toolbox .....</b>	<b>2295</b>
<b>SysML Block Definition Toolbox .....</b>	<b>2300</b>
Generate Properties From Block Associations .....	2304
Block Element Compartments .....	2305
Show Direction on SysML 1.3 Ports .....	2306
<b>SysML Interaction Toolbox .....</b>	<b>2306</b>
<b>SysML Internal Block Toolbox .....</b>	<b>2308</b>
<b>SysML Model Elements Toolbox .....</b>	<b>2310</b>
<b>SysML Parametrics Toolbox .....</b>	<b>2312</b>
<b>SysML Requirements Toolbox .....</b>	<b>2314</b>
<b>SysML State Machine Toolbox .....</b>	<b>2316</b>
<b>SysML Use Case Toolbox .....</b>	<b>2318</b>
<b>SysML Parametric Models .....</b>	<b>2320</b>
Simulate a SysML Model .....	2322
<b>A SysML Requirements Model .....</b>	<b>2325</b>
<b>A SysML Operational Domain Model .....</b>	<b>2326</b>
<b>Compose System Design .....</b>	<b>2328</b>
<b>Create Reusable Subsystems .....</b>	<b>2330</b>
<b>Migrate SysML Model to Later SysML Version .....</b>	<b>2331</b>

## **Part XIII Database Engineering 2334**

<b>Physical Data Model .....</b>	<b>2335</b>
<b>Create a Data Model Diagram .....</b>	<b>2336</b>
Example Data Model Diagram.....	2337
<b>Tables and Columns .....</b>	<b>2338</b>
Create a Table.....	2339
Working with Table Properties.....	2340
Set the Database Type.....	2341
Set Table Owner/Schema.....	2342
Set MySQL Options.....	2342
Set Oracle Table Properties.....	2343
Create Columns.....	2345
Reorder Columns .....	2347
<b>Manage DBMS Settings .....</b>	<b>2347</b>
<b>Data Types .....</b>	<b>2348</b>
DBMS Product Conversion for a Package.....	2349
Data Type Conversion For a Table.....	2350
Add New Datatypes.....	2350
MySQL Data Types.....	2352
Oracle Data Types.....	2352
Map Data Types Between DBMS Products.....	2353
<b>Database Keys .....</b>	<b>2354</b>

Primary Key.....	2355
Create a Primary Key.....	2356
Define a Primary Key Name Template.....	2357
SQL Server Non Clustered Keys.....	2358
Foreign Key.....	2358
Create a Foreign Key.....	2359
Composite Foreign Key.....	2361
Define a Foreign Key Name Template.....	2362
Define a Foreign Key Index Template.....	2363
<b>Stored Procedures</b> .....	<b>2364</b>
Create a Stored Procedure.....	2364
<b>Indexes</b> .....	<b>2365</b>
<b>Triggers</b> .....	<b>2368</b>
Create a Trigger .....	2369
<b>Create a Check Constraint</b> .....	<b>2370</b>
<b>Default Constraints</b> .....	<b>2371</b>
<b>Views</b> .....	<b>2371</b>
Create a View .....	2372
<b>Oracle Packages</b> .....	<b>2374</b>
<b>Supported Databases</b> .....	<b>2375</b>
<b>Import Database Schema</b> .....	<b>2376</b>
Select a Data Source .....	2378
Select Tables .....	2378
The Imported Class Elements .....	2379
<b>Generate DDL</b> .....	<b>2380</b>
Generate DDL For a Table .....	2380
Generate DDL for a Package .....	2381
<b>Data Modeling Notations</b> .....	<b>2384</b>

## Part XIV SOA and XML

**2386**

<b>XML Schema - XSD</b> .....	<b>2387</b>
<b>Model XSD</b> .....	<b>2387</b>
Example XML Schema Diagram.....	2389
Schema Package.....	2390
Global Element.....	2392
Local Element.....	2394
Global Attribute.....	2396
Local Attribute.....	2398
Attribute Group.....	2399
Complex Type.....	2401
Simple Type.....	2402
Group.....	2404
Any.....	2405
Any Attribute.....	2407
Union.....	2408
Model Group.....	2410
Enumeration.....	2412
<b>XML from Abstract Class Models</b> .....	<b>2413</b>
Default UML to XSD Mappings.....	2415
<b>Generate XSD</b> .....	<b>2417</b>
Generate Global Element.....	2418
<b>Import XSD</b> .....	<b>2420</b>
Global Element and ComplexType.....	2421
<b>Web Services - WSDL</b> .....	<b>2423</b>
<b>WSDL 1.1 Model Structure</b> .....	<b>2423</b>
<b>Model WSDL</b> .....	<b>2426</b>

WSDL Namespace.....	2429
WSDL Message.....	2430
WSDL Message Part.....	2431
WSDL Port Type.....	2433
WSDL Port Type Operation.....	2434
WSDL Binding.....	2437
WSDL Binding Operation.....	2438
WSDL Service.....	2441
WSDL Document.....	2443
Generate WSDL .....	2445
Import WSDL .....	2447
SoaML .....	2449
SoaML Toolbox Pages .....	2451
SOMF 2.1 .....	2454
MOF .....	2456
Create MOF Diagrams .....	2458
Export MOF Model to XMI .....	2460

## **Part XV Model Simulation 2463**

How It Works .....	2466
How it Looks .....	2468
Simulation Windows .....	2469
Set Up Simulation Script .....	2472
Activate Simulation Script .....	2474
Run Model Simulation .....	2475
Simulation Breakpoints .....	2477
Objects and Instances in Simulation .....	2479
Create Objects in a Simulation .....	2480
Destroy Objects in a Simulation .....	2483
Dynamic Simulation with Javascript .....	2486
Call Behaviors .....	2489
Interaction Operand Condition and Message Behavior .....	2490
Guards and Effects .....	2492
Triggers .....	2494
Action Behavior By Type .....	2497
Structured Activity Simulation .....	2499
Activity Return Value Simulation .....	2501
Simulation Events Window .....	2504
Waiting Triggers .....	2508
Re-Signal Triggers .....	2509
Trigger Parameters .....	2510
Trigger Sets and Auto-Firing .....	2512
Using Trigger Sets to Simulate an Event Sequence .....	2515
Multi-threading - Forks and Joins .....	2516
Multi-threading - Concurrent State Regions .....	2517
Using Composite Diagrams .....	2518
Win32 Dialog Simulation .....	2520
BPMN Simulation .....	2521

Create a BPMN Simulation Model .....	2521
Initialize Variables and Conditions .....	2523
Compare UML Activities to BPMN Processes .....	2524

## Part XVI Execution Analyzer 2527

Visual Execution Analyzer Samples .....	2529
Recording Sequence Diagrams .....	2531
How it Works .....	2532
The Recording History .....	2533
Diagram Features .....	2535
Setup for Recording .....	2535
Control Stack Depth .....	2536
Place Recording Markers .....	2536
Set Record Markers .....	2537
Marker Types .....	2538
The Breakpoints & Markers Window .....	2540
Working with Marker Sets .....	2541
Recording Activity for a Class .....	2542
Control the Recording Session .....	2544
Recorder Toolbar .....	2544
Working With Recording History .....	2546
Start Recording .....	2547
Step Through Function Calls .....	2548
Nested Recording Markers .....	2548
Generating Sequence Diagrams .....	2549
Reporting State Transitions .....	2550
Reporting a State Machine .....	2551
Recording and Mapping State Changes .....	2553
Profiling .....	2555
System Requirements .....	2556
Profiler Operation .....	2557
Getting Started .....	2558
Generate, Save and Load Profile Reports .....	2560
Setting Options .....	2563
Start & Stop the Profiler .....	2564
Function Line Reports .....	2564
Save Report in Team Review .....	2566
Object Workbench .....	2567
How it Works .....	2567
Create & Delete Workbench Instances .....	2568
Invoke Methods .....	2570
Unit Testing .....	2573
Set Up Unit Testing .....	2573
Run Unit Tests .....	2575
Record Test Results .....	2575
Testpoint Management .....	2577
The Testpoints Window .....	2578
The Testpoints Window Toolbar .....	2580
Constraint Composition .....	2581
Testpoint Editor .....	2584
Combine Testpoints .....	2586
Test Cut .....	2588
Test Set .....	2588
Test Suite .....	2589
Object Run State Diagrams .....	2591



## Part XVII Testing 2593

<b>Model Validation .....</b>	<b>2594</b>
Configure Model Validation .....	2596
Run Validation .....	2596
Rules Reference .....	2597
Well-Formedness .....	2599
Element Composition.....	2600
Property Validity.....	2600
OCL Conformance.....	2601
<b>Testing .....</b>	<b>2604</b>
Working On Test Records .....	2605
Create Test Records .....	2607
Move or Copy Tests Between Categories .....	2610
Import Scenario as Test .....	2611
Import Test From Other Elements .....	2613
Import Responsibility or Constraint as Test .....	2614
Create Maintenance Item From Test .....	2615
Show Test Script Compartments .....	2616
Test Documentation .....	2617

## Part XVIII Maintenance 2619

Introduction to Maintenance .....	2621
Working on Maintenance Items .....	2623
Create Maintenance Items .....	2625
Move or Copy Maintenance Items .....	2628
Create Elements From Maintenance Item .....	2629
Show Maintenance Items in Diagram .....	2630
Changes and Issues .....	2631
Issues (Defects) .....	2631
Changes .....	2633
Maintenance Diagram .....	2634
Example Maintenance Diagram.....	2635

## Part XIX Reporting 2638

<b>Document Reports .....</b>	<b>2640</b>
Generate Document Reports .....	2642
Generate Documentation.....	2644
Selecting a Template.....	2648
System Document Templates.....	2648
Selecting a Table of Contents.....	2652
Selecting a Stylesheet.....	2653
Selecting a Cover Page.....	2654
Document Options.....	2654
Exclude Package Query and Script.....	2658
Exclude Filters.....	2660
Element Filters .....	2660
Other Filters.....	2663
Project Constants.....	2664
Word Substitution.....	2665
Language Substitution.....	2666
Diagram Options.....	2667
Resource Documents.....	2668
Virtual Documents .....	2669

Create Master Document.....	2672
Create Model Document.....	2673
Add Packages to Model Document.....	2675
Delete Package in Model Document.....	2676
Document Order.....	2677
Generate the Document.....	2679
Generate Report to an Artifact Element.....	2680
<b>Custom Document Templates .....</b>	<b>2681</b>
Design Custom Document Templates.....	2684
Setting Sections for Reporting.....	2688
Child Objects.....	2690
Report on Embedded Elements .....	2691
Reporting Profiled Relationship Matrices.....	2692
Report on Constraints and Scenarios.....	2693
Reporting Linked Documents.....	2694
Report Elements From External Packages .....	2695
Report on Tagged Values.....	2698
Create Sections as Tables.....	2700
Add Section Content.....	2702
Custom Template Design Options.....	2704
Notes on Creating Stylesheets.....	2705
Notes on Creating Tables of Contents.....	2706
Notes on Creating Cover Pages.....	2707
Import a Document Template.....	2707
Template Fragments.....	2708
Creating a Template Fragment.....	2709
Custom Query Fragments.....	2711
Custom SQL Fragments.....	2712
Custom Script Fragments.....	2714
Example Template Fragment Script.....	2716
Example Output of a Template Fragment Script.....	2717
Adding Fragments to a Document Template .....	2718
<b>The Legacy RTF Report Generator .....</b>	<b>2720</b>
Document a Single Element.....	2721
Set the Main RTF Properties.....	2721
Apply a Filter .....	2722
Exclude Elements .....	2722
RTF Diagram Format.....	2723
Model Include.....	2723
RTF Report Options.....	2723
RTF Report Selections.....	2724
Custom Language Settings .....	2725
Generate the Report.....	2726
Legacy Report Style Templates .....	2726
Save as Document.....	2728
Use Microsoft Word.....	2729
Open a Report in Microsoft Word.....	2729
Change Linked Images to Embedded.....	2730
Document Bookmarks .....	2730
Features of Word.....	2732
Add Table of Contents.....	2733
Add Table of Figures .....	2733
Add Headers and Footers .....	2734
Manipulate Tables in Word.....	2734
Refresh Links.....	2735
<b>System Documents .....</b>	<b>2736</b>
Testing Details Report.....	2736
Implementation Details Report.....	2737

Implementation Targets Dialog.....	2739
Dependency Details Report.....	2739
Maintenance Report.....	2740
Diagrams Only Report.....	2741
Testing Report.....	2742
<b>Web Reports .....</b>	<b>2744</b>
Create a Web Page Report .....	2744
Create Web Style Templates .....	2747
HTML Template Fragments.....	2749
Master Documents and Model Documents .....	2759
Exclude Package from Report .....	2761
<b>Charts .....</b>	<b>2762</b>
Chart Elements .....	2763
Define a Model View Chart .....	2765
Define a Time Series Chart .....	2767
Standard Chart Data .....	2770
Source Package.....	2771
Element Filters in Standard Charts.....	2772
Custom Query .....	2773
External Data.....	2773
Chart Appearance .....	2774
2D Bar Chart.....	2775
3D Bar Chart.....	2778
Pie Chart.....	2782
Time Series Chart.....	2785
Including Charts in Reports .....	2788

## Part XX Automation and Scripting 2790

<b>Scripting .....</b>	<b>2791</b>
Scripts Tab .....	2792
Script Group Properties.....	2794
Console Tab .....	2796
Script Editor .....	2798
Session Object .....	2800
Script Debugging .....	2801
Simulation Scripts .....	2802
<b>Enterprise Architect Object Model .....</b>	<b>2804</b>
Using the Automation Interface .....	2805
Connect to the Interface.....	2805
Set References In Visual Basic.....	2808
Examples and Tips .....	2809
Call from Enterprise Architect.....	2810
Available Resources.....	2811
Reference .....	2812
Interface Overview Package.....	2813
App Object.....	2814
Enumerations.....	2815
ConstLayoutStyles.....	2816
CreateBaselineFlag.....	2817
CreateModelType.....	2818
DocumentBreak.....	2818
DocumentPageOrientation.....	2818
DocumentType.....	2819
EAEditionTypes.....	2819
EnumRelationSetType.....	2820
ExportPackageXMIFlag.....	2820

MDGMenu.....	2821
MessageFlag.....	2821
ObjectType.....	2822
PropType.....	2823
ReloadType.....	2823
ScenarioDiagramType.....	2824
ScenarioStepType.....	2825
ScenarioTestType.....	2825
XMIType.....	2825
Repository Package.....	2826
Author Class.....	2827
Client Class.....	2828
Collection Class.....	2829
Datatype Class.....	2831
EventProperties Class.....	2833
EventProperty Class.....	2833
ModelWatcher Class.....	2834
Package Class.....	2835
ProjectIssues Class.....	2844
ProjectResource Class.....	2846
ProjectRole Class.....	2847
PropertyType Class.....	2848
Reference Class.....	2848
Repository Class.....	2850
Stereotype Class.....	2873
Task Class.....	2874
Term Class.....	2876
Element Package.....	2877
Constraint Class.....	2879
Effort Class.....	2880
Element Class.....	2881
File Class.....	2895
Issue (Maintenance) Class.....	2896
Metric Class.....	2898
Requirement Class.....	2899
Resource Class.....	2900
Risk Class.....	2902
Scenario Class.....	2903
ScenarioExtension Class.....	2904
ScenarioStep Class.....	2905
TaggedValue Class.....	2907
Test Class.....	2908
Element Features Package.....	2910
Attribute Class.....	2911
AttributeConstraint Class.....	2915
AttributeTag Class.....	2916
CustomProperties Collection.....	2918
EmbeddedElements Collection.....	2918
Method Class.....	2919
MethodConstraint Class.....	2923
MethodTag Class.....	2924
Parameter Class.....	2925
ParamTag Class.....	2927
Partitions Collection.....	2928
Properties Class.....	2929
TemplateParameter Class.....	2931
Transitions Collection.....	2932
Connector Package.....	2933

Connector Class .....	2933
ConnectorConstraint Class.....	2939
ConnectorEnd Class .....	2940
ConnectorTag Class .....	2943
RoleTag Class.....	2944
TemplateBinding Class.....	2946
Diagram Package.....	2948
Diagram Class.....	2948
DiagramLinks Class.....	2955
DiagramObject Class.....	2956
SwimlaneDef Class.....	2958
Swimlanes Class .....	2960
Swimlane Class .....	2961
Project Interface Package.....	2962
Project Class.....	2962
Document Generator Interface Package.....	2985
DocumentGenerator Class.....	2985
Mail Interface Package.....	2990
MailInterface Class.....	2990
Simulation Package.....	2993
Simulation Class.....	2993
Code Samples .....	2994
Open the Repository.....	2995
Iterate Through a .EAP File.....	2996
Add and Manage Packages.....	2996
Add and Manage Elements .....	2997
Add a Connector.....	2998
Add and Manage Diagrams.....	2999
Add and Delete Features.....	3000
Element Extras .....	3001
Repository Extras .....	3004
Stereotypes .....	3006
Work With Attributes.....	3006
Work With Methods.....	3007
<b>Enterprise Architect Add-In Model .....</b>	<b>3010</b>
<b>Add-In Tasks .....</b>	<b>3011</b>
Create Add-Ins.....	3012
Define Menu Items.....	3013
Deploy Add-Ins .....	3014
Tricks and Traps.....	3016
<b>The Add-In Manager .....</b>	<b>3018</b>
<b>Add-In Search .....</b>	<b>3019</b>
XML Format (Search Data).....	3020
<b>Add-In Events .....</b>	<b>3021</b>
EA_Connect.....	3022
EA_Disconnect.....	3023
EA_GetMenuItems.....	3023
EA_GetMenuState.....	3024
EA_MenuClick.....	3025
EA_OnOutputItemClicked.....	3027
EA_OnOutputItemDoubleClicked.....	3028
EA_Show Help.....	3029
<b>Broadcast Events .....</b>	<b>3029</b>
Add-In License Management Events.....	3031
EA_AddinLicenseValidate.....	3031
EA_AddinLicenseGetDescription.....	3032
EA_GetSharedAddinName.....	3033
Compartment Events .....	3034

EA_QueryAvailableCompartments.....	3034
EA_GetCompartmentData.....	3035
Context Item Events.....	3037
EA_OnContextItemChanged.....	3037
EA_OnContextItemDoubleClicked.....	3038
EA_OnNotifyContextItemModified.....	3039
EA_FileClose.....	3040
EA_FileNew.....	3041
EA_FileOpen.....	3042
EA_OnPostCloseDiagram.....	3042
EA_OnPostInitialized.....	3043
EA_OnPostOpenDiagram.....	3044
EA_OnPostTransform.....	3044
EA_OnPreExitInstance.....	3045
EA_OnRetrieveModelTemplate.....	3046
EA_OnTabChanged.....	3047
Model Validation Broadcasts.....	3047
EA_OnInitializeUserRules.....	3048
EA_OnStartValidation.....	3049
EA_OnEndValidation.....	3050
EA_OnRunElementRule.....	3050
EA_OnRunPackageRule.....	3051
EA_OnRunDiagramRule.....	3052
EA_OnRunConnectorRule.....	3053
EA_OnRunAttributeRule.....	3053
EA_OnRunMethodRule.....	3054
EA_OnRunParameterRule.....	3055
Model Validation Example.....	3056
Post-New Events.....	3060
EA_OnPostNew Element.....	3061
EA_OnPostNew Connector.....	3062
EA_OnPostNew Diagram.....	3063
EA_OnPostNew DiagramObject.....	3063
EA_OnPostNew Attribute.....	3064
EA_OnPostNew Method.....	3065
EA_OnPostNew Package.....	3066
EA_OnPostNew GlossaryTerm.....	3067
Pre-Deletion Events.....	3067
EA_OnPreDeleteElement.....	3068
EA_OnPreDeleteAttribute.....	3069
EA_OnPreDeleteMethod.....	3070
EA_OnPreDeleteConnector.....	3071
EA_OnPreDeleteDiagram.....	3071
EA_OnPreDeleteDiagramObject.....	3072
EA_OnPreDeletePackage.....	3073
EA_OnPreDeleteGlossaryTerm.....	3074
Pre New -Object Events.....	3075
EA_OnPreNew Element.....	3076
EA_OnPreNew Connector.....	3077
EA_OnPreNew Diagram.....	3078
EA_OnPreNew DiagramObject.....	3078
EA_OnPreDropFromTree.....	3079
EA_OnPreNew Attribute.....	3080
EA_OnPreNew Method.....	3081
EA_OnPreNew Package.....	3082
EA_OnPreNew GlossaryTerm.....	3083
Tagged Value Broadcasts.....	3084
EA_OnAttributeTagEdit.....	3084

EA_OnConnectorTagEdit .....	3085
EA_OnElementTagEdit .....	3086
EA_OnMethodTagEdit .....	3087
Technology Events .....	3088
EA_OnInitializeTechnologies .....	3089
EA_OnPreActivateTechnology .....	3089
EA_OnPostActivateTechnology .....	3090
EA_OnPreDeleteTechnology .....	3091
EA_OnDeleteTechnology .....	3092
EA_OnImportTechnology .....	3094
<b>Custom Views .....</b>	<b>3095</b>
Create a Custom View .....	3095
<b>Custom Docked Window .....</b>	<b>3096</b>
<b>MDG Add-Ins .....</b>	<b>3097</b>
MDG Events .....	3098
MDGBuild Project .....	3099
MDGConnect .....	3100
MDGDisconnect .....	3101
MDGGetConnectedPackages .....	3102
MDGGetProperty .....	3102
MDGMerge .....	3103
MDGNew Class .....	3106
MDGPostGenerate .....	3107
MDGPostMerge .....	3108
MDGPreGenerate .....	3109
MDGPreMerge .....	3109
MDGPreReverse .....	3110
MDGRunExe .....	3111
MDGView .....	3112

## Part XXI Glossary

**3115**

<b>A .....</b>	<b>3116</b>
<b>B .....</b>	<b>3119</b>
<b>C .....</b>	<b>3121</b>
<b>D .....</b>	<b>3125</b>
<b>E .....</b>	<b>3128</b>
<b>F .....</b>	<b>3130</b>
<b>G .....</b>	<b>3132</b>
<b>H .....</b>	<b>3133</b>
<b>I .....</b>	<b>3134</b>
<b>J .....</b>	<b>3136</b>
<b>L .....</b>	<b>3137</b>
<b>M .....</b>	<b>3138</b>
<b>N .....</b>	<b>3141</b>
<b>O .....</b>	<b>3142</b>
<b>P .....</b>	<b>3143</b>
<b>Q .....</b>	<b>3147</b>
<b>R .....</b>	<b>3148</b>
<b>S .....</b>	<b>3151</b>
<b>T .....</b>	<b>3156</b>
<b>U .....</b>	<b>3158</b>

V .....	3160
---------	------

## **Part XXII License Management 3162**

Finding Your License Information .....	3164
--	------

Adding License Keys .....	3165
---------------------------	------

Keystore Troubleshooting .....	3168
--------------------------------	------

Upgrade an Existing License .....	3169
-----------------------------------	------

Register Add-In .....	3171
-----------------------	------

Add an Add-In Key .....	3172
-------------------------	------

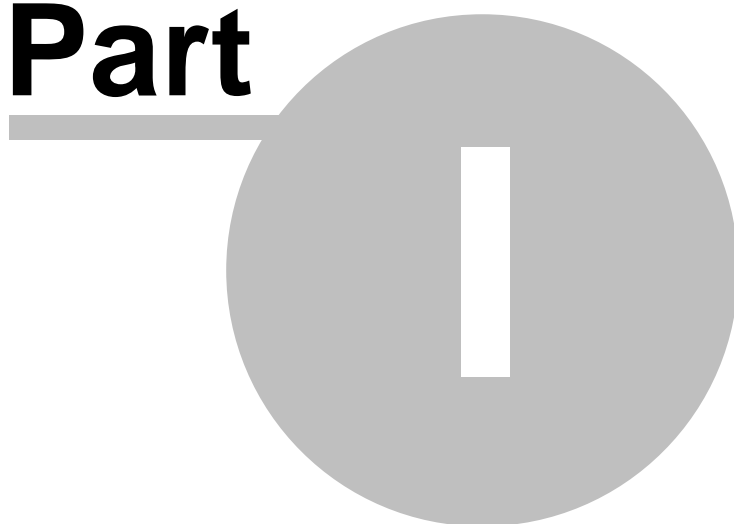
<b>Index</b>	<b>3174</b>
--------------	-------------



# Foreword

This user guide provides an introduction to the features contained in Enterprise Architect - a UML modeling tool for developing and building software systems with UML.

**Part**



# 1 Introduction



Sparx Systems **Enterprise Architect** is a full life-cycle **UML** based tool for:

- Designing and constructing a wide range of software systems
- Business analysis, business process modeling, managing requirements
- Systems modeling, systems architecture modeling, component design, simulation
- Generalized domain specific modeling
- Building domain specific modeling languages based on UML
- Visualizing a wide range of systems, processes, data, activities and structures
- Simulating behavioral processes, state machines and interactions
- Collaborating and sharing information and models
- Testing, quality control and end-to-end verification/traceability of complex systems
- Managing developmental tasks, work and scheduling

For over a decade, Sparx Systems has actively and continually developed Enterprise Architect based on the current UML standards. Over those years it has proven itself in a wide range of projects from small single user models to extremely large team based enterprise repositories. With over a quarter of a million licensed users in over 100 countries world wide, the power and effectiveness of Enterprise Architect has been convincingly demonstrated in many different areas.

## Starting Out

Where to Start	Link
A summary of <b>what Enterprise Architect can do</b> and what you can use it for.	<a href="#">Overview</a> <sup>5</sup> <a href="#">What You Can Do</a> <sup>7</sup> <a href="#">Key Benefits</a> <sup>9</sup> <a href="#">Key Features</a> <sup>13</sup>
A quick review of the process of modeling with Enterprise Architect. How to start Enterprise Architect and create a model project.	<a href="#">Getting Started</a> <sup>42</sup> <a href="#">A Quickstart Tutorial</a> <sup>47</sup>
The initial tutorial has many links to more extensive <b>descriptions of features, functions, tasks and procedures.</b>	<a href="#">User Interface Guide</a> <sup>67</sup>
The <b>Sparx Systems Formal Statements</b> includes Copyright Notices, Acknowledgements and the End User License Agreement.	<a href="#">Formal Statements</a> <sup>25</sup>

Where to Start	Link
A <b>glossary of terms</b> is included that you can refer to for definitions of various terms and concepts.	<a href="#">Glossary</a> <sup>3115</sup>
<b>How to give feedback</b> We value any suggestions, feedback and comments you might have regarding this product, documentation or install process. You can provide your feedback: <ul style="list-style-type: none"><li>• Using a fault (bug) report or feature request, or</li><li>• By email</li></ul>	<a href="http://www.sparxsystems.com/bug_report.htm">www.sparxsystems.com/bug_report.htm</a> <a href="http://www.sparxsystems.com/feature_request.htm">www.sparxsystems.com/feature_request.htm</a> <a href="mailto:support@sparxsystems.com">support@sparxsystems.com</a>
How to purchase Enterprise Architect and find pricing information.	<a href="#">Order Enterprise Architect</a> <sup>33</sup>

## 1.1 Overview

### Enterprise Architect

- Is a comprehensive **UML** analysis and design tool.
- Covers all aspects of **software**, **business** and **systems** modeling and design
- Is suitable for **requirements gathering** through analysis, design, construction, build, debug, simulation, testing, change management and maintenance to implementation, with full **traceability**.
- Combines the power of the latest UML specification ([www.omg.org](http://www.omg.org)) with a high performance, intuitive interface, to bring an integrated and advanced toolset to the whole development team.
- Is a **scalable**, multi-user, visual tool with a rich feature set
- Helps build and document robust, maintainable systems and processes.
- Is a **proven**, highly popular tool for analysts, developers, consultants and managers in over 130 countries.
- Is used in the development of many kinds of application and system in a wide range of industries
- Covers **multiple domains**, including: aerospace, banking, web development, engineering, finance, medicine, military, research, academia, transport, retail, utilities and electrical engineering. It also has a long history of being used by standards organization worldwide to structure and organize their domain specific knowledge and communication channels.
- Is **supported** by many **partners**, **consultants**, colleges and other organizations who provide training and consulting services based around Enterprise Architect.

### How this Help is Organized

Topic	Description	Link
<b>Getting Started</b>	Some introductory tutorials and important information to get up and running quickly.	<a href="#">Getting Started</a> <sup>[42]</sup>
<b>Setting up and managing Projects</b>	Configuring, managing and maintaining <b>modeling projects</b> . Includes storage mechanisms, team development environments, managing changes and monitoring progress.	<a href="#">Projects and Teams</a> <sup>[197]</sup>
<b>Models and Modeling</b>	Includes how to work with the models and components, their properties, and the reference data with which you populate the model.	<a href="#">Modeling Basics</a> <sup>[750]</sup>
<b>UML Concepts and Language</b>	Basic UML and open standards based extensions. How those concepts are <b>extended</b> in Enterprise Architect's support of other modeling languages, and how you can create your own modeling languages.	<a href="#">Define a Modeling Language</a> <sup>[1483]</sup>
<b>Navigating and Searching</b>	How to find the information you require in the model. Selecting aspects of the model to extract and review.  <b>Tracing</b> developmental relationships in both the structure and the development timeframe.	<a href="#">Navigate, Search &amp; Trace</a> <sup>[644]</sup>

Topic	Description	Link
<b>Transforming</b>	Transforming model elements and fragments from one domain to another, using standard and customized Model Driven Architecture transformations.	<a href="#">Model Transformation</a> <sup>[2013]</sup>
<b>Generating Code</b>	Forward engineering from models and importing code (reverse engineering). Synchronizing code and model.  <b>Database</b> engineering is also supported, including forward engineering and import of legacy database structures.	<a href="#">Software Engineering</a> <sup>[2073]</sup> <a href="#">Database Engineering</a> <sup>[2334]</sup> <a href="#">SOA and XML</a> <sup>[2386]</sup>
<b>Visual Analysis of Executing code</b>	Advanced debugging and run-time inspection in a variety of languages. Tools for recording execution and stack traces and automatically generating Sequence diagrams. Profiling tools for native Windows applications. A powerful object workbench to test and work with individual objects. A rich testing capability featuring custom Testpoints based on Programming by Contract ideas and using a "code free" xUnit like approach.	<a href="#">Visual Execution Analyzer</a> <sup>[2527]</sup> <a href="#">Profiling</a> <sup>[2555]</sup> <a href="#">Object Workbench</a> <sup>[2567]</sup> <a href="#">Testpoint Management</a> <sup>[2577]</sup> <a href="#">Unit Testing</a> <sup>[2573]</sup>
<b>Simulations</b>	Run model <b>Simulations</b> on behavioral diagrams. Extensive support for State machines, Activity graphs, Interactions and BPMN. A powerful and easy to use feature that brings your model to life and helps ensure the validity and behavior of your solution.	<a href="#">Model Simulation</a> <sup>[2463]</sup>
<b>Domain Specific Modeling</b>	Developing models for <b>specific modeling domains</b> .	<a href="#">Analysis and Business Modeling</a> <sup>[1804]</sup> <a href="#">Database Engineering</a> <sup>[2334]</sup> <a href="#">Systems Engineering</a> <sup>[2288]</sup> <a href="#">SOA and XML Engineering</a> <sup>[2386]</sup>
<b>Quality Control</b>	Ensuring <b>quality control</b> of your models and code through <b>model validation</b> and running your own <b>test scripts</b> .	<a href="#">Testing</a> <sup>[2593]</sup>
<b>Maintenance</b>	Managing the <b>maintenance</b> of model elements and <b>changes and issues</b> across the project.	<a href="#">Maintenance</a> <sup>[2619]</sup>
<b>Reporting</b>	<b>Documenting</b> your model in either document report or web output format.	<a href="#">Reporting</a> <sup>[2638]</sup>

Topic	Description	Link
<b>Automation and Scripting</b>	How to use the ActiveX automation interface provided by Enterprise Architect to script and extend the capabilities of the tool. How to write Add-ins and how to use the scripting capabilities.	<a href="#">Automation and Scripting</a> <sup>[2813]</sup> <a href="#">Scripting</a> <sup>[2791]</sup> <a href="#">Enterprise Architect Add-In Model</a> <sup>[3010]</sup>
<b>Licensing</b>	Managing your private or shared <b>Enterprise Architect product license keys</b> .	<a href="#">License Management</a> <sup>[3162]</sup>

### 1.1.1 What You Can Do

Enterprise Architect is a rich and diverse modeling environment. At the highest level it uses a wide range of tools to help manage, understand, visualize, explore, trace and work with complex systems.

- General analysis, design, construction, visualization, testing, simulation and management of many domains
- Software construction, analysis, management, testing, coding, debugging and visualizing
- Business analysis, process definition, requirements management, process simulation, traceability
- Systems analysis, design, simulation, engineering and verification

#### A summary of some of the main activities supported by Enterprise Architect

Topic	Detail	See also
<b>Modeling Analysis and Design Requirements</b>	To <b>create models</b> with Enterprise Architect, you should become familiar with: <ul style="list-style-type: none"> <li>• The UML standards and</li> <li>• How you apply UML in Enterprise Architect to develop your models</li> <li>• UML extensions such as SysML, BPMN, DDS and others</li> <li>• Domain based modeling</li> <li>• Model Transformations and MDA</li> </ul>	<a href="#">Modeling Fundamentals</a> <sup>[750]</sup> <a href="#">Standard UML Models</a> <sup>[1179]</sup> <a href="#">Requirement Models</a> <sup>[1726]</sup> <a href="#">Domain Based Models</a> <sup>[1789]</sup> <a href="#">Model Transformation</a> <sup>[2013]</sup>
<b>Managing Models</b>	To <b>manage</b> the models in your projects you: <ul style="list-style-type: none"> <li>• Create and structure models</li> <li>• Work with elements within packages and views</li> <li>• Create diagrams as visual representation of model elements</li> <li>• Import and export model information</li> <li>• Protect and manage the model data itself</li> </ul>	<a href="#">Projects and Teams</a> <sup>[197]</sup> <a href="#">Team Development</a> <sup>[306]</sup> <a href="#">Reporting</a> <sup>[2636]</sup> <a href="#">Change Management</a> <sup>[382]</sup>

	<ul style="list-style-type: none"> <li>Communicate information on the data in the form of online, printable and web-page documentation and reports, and</li> <li>Manage change with baselines, XML import and export and auditing</li> </ul>	
<b>Code Engineering</b>	<p>In Enterprise Architect, UML modeling both depends on and supports code engineering - you <b>generate</b> and <b>update code</b> from a model, and you <b>create and update models</b> from code</p> <p>In this broad sense, Enterprise Architect enables you to:</p> <ul style="list-style-type: none"> <li>Forward engineer, reverse engineer, round-trip and synchronize code in a range of programming languages</li> <li>Debug and profile code</li> <li>Model and generate code for XML Technologies</li> <li>Perform database modeling and design for a range of database management systems</li> <li>Convert model components from one domain to another using Model Driven Architecture (<b>MDA</b>) <b>Transformations</b></li> </ul>	<a href="#">Overview Of Development</a> <sup>[2076]</sup>
<b>Model Simulation</b>	<p><b>Model Simulation</b> brings your <b>behavioral models to life</b> with instant, real-time <b>behavioral model execution</b>. Coupled with tools to manage triggers, events, guards, effects, breakpoints and simulation variables, plus the ability to <b>visually track execution at run-time</b>, the Simulator is a powerful means verifying the correctness of your behavioral models.</p>	<a href="#">Model Simulation</a> <sup>[2463]</sup>
<b>Managing Projects</b>	<p>Enterprise Architect provides strong support for Project Management, particularly in the following areas:</p> <ul style="list-style-type: none"> <li>Project estimation - working out how much time and effort is required to build and deploy a solution, using the Use Case metrics facility and carefully-calibrated metrics</li> <li>Defining, assigning and managing resources</li> <li>Communicating Project Management discussions and decisions to the project</li> <li>Monitoring and managing problems, changes, issues and tasks that affect both individual elements and the project as a whole</li> <li>Managing the development, execution and results of testing, from Integration through to User Acceptance, and</li> <li>Maintaining a project glossary of terms, procedures and policies applied to the project</li> </ul> <p>The scope of your project management might include upgrades to Enterprise Architect and installation of related technologies</p>	<a href="#">Projects and Teams</a> <sup>[197]</sup> <a href="#">Testing</a> <sup>[2593]</sup> <a href="#">Maintenance</a> <sup>[2619]</sup> <a href="#">License Management</a> <sup>[3162]</sup>



<b>Extending Enterprise Architect Facilities</b>	<p>Experienced technology developers can develop customized additions to the functionality already present within Enterprise Architect. These additions include:</p> <ul style="list-style-type: none"> <li>• UML Profiles and Stereotypes</li> <li>• UML Patterns</li> <li>• Code Templates</li> <li>• Tagged Value Types</li> <li>• MDG Technologies</li> <li>• Enterprise Architect Add-Ins</li> <li>• Scripts</li> </ul> <p>By creating these extensions the Technology Developer can customize the Enterprise Architect modeling process to specific tasks and speed up development.</p>	<p><a href="#">Extending UML Models</a> <sup>[147]</sup></p> <p><a href="#">Automation and Scripting</a> <sup>[2813]</sup></p>
--	---	--

### 1.1.2 Key Benefits

Enterprise Architect is a powerful tool for specifying, documenting and building your software and business process projects. Using Enterprise Architect's **support for UML** and its related standards, you can model new complex software and business systems, or visualize and maintain existing systems.

#### Key Benefits and Usage

Topic	Detail	See also
<b>Design and Build Diverse Systems Using UML</b>	<ul style="list-style-type: none"> <li>• UML is an open standard that provides a rich language for describing, documenting and designing software, business and IT systems in general.</li> <li>• Enterprise Architect enables you to leverage the full expressive power of UML 2.4.1 to model, design and build diverse systems in an open and well understood manner.</li> <li>• You can generate code, database structures, documentation and metrics; transform models; or specify behavior and structure as the basis for contractual agreements.</li> </ul>	<p><a href="#">Standard UML Models</a> <sup>[1179]</sup></p> <p><a href="#">Modeling Basics</a> <sup>[750]</sup></p>
<b>Model and Manage Complexity</b>	<ul style="list-style-type: none"> <li>• Enterprise Architect helps individuals, groups and large organizations model and manage complex information.</li> <li>• Often this relates to software development and IT systems design and deployment, but it can also relate to business analysis and business process modeling.</li> <li>• Enterprise Architect integrates and connects a wide range of structural and behavioral information, helping to build a coherent and verifiable architectural model, either what-is or what-will-be.</li> <li>• Tools to manage version control, track and compare differences, audit changes and enforce security, help control</li> </ul>	<p><a href="#">User Security</a> <sup>[316]</sup></p> <p><a href="#">Change Management</a> <sup>[382]</sup></p>

Topic	Detail	See also
	project development and enforce compliance to standards.	
<b>Structured Use Case Scenarios</b>	<ul style="list-style-type: none"> <li>Enterprise Architect's Structured Scenario editor enables you to develop structured Use Case Scenarios, to capture vital analysis information in the form of natural language descriptions.</li> <li>The editor helps you use this information to drive downstream development and maximize traceability across the development life-cycle.</li> <li>The editor also helps you to dynamically link scenario steps to associated model elements, such as domain elements, business rules and glossary terms.</li> <li>From structured scenarios, you can automatically generate test case descriptions, and Activity and other UML behavioral diagrams.</li> <li>You can even reverse engineer existing process diagrams into structured, textual specifications to produce documentation deliverables.</li> </ul>	<a href="#">Scenarios</a> <sup>[96]</sup>
<b>Share Models</b>	<ul style="list-style-type: none"> <li>Enterprise Architect enables you to share complete models or specific aspects of a model between members of a team, including (through the 'Lite', read-only edition) stakeholders who can study a model but not change or manage it.</li> <li>You can make your project .EAP file available on a shared network drive, or replicate the .EAP file for complex distributed development.</li> <li>Alternatively, you can develop the project in one of several shared DBMS repositories.</li> <li>You can import and export data as XML files to distribute and update frameworks and other package-based model structures; you control changes through the version control repository.</li> <li>Enterprise Architect provides a data transfer wizard that enables you to upsize or downsize the complete model for maximum flexibility, and it enables you to export and import reference data so that you do not have to recreate it for related projects.</li> </ul>	<a href="#">The Read-only 'Lite' Edition</a> <sup>[23]</sup> <a href="#">Project Sharing</a> <sup>[307]</sup> <a href="#">Server Based Repositories</a> <sup>[214]</sup> <a href="#">XML Import and Export</a> <sup>[473]</sup> <a href="#">Version Control</a> <sup>[383]</sup> <a href="#">Perform a Project Data Transfer</a> <sup>[504]</sup> <a href="#">Sharing Reference Data</a> <sup>[374]</sup>
<b>Model, Manage and Trace Requirements</b>	<ul style="list-style-type: none"> <li>Enterprise Architect enables you to capture requirements and use full traceability from base requirements to design, build, deployment and beyond.</li> <li>You can use impact analysis to trace from proposed changes to original requirements, and build the 'right' system.</li> </ul>	<a href="#">Requirements</a> <sup>[1726]</sup> <a href="#">Trace: Tracking Dependencies</a> <sup>[723]</sup>
<b>Develop Personal Views and</b>	<ul style="list-style-type: none"> <li>Enterprise Architect enables you to develop any number of different views of your model, or parts of it, either for your personal use or for the use of your team.</li> </ul>	<a href="#">Model Views</a> <sup>[686]</sup>

Topic	Detail	See also
<b>Extracts of the Model</b>	<ul style="list-style-type: none"> <li>These Model Views are generated by reports, so they can be set up to always show the current status of the selected view.</li> <li>The facility also enables you to create Favorites folders of hyperlinks to frequently-used data structures.</li> </ul>	
<b>Track and Trace Model Structures</b>	<ul style="list-style-type: none"> <li>In even a small model, it can be difficult to locate specific packages, diagrams or elements, even if you apply a rigorous naming and structure policy.</li> <li>Enterprise Architect has a wealth of facilities that enable you to locate structures quickly and easily, through the Model Search, Package Browser, Diagram List, Auditing facility, Traceability window, Relationship Matrix and reports.</li> <li>The Element menu, Diagram menu and Project Browser context menus also enable you to locate elements in diagrams and in the Project Browser, and you can store hyperlinks to important or commonly-used elements and diagrams in the Model Views.</li> <li>Finally, having located one element you can import any related elements into a diagram in a single operation.</li> </ul>	<a href="#">Navigate, Search &amp; Trace</a> <sup>[644]</sup> <a href="#">Auditing</a> <sup>[446]</sup> <a href="#">System Documents</a> <sup>[2736]</sup> <a href="#">Main Menu</a> <sup>[77]</sup> <a href="#">Project Browser Context Menus</a> <sup>[648]</sup> <a href="#">Insert Related Elements</a> <sup>[933]</sup>
<b>Generate Documentation</b>	<ul style="list-style-type: none"> <li>Enterprise Architect provides powerful document generation and reporting tools with a full WYSIWYG template editor for document or web page output.</li> <li>You can generate complex and detailed reports from Enterprise Architect with the information you require in the format your company or client demands.</li> </ul>	<a href="#">Reporting</a> <sup>[2638]</sup>
<b>Generate and Reverse Engineer Source Code</b>	<ul style="list-style-type: none"> <li>Enterprise Architect supports generation and reverse engineering of source code for many popular languages.</li> <li>With a built in 'syntax highlighting' source code editor, Enterprise Architect enables you to quickly navigate and explore your model source code in the same environment.</li> <li>Code generation templates enable you to customize the generated source code to your company specifications.</li> </ul>	<a href="#">Generate Source Code</a> <sup>[2111]</sup> <a href="#">Importing Source Code</a> <sup>[2136]</sup> <a href="#">Editing Source Code</a> <sup>[2146]</sup> <a href="#">Code and Transform Templates</a> <sup>[1632]</sup>
<b>Visualize, Inspect and Understand Complex Software</b>	<ul style="list-style-type: none"> <li>Software is complex and often hard to understand; you can use Enterprise Architect to reverse engineer code in a wide range of software development languages and database repository schema, to understand static structure.</li> <li>To complete the picture, use the unique built-in profiling and debugging tools to capture and visualize executing software at run-time.</li> <li>Create run-time instances of model elements and invoke methods using the built in Object Workbench.</li> <li>You can also bring in complete frameworks from source code or Java .jar files - or even .Net binary assemblies!</li> </ul>	<a href="#">Importing Source Code</a> <sup>[2136]</sup> <a href="#">Execution Analysis</a> <sup>[2527]</sup> <a href="#">Object Workbench</a> <sup>[2567]</sup> <a href="#">Import Source Code</a> <sup>[2136]</sup> <a href="#">Import Binary Module</a> <sup>[2143]</sup>

Topic	Detail	See also
	<ul style="list-style-type: none"> <li>By importing frameworks and library code, you can maximize re-use and understanding of your existing investment.</li> </ul>	
<b>Use Model Driven Architecture and Transformations</b>	<ul style="list-style-type: none"> <li>Model Driven Architecture (MDA) is an open standard designed to facilitate rapid application development in a platform independent manner.</li> <li>Models can be built at a high level of abstraction and, using MDA based tools, transformed into models and code targeting a specific platform or domain.</li> <li>Enterprise Architect supports advanced MDA transformations using easily edited and developed transformation templates.</li> <li>With built-in transformations for DDL, C#, Java, EJB and XSD, you can quickly develop complex solutions from simple platform independent models (PIMs) targeted at platform specific models (PSMs); one PIM can be used to generate and synchronize multiple PSMs, providing a significant productivity boost.</li> </ul>	<a href="#">Model Transformation</a> <sup>[2013]</sup> <a href="#">Built-in Transformations</a> <sup>[2020]</sup>
<b>SOA (Service Oriented Architecture) Support</b>	<p>Enterprise Architect enables you to rapidly model and forward- and reverse-engineer two key W3C XML technologies:</p> <ul style="list-style-type: none"> <li>XML Schema (XSD) and</li> <li>Web Service Definition Language (WSDL)</li> </ul> <p>XSD and WSDL support is critical for the development of a complete Service Oriented Architecture (SOA), and the coupling of UML 2.4.1 and XML provides the natural mechanism for specifying, constructing and deploying XML-based SOA artifacts within an organization.</p>	<a href="#">XML Schema - XSD</a> <sup>[2387]</sup> <a href="#">Web Services - WSDL</a> <sup>[2423]</sup>
<b>Systems Engineering support</b>	<p>Integrating many high-end features for Systems Engineers, the Ultimate and Systems Engineering editions of Enterprise Architect provide built-in support for:</p> <ul style="list-style-type: none"> <li>SysML</li> <li>Parametric model simulation</li> <li>Executable code generation</li> <li>Model to code transformations for Hardware Description Languages and Ada 2005</li> </ul>	<a href="#">Editions Available</a> <sup>[20]</sup> <a href="#">SysML</a> <sup>[2294]</sup> <a href="#">Behavior</a> <sup>[1014]</sup> <a href="#">State Machine Modeling For HDLs</a> <sup>[2137]</sup> <a href="#">Ada 2005 Conventions</a> <sup>[2082]</sup>
<b>Model Databases</b>	<ul style="list-style-type: none"> <li>Enterprise Architect enables you to reverse engineer from many popular DBMS systems.</li> <li>You can model database tables, columns, keys, foreign keys and complex relationships using UML and an inbuilt data modeling profile, and forward generate DDL scripts to create target database structures.</li> </ul>	<a href="#">Supported Databases</a> <sup>[2375]</sup> <a href="#">Data Models</a> <sup>[1937]</sup>

Topic	Detail	See also
<b>Customize Enterprise Architect</b>	<ul style="list-style-type: none"> <li>Enterprise Architect also includes facilities that enable experienced tool developers to customize and extend Enterprise Architect to suit the specific requirements of their organization with, for example, in-house UML Profiles, Add-Ins and Code Templates.</li> <li>The very detailed Automation Interface gives you access to most element features, major functions such as XML import/export, and attached information; most properties are fully writable from the automation client.</li> <li>The Automation Interface provides great support for plug-ins, with the ability to embed automation client windows in the main diagram view.</li> <li>The Interface is accessible from any automation-aware client language, such as VB, C#, C++ and Delphi.</li> </ul>	<a href="#">Extending UML Models</a> <sup>[147]</sup> <a href="#">Enterprise Architect Add-In Model</a> <sup>[8010]</sup> <a href="#">Code Template Framework</a> <sup>[163]</sup> <a href="#">Using the Automation Interface</a> <sup>[2805]</sup>
<b>Link Enterprise Architect to IDEs</b>	<ul style="list-style-type: none"> <li>Using Sparx Systems Model Driven Generation (MDG) Link plug-ins, you can develop source code in your preferred Integrated Development Environment such as Visual Studio .NET or Eclipse, while you use Enterprise Architect to locate the source code for Classes, attributes and operations, and to model, navigate, track, reverse engineer, build and run your project.</li> <li>The MDG Integration products for Eclipse and Visual Studio provide an even closer, seamless integration of Enterprise Architect and UML 2.4.1 with your IDE, bringing the functionality required of a fully fledged modeling platform right inside the IDE.</li> </ul>	<a href="#">Visual Studio .NET Eclipse (Link)</a> <a href="#">Eclipse (Integration)</a> <a href="#">Visual Studio 2008</a>

### 1.1.3 Key Features

Enterprise Architect has a rich and advanced set of tools for conducting a wide range of modeling, analysis, requirements, design, construction and visualization tasks. The table below lists some of the more important capabilities and how they assist you in your modeling and design tasks.

#### Key Features

Feature	Facilities
<b>Modeling</b>	<ul style="list-style-type: none"> <li>Model information, software, business rules and processes, hardware systems and more using UML compliant notation (for the latest UML specification, see <a href="http://www.omg.org">www.omg.org</a>)</li> <li>Leverage SysML, BPMN, DDS and other domain specific UML extensions</li> <li>Model dependencies between elements, system dynamics and states</li> <li>Model class hierarchies, deployment, components and implementation details</li> <li>Save and load complete diagrams as UML Patterns</li> <li>Create and share dynamic views of model elements and diagram sets using Model Views</li> <li>Analyze and trace relationships between elements using the tabular</li> </ul>

Feature	Facilities
	<p>Relationship Matrix</p> <ul style="list-style-type: none"> <li>Specify, trace and verify requirements directly against the design, right through to the deployed solution using built-in Requirements Management</li> <li>Apply and manage Traceability, Auditing and Impact Analysis</li> </ul>
<b>Projects</b> <b>Sharing</b> <b>Import/Export</b>	<ul style="list-style-type: none"> <li>Simple file based repositories for small teams</li> <li>Scalable large team support using shared repositories ( MS SQL Server, MySQL, Oracle and others supported )</li> <li>Record project issues, tasks and system glossary</li> <li>Assign resources to model elements and track effort expended against required effort</li> <li>Manage personal tasks, work flow and collaborative model based mail</li> <li>Schedule tasks and track important events using built in calendars and Gantt charts</li> <li>Distributed development through shareable files, use of shared repositories in a range of major Database Management Systems, file replication, data transfer, and import and export of reference data</li> <li>Share models using the latest XML 2.1 format</li> <li>Import models in XML format from other tools (where compliant)</li> <li>Audit all changes and report on recent activity</li> <li>Visual and textual differencing based on package level baselines</li> </ul>
<b>Business</b>	<ul style="list-style-type: none"> <li>Model and simulate processes with BPMN</li> <li>Use structured scenarios to rapidly define use case based behavior</li> <li>Import plain text use case descriptions into structured scenarios</li> <li>Generate diagrams from structured scenarios</li> <li>Generate BPEL scripts automatically from Business Process models</li> <li>Business rules editor/designer</li> <li>Generate executable business logic from rule tasks and trace to natural language business rules</li> </ul>
<b>Database Modeling</b>	<ul style="list-style-type: none"> <li>Database modeling profile</li> <li>Reverse engineer from a range of DBMSs via ODBC</li> <li>Forward engineer DDL scripts to create database structures</li> <li>Import database schema, XSD and WSDL source</li> <li>Support for GML and ArcGIS</li> </ul>
<b>Software</b>	<ul style="list-style-type: none"> <li>Support for forward and reverse code engineering for many languages including: <ul style="list-style-type: none"> <li><i>ActionScript 3.0</i></li> <li><i>Java</i></li> </ul> </li> </ul>

Feature	Facilities
	<ul style="list-style-type: none"> <li>• <i>C#</i></li> <li>• <i>C++</i></li> <li>• <i>VB.Net</i></li> <li>• <i>Delphi</i></li> <li>• <i>Visual Basic</i></li> <li>• <i>Python</i></li> <li>• <i>PHP</i></li> <li>• Import .NET and Java binaries</li> </ul>
<b>Systems</b>	<ul style="list-style-type: none"> <li>• Simulate SysML parametric models</li> <li>• Simulate state and activity diagrams</li> <li>• Verilog,</li> <li>• VHDL</li> <li>• SystemC</li> </ul>
<b>Visual Execution Analyzer</b> <b>Debuggers</b> <b>Build</b> <b>Testing</b>	<p>The Visual Execution Analyzer contains many important tools for building, verifying and testing your software. Some of the standout features include:</p> <ul style="list-style-type: none"> <li>• Debug numerous languages, including C++, Java, C#, VB.net, PHP and others</li> <li>• Integrate coding and development with the UML model</li> <li>• Quickly search through source files and through your model during coding</li> <li>• Intelli-sense based on model data</li> <li>• Record and generate sequence diagrams from execution traces</li> <li>• Use profiling tools for native windows applications written in C/C++</li> <li>• Use the Object Workbench to test individual objects</li> <li>• Testpoint facility to quickly apply xUnit like tests to source without writing code</li> <li>• Generate Testpoint diagrams and sets from execution traces</li> </ul>
<b>Simulation</b>	<p>Simulation of behavioral models provides numerous benefits, including:</p> <ul style="list-style-type: none"> <li>• Verify the correct behavior of modeled state machines and activity graphs</li> <li>• Demonstrate modeled solutions to customers with attractive, configurable simulation walk-throughs</li> <li>• Simulate buttons presses and real-world events with triggers</li> <li>• Simulate rules and conditional execution with both model elements and Javascript based code</li> <li>• Record sets of triggers and automate playback based on scenarios</li> <li>• Walk through sequence diagrams</li> </ul>
<b>Domain Specific modeling and</b>	<p>Modeling languages and frameworks for specialized domains are easy to build and use within Enterprise Architect. Several are supplied by Sparx Systems, and some additional ones by 3rd parties. The tools available in Enterprise Architect also let you build your</p>

Feature	Facilities
<b>Frameworks</b>	<p>own frameworks for your special domain.</p> <ul style="list-style-type: none"> <li>• Leverage industry-standard Enterprise Architecture frameworks such as Zachman, TOGAF, UPDM and others</li> <li>• Extended modeling for Requirements, User Interface Design, Mind Mapping, Data Modeling</li> <li>• SysML, DDS</li> <li>• BPMN</li> <li>• Built-in Model Driven Architecture (MDA) Transformations</li> <li>• Create custom transforms</li> <li>• Use UML Profiles to for domain-specific modeling</li> <li>• A range of add-Ins to integrate and extend the facilities of Enterprise Architect</li> <li>• Write your own add-Ins</li> <li>• Transform behavioral models into executable source code for software and hardware description languages (HDLs) such as Verilog, VHDL, and SystemC</li> </ul>
<b>Reporting</b>	<ul style="list-style-type: none"> <li>• Comprehensive and flexible MS Word-compatible web and document report options</li> <li>• Read-only Viewer enables stakeholders to view but not change milestone deliverables</li> </ul>
<b>Manage Change</b>	<ul style="list-style-type: none"> <li>• Manage, track and control change using baseline model merge and auditing capabilities</li> <li>• Migrate changes across a distributed environment using Controlled XML Packages</li> <li>• Manage Version control though XML using SCC CVS and Subversion configurations</li> <li>• Inbuilt user and group security and access control management</li> <li>• Auditing</li> </ul>
<b>Testing</b>	<ul style="list-style-type: none"> <li>• Testing support for test cases, JUnit and NUnit</li> <li>• Integrated Debug Workbench for visualizing executable Java and .Net applications, instantiating run-time model objects and generating Sequence diagrams from a stack traces</li> <li>• Extensive Testpoint facility.</li> </ul>
<b>Automation and Scripting</b>	<ul style="list-style-type: none"> <li>• Script and automate common tasks using a detailed Automation Interface and Model Scripts</li> </ul>
<b>User Experience</b>	<ul style="list-style-type: none"> <li>• Speed: Enterprise Architect is quick to load and a spectacularly fast performer, even with large models</li> <li>• Scalability: Enterprise Architect supports single users and the development of small models, or many concurrent users developing extremely large models,</li> </ul>



Feature	Facilities
	<p>with equal ease</p> <ul style="list-style-type: none"><li>• Usability: many of our users agree, Enterprise Architect gets you started and productive quickly, with a rich user interface and the ability to create templates, model views and 'favorites' collections of commonly-used elements and diagrams.</li></ul>

### Notes

- For a complete list of the new features of the latest version of Enterprise Architect, click on the **Help | Read Me** menu option
- Enterprise Architect is available in six editions: **Ultimate, Business and Software Engineering, Systems Engineering, Corporate, Professional**, and **Desktop**, each of which offers a different range of features

### Learn more

- [Editions Available](#)<sup>[20]</sup>
- [Extensions - MDG Technologies](#)<sup>[148]</sup>

## 1.2 Enterprise Architect Editions



Enterprise Architect is available in a number of different editions, each tailored to support a particular business case

### Trial and Beyond

Topic	Detail	See also
<b>Try before Purchase</b>	Test the product in a number of configurations in the Trial Version	<a href="#">The Trial Version</a> <sup>[18]</sup>
<b>Different Editions</b>	Explore the specific features of the six work environment editions available	<a href="#">Editions Available</a> <sup>[20]</sup>
<b>The Lite Edition</b>	Use the free, read-only or 'Lite' edition	<a href="#">The Read-only 'Lite' Edition</a> <sup>[23]</sup>

### 1.2.1 The Trial Version

The fully functional 30 day trial version of Enterprise Architect is available free of charge on the Sparx Systems website [www.sparxsystems.com/bin/easetup.exe](http://www.sparxsystems.com/bin/easetup.exe). The trial version is identical to the registered edition with the exception that all diagrams are output to files with an embedded watermark.

### Trial Details

Option	Detail	See also
<b>Try Out Editions</b>	<p>You can use the trial version to explore and evaluate Enterprise Architect, checking the edition you are interested in and trying out other editions for comparison.</p> <p>When you start up the Enterprise Architect trial version, the Select Trial Version prompt displays; select the mode to trial.</p> <p>If necessary, you can close down Enterprise Architect and restart it in another mode for comparison.</p> <p>The prompt also directs you to useful information such as a walkthrough of the Enterprise Architect facilities, and options to select one of several workspace layouts.</p>	<a href="#">Manage Workspace Layout</a> <sup>[163]</sup>
<b>Trial Period</b>	As you are evaluating the Enterprise Architect trial version, note	

Option	Detail	See also
	<p>that the software operates for a limited period, and denies access after the trial period has elapsed.</p> <p>To continue using Enterprise Architect when the trial period expires, you can either:</p> <ul style="list-style-type: none"> <li>• Apply to extend the trial period, or</li> <li>• Purchase and register a full license; on purchase of a suitable license or licenses, the registered version is made available for download</li> </ul> <p>The latest information on pricing and purchasing is available on the Sparx Systems website; for more information, contact Sparx Systems by email.</p> <p>When you order and pay for an edition of Enterprise Architect, you receive installation instructions and the location of the executable files for download.</p>	<p><a href="#">Order Enterprise Architect</a><sup>[33]</sup></p> <p><a href="#">Installation</a><sup>[34]</sup></p> <p><a href="#">Register a Full License</a><sup>[36]</sup></p> <p><a href="mailto:sales@sparxsystems.com">sales@sparxsystems.com</a></p>
<b>Extend Trial Period</b>	<p>If you are testing the trial version and require more than 30 days to evaluate it, you can apply to Sparx Systems Sales for an extension of the trial period.</p> <p>The trial period must expire before you can enter the extension key.</p>	<p><a href="mailto:sales@sparxsystems.com">sales@sparxsystems.com</a></p>

#### Extend the trial period after receipt of extension key


Step	Action	See also
<b>1</b>	<p>Open the Enterprise Architect trial version.</p> <p>The Evaluation Version of Enterprise Architect dialog displays; once the trial period has expired, you cannot proceed beyond this dialog without extending the trial period.</p>	
<b>2</b>	<p>Press ( <b>Ctrl</b> ) whilst you click on the <b>Continue Trial</b> button.</p> <p>The Upgrade Key dialog displays.</p>	
<b>3</b>	<p>In the <b>Upgrade Key</b> field, type or copy-and-paste the extension key you received from Sparx Systems Sales.</p>	
<b>4</b>	<p>Click on the <b>OK</b> button.</p> <p>Enterprise Architect confirms that your trial period has been extended.</p> <p>Your trial period is extended by the period of days stated in the email from Sparx Systems Sales; you can now restart and continue to use the Trial version of Enterprise Architect.</p>	


Step	Action	See also



### 1.2.2 Editions Available



Enterprise Architect is available in six editions: the **Ultimate**, **Business and Software Engineering**, and **Systems Engineering** 'suite' editions, and the **Corporate**, **Professional** and **Desktop** editions.

#### Edition Comparison

Topic	Detail	See also
<b>Comparison</b>	<p>The functionality for each edition is described below; the features of and differences between the editions are listed in the table provided on the Sparx Systems website.</p> <p>To help you understand the differences between these editions and the advantages and limitations of each, you can open and use the Trial version of Enterprise Architect to mimic any of the editions.</p>	<a href="#">Sparx Systems website</a> <a href="#">The Trial Version</a> <sup>[18]</sup>
<b>Ultimate Edition</b> 	<p>The Ultimate edition is designed for power users and those working across multiple domains, providing deep support for Business, for Software Engineering and for Systems Development seamlessly integrated into a single development environment.</p> <p>It enables you to drill down to the lowest levels of systems design and construction, with SysML and executable code generation for standard and hardware description languages.</p> <p>Business users can leverage BPEL, the Rules Composer and executable UML, in addition to all the advanced features of the other editions of Enterprise Architect.</p> <p>Software developers can integrate their Eclipse and Visual Studio projects with their UML models and leverage the advanced executable code generators to target different domains.</p> <p>The Ultimate edition enables end to end traceability throughout a global vision of your enterprise - unifying strategy, business process, interfaces, software, rules, data and fine grained systems; powerful tools, domain-specific technologies, frameworks, integration platforms and a consistent, scalable, and robust interface work in unison to help you deliver on the promise of Model Driven Development.</p> <p>The Ultimate edition incorporates a number of MDG Technologies and Add-Ins; the Ultimate edition and MDG Technologies are all available in either Fixed License or Floating License form.</p> <p>The Floating License arrangement is particularly useful for companies that manage a central store of license keys, which can be used by different employees over time, temporarily or permanently.</p>	<a href="#">MDG Technologies</a> <sup>[148]</sup>

Topic	Detail	See also
	<p>The Ultimate edition provides:</p> <ul style="list-style-type: none"> <li>• Executable Code Generation - support for generating functional source code for State Machines, Interactions and Activities in C, C++, C#, Java and VB.NET</li> <li>• Full round trip support for Hardware Description Languages (Verilog, VHDL and SystemC) including support for generating State Machine code</li> <li>• SysML Simulation Support - including support for simulating SysML 1.3 constraint models with results graphing capabilities</li> <li>• BPEL Generation – transform BPMN 1.1 and BPMN 2.0 Business process models down to BPEL 1.1 and BPEL 2.0 code respectively</li> <li>• Business Rules – trace from abstract business rules down to automatically generated behavioral code</li> </ul>	
<p><b>Business and Software Engineering Edition</b></p> 	<p>The Business and Software Engineering edition is aimed at software development professionals, business modelers, architects, requirements experts, project managers and others involved in the design and construction of quality software and business services.</p> <p>It combines powerful new features such as executable code generation from UML models, BPEL, advanced scripting and a multi-purpose Rules Composer targeting executable code from Business Domain models; it also bundles licenses for integration products and frameworks such as UPDM (DoDAF-MODAF), TOGAF and Zachman, to provide advanced model-driven construction tools to tightly bind your code development in Eclipse or Visual Studio.</p> <p>The Business and Software Engineering edition incorporates a number of MDG Technologies and Add-Ins; the Business and Software Engineering edition and MDG Technologies are all available in either Fixed License or Floating License form.</p> <p>The Floating License arrangement is particularly useful for companies that manage a central store of license keys, which can be used by different employees over time, temporarily or permanently.</p> <p>The Business and Software Engineering edition provides:</p> <ul style="list-style-type: none"> <li>• Generation of Behavioral Code from State, Sequence and Activity models, supporting standard programming languages such as Java and .NET</li> <li>• Advanced math functions within the scripting engine</li> <li>• BPEL Generation from BPMN 1.1 and BPMN 2.0 models - including validation and WSDL support</li> <li>• A Business Rules Composer that enables you to build Business Domain models and generate code to implement complex business rules in standard programming languages</li> </ul>	<p><a href="#">MDG Technologies</a></p>


Topic	Detail	See also
<b>Systems Engineering Edition</b> 	<p>The Systems Engineering edition is designed for systems and software development professionals working on real-time, embedded and systems solutions.</p> <p>It combines new features such as executable code generation from UML models (including support for hardware languages such as Verilog and VHDL), Ada, SysML 1.3, executable SysML Parametric diagrams and advanced scripting.</p> <p>It also bundles licenses for UPDM (DoDAF-MODAF), SysML, DDS and integration products to provide powerful model-driven construction tools for the Systems Engineering domain to tightly bind your code development in Eclipse or Visual Studio with the UML/SysML models developed in Enterprise Architect.</p> <p>The Systems Engineering edition incorporates a number of MDG Technologies and Add-Ins; the Systems Engineering edition and MDG Technologies are all available in either Fixed License or Floating License form.</p> <p>The Floating License arrangement is particularly useful for companies that manage a central store of license keys, which can be used by different employees over time, temporarily or permanently.</p> <p>This edition provides:</p> <ul style="list-style-type: none"> <li>• Executable Code Generation - support for generating functional source code for State Machines, Interactions and Activities in C, C++, C#, Java and VB.NET</li> <li>• Full round trip support for Hardware Description Languages, including Verilog, VHDL and SystemC, with support for generating State Machine code</li> <li>• SysML Simulation Support - Includes support for simulating SysML 1.3 constraint models with results graphing capabilities</li> </ul>	<a href="#">MDG Technologies</a> <small>1481</small>
<b>Corporate Edition</b> 	<p>Aimed at larger development teams, the Corporate edition enables you to connect to the following DBMS back ends as the shared repository:</p> <ul style="list-style-type: none"> <li>• MySQL</li> <li>• SQL Server</li> <li>• PostgreSQL,</li> <li>• Sybase Adaptive Server Anywhere</li> <li>• Access 2007 and</li> <li>• Oracle 9i, 10g or 11g</li> </ul> <p>This provides additional scalability and improved concurrency over the shared .EAP file approach to model sharing.</p> <p>User security, user logins, user groups and user level locking of elements, user/group based security (with locking at diagram and element levels) are also supported.</p> <p>Security comes in two modes: in the first mode, all elements are</p>	

Topic	Detail	See also
	<p>considered 'writeable' until explicitly locked by a user or group; in the second mode, all elements are considered locked until checked out with a user lock.</p> <p>The Corporate edition forms the base for the three extended editions described above; like those editions, it is available in either Fixed License or Floating License form.</p> <p>The Floating License arrangement is particularly useful for companies that manage a central store of license keys, which can be used by different employees over time, temporarily or permanently.</p>	
<b>Professional Edition</b> 	<p>Aimed at work groups and developers, the Professional edition supports shared projects through replication and shared network files.</p> <p>This edition has an ActiveX interface for interrogating Enterprise Architect projects and extracting information in XML format.</p> <p>The Professional edition fully supports code import/export and synchronization of model elements with source code; it enables reverse engineering of SQL Server; MS Access 97, 2000 and 2003; and Oracle 9i, 10g or 11g databases.</p> <p>Support for MDG Technologies and MDG Link (sold separately) is included with the Professional version.</p> <p>The shared repository available in the Professional edition is restricted to the .EAP file format (JET database).</p>	
<b>Desktop Edition</b> 	<p>The Desktop edition is targeted at single analysts and developers producing UML analysis and design models.</p> <p>It provides facilities for:</p> <ul style="list-style-type: none"> <li>• UML modeling</li> <li>• XML import/export</li> <li>• Document generation</li> <li>• Version control integration and</li> <li>• Profile/metamodel extensibility</li> </ul>	

### 1.2.3 The Read-only 'Lite' Edition

*Enterprise Architect Lite* is a **free**, read-only edition of Enterprise Architect that enables people such as project sponsors to review the project without making any changes.

#### The read-only Lite Version

Topic	Detail	See also
<b>Further Access</b>	Users of Enterprise Architect Lite also have wider access to	<a href="#">Team Review Tools</a> 

Topic	Detail	See also
	<ul style="list-style-type: none"> <li>The Team Review, where readers can create and respond to posts, and link their comments to elements</li> <li>The Source Code Editor, where readers can open and edit external source code files, debug code, and configure and run Analyzer scripts</li> <li>The <b>File</b> menu, where readers can copy the project or create a shortcut to access it</li> <li>The Relationship Matrix, where readers can export the matrix contents to a CSV file to be opened in a spreadsheet application</li> <li>The Default Hours tab to review project metrics.</li> </ul> <p>You can download the Enterprise Architect Lite edition (as the Enterprise Architect Viewer) from the Sparx Systems website.</p>	<a href="#">Editing Source Code</a> <sup>[2146]</sup> <a href="#">File Menu</a> <sup>[79]</sup> <a href="#">Relationship Matrix Options</a> <sup>[738]</sup> <a href="#">Default Hours</a> <sup>[588]</sup> <a href="http://www.sparxsystems.com/products/ea/downloads.html">http://www.sparxsystems.com/products/ea/downloads.html</a> .
<b>Other Read-Only Options</b>	<p>You can also make your model available to others in a read-only format by:</p> <ul style="list-style-type: none"> <li>Generating an HTML report on the model, which can be published on the web with read-only access</li> </ul>	<a href="#">HTML Reports</a> <sup>[2744]</sup>



## 1.3 Formal Statements



Please take the time to read the following legal statements concerning Sparx Systems Enterprise Architect:

### Legal Notes

Topic	Link
Software Copyright Notice	<a href="#">Software Copyright Notice</a> <sup>[25]</sup>
Enterprise Architect End User Licensing Agreement	<a href="#">Enterprise Architect End User Licensing Agreement</a> <sup>[26]</sup>
Acknowledgement of Trademarks	<a href="#">Acknowledgement of Trademarks</a> <sup>[29]</sup>

Spark Systems would also like to gratefully [acknowledge contributions](#)<sup>[32]</sup> to the development of Enterprise Architect.

### 1.3.1 Copyright Notice

**Copyright © 1998 - 2014 Sparx Systems Pty. Ltd. All rights reserved.**

The software contains proprietary information of Sparx Systems Pty Ltd. It is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright law. Reverse engineering of the software is prohibited. Please read the [license agreement](#)<sup>[26]</sup> for full details.

Due to continued product development, this information can change without notice. The information and intellectual property contained herein is confidential between Sparx Systems and the client and remains the exclusive property of Sparx Systems. If you find any problems in the documentation, please report them to us in writing. Sparx Systems does not warrant that this document is error-free. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without the prior written permission of Sparx Systems. Licensed users are granted the right to print a single hardcopy of the user manual per licensed copy of the software, but may not sell, distribute or otherwise dispose of the hardcopy without written consent of Sparx Systems.

**Sparx Systems Pty. Ltd.**

99 Albert St,  
Creswick, Victoria 3363,  
AUSTRALIA

Phone: +61 (3) 5345 1140  
Fax: +61 (3) 5345 1104

Support Email: [support@sparxsystems.com](mailto:support@sparxsystems.com)  
Sales Email: [sales@sparxsystems.com](mailto:sales@sparxsystems.com)

Website: [www.sparxsystems.com](http://www.sparxsystems.com)

## Scintilla and SciTE

Copyright 1998-2003 by Neil Hodgson <[neilh@scintilla.org](mailto:neilh@scintilla.org)> All Rights Reserved.

Permission to use and distribute this (Scintilla) software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appears in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

### 1.3.2 End User License Agreement

**Enterprise Architect Modeling Tool Version 11.0**

**Desktop, Professional, Corporate, Business and Software Engineering, Systems Engineering and Ultimate Editions**

Copyright © 1998-2014 Sparx Systems Pty Ltd. All Rights Reserved.

**IMPORTANT - READ CAREFULLY:** This End User License Agreement ("EULA") is a legal agreement between YOU as Licensee and SPARX SYSTEMS ("SPARX") for the SOFTWARE PRODUCT identified above. By installing, copying, or otherwise using the SOFTWARE PRODUCT, YOU agree to be bound by the terms of this EULA. If YOU do not agree to the terms of this EULA, promptly return the unused SOFTWARE PRODUCT to the place of purchase for a full refund.

The copyright in the SOFTWARE PRODUCT and its documentation is owned by Sparx Systems Pty Ltd A.C.N 085 034 546. Subject to the terms of this EULA, YOU are granted a non-exclusive right for the duration of the EULA to use the SOFTWARE PRODUCT. YOU do not acquire ownership of copyright or other intellectual property rights in any part of the SOFTWARE PRODUCT by virtue of this EULA.

Your use of this software indicates your acceptance of this EULA and warranty.

## DEFINITIONS

In this End User License Agreement, unless the contrary intention appears,

- "ACADEMIC EDITION" means an edition of the Software Product purchased for educational purposes at an academic discount price.
- "EULA" means this End User License Agreement
- "SPARX" means Sparx Systems Pty Ltd A.C.N 085 034 546
- "Licensee" means YOU, or the organization (if any) on whose behalf YOU are taking the EULA.
- "Registered Edition of Enterprise Architect" means the edition of the SOFTWARE PRODUCT which is available for purchase from the web site: <<http://www.sparxsystems.com/products/ea/purchase.html>>, following the thirty-day free evaluation period.

- "SOFTWARE PRODUCT" or "SOFTWARE" means Enterprise Architect, UML Case Tool, Desktop, Professional, Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions, which includes computer software and associated media and printed materials, and may include online or electronic documentation.
- "Support Services" means email-based support provided by SPARX, including advice on usage of Enterprise Architect, investigation of bugs, fixes, repairs of models, if and when appropriate, and general product support.
- "SPARX support engineers" means employees of SPARX who provide on-line support services.
- "Trial Edition of Enterprise Architect" means the edition of the SOFTWARE PRODUCT which is available free of charge for evaluation purposes for a period of thirty (30) days.
- "EA LITE" means the LITE edition of Enterprise Architect that is distributed free of charge as a read-only viewer of .EAP files.

## GRANT OF LICENSE

In accordance with the terms of this EULA, YOU are granted the following rights:

- To install and use one copy of the SOFTWARE PRODUCT, or in its place, any prior version for the same operating system, on a single computer. As the primary user of the computer on which the SOFTWARE PRODUCT is installed, YOU may make a second copy for your exclusive use on either a home or portable computer.
- To store or install a copy of the SOFTWARE PRODUCT on a storage device, such as a network server, used only to install or run the SOFTWARE PRODUCT over an internal network. If YOU want to increase the number of users entitled to concurrently access the SOFTWARE PRODUCT, YOU must notify SPARX and agree to pay an additional fee.
- To make copies of the SOFTWARE PRODUCT for backup and archival purposes only.

## EVALUATION LICENSE

The Trial Edition of Enterprise Architect is not free software. Subject to the terms of this agreement, YOU are hereby licensed to use this software for evaluation purposes without charge for a period of thirty (30) days.

Upon expiration of the thirty (30) days, the Software Product must be removed from the computer. Unregistered use of Enterprise Architect after the 30-day evaluation period is in violation of Australian, U.S. and international copyright laws.

SPARX may extend the evaluation period on request and at their discretion.

If YOU choose to use this software after the 30-day evaluation period, a license must be purchased (as described at <<http://www.sparxsystems.com/products/ea/purchase.html>>). Upon payment of the license fee, YOU will be sent details on where to download the Registered Edition of Enterprise Architect and will be provided with a suitable software 'key' by email.

## EA LITE

Subject to the terms of this Agreement, EA LITE may be installed on any machine indefinitely and free of charge. There are no fees or Sparx support services in relation to EA LITE.

## ADDITIONAL RIGHTS AND LIMITATIONS

YOU hereby undertake not to sell, rent, lease, translate, adapt, vary, modify, decompile, disassemble, reverse engineer, create derivative works of, modify, sub-license, loan or distribute the SOFTWARE PRODUCT other than as expressly authorized by this EULA.

YOU further undertake not to reproduce or distribute license key-codes except under the express and written permission of SPARX.

If the Software Product purchased is an Academic Edition, YOU ACKNOWLEDGE THAT the license is limited to use in an educational context, either for self-education or use in a registered teaching institution. The Academic Edition may not be used to produce commercial software products or be used in a commercial environment, without the express written permission of SPARX.

## **ASSIGNMENT**

YOU may only assign all your rights and obligations under this EULA to another party if YOU supply to the transferee a copy of this EULA and all other documentation including proof of ownership. Your license is then terminated.

## **TERMINATION**

Without prejudice to any other rights, SPARX may terminate this EULA if YOU fail to comply with the terms and conditions. Upon termination YOU or YOUR representative shall destroy all copies of the SOFTWARE PRODUCT and all of its component parts or otherwise return or dispose of such material in the manner directed by SPARX.

## **WARRANTIES AND LIABILITY**

### **WARRANTIES**

SPARX warrants that the SOFTWARE PRODUCT will perform substantially in accordance with the accompanying written materials for a period of ninety (90) days from the date of receipt, and

Any Support Services provided by SPARX shall be substantially as described in applicable written materials provided to YOU by SPARX, and SPARX support engineers will make commercially reasonable efforts to solve any problems associated with the SOFTWARE PRODUCT.

### **EXCLUSIONS**

To the maximum extent permitted by law, SPARX excludes, for itself and for any supplier of software incorporated in the SOFTWARE PRODUCT, all liability for all claims, expenses, losses, damages and costs made against or incurred or suffered by YOU directly or indirectly (including without limitation lost costs, profits and data) arising out of:

- YOUR use or misuse of the SOFTWARE PRODUCT;
- YOUR inability to use or obtain access to the SOFTWARE PRODUCT;
- Negligence of SPARX or its employees, contractors or agents, or of any supplier of software incorporated in the SOFTWARE PRODUCT, in connection with the performance of SPARX's obligations under this EULA; or
- Termination of this EULA by either party for any reason.

### **LIMITATION**

The SOFTWARE PRODUCT and any documentation are provided "AS IS" and all warranties, whether express, implied, statutory or otherwise, relating in any way to the subject matter of this EULA or to this EULA generally, including without limitation, warranties as to: quality; fitness; merchantability; correctness; accuracy; reliability; correspondence with any description or sample, meeting your or any other requirements; uninterrupted use; compliance with any relevant legislation; and being error or virus free are excluded. Where any legislation implies in this EULA any term, and that legislation avoids or prohibits provisions in a contract excluding or modifying such a term, such term shall be deemed to be included in

this EULA. However, the liability of SPARX for any breach of such term shall, if permitted by legislation, be limited, at SPARX's option to any one or more of the following upon return of the SOFTWARE PRODUCT and a copy of the receipt:

If the breach relates to the SOFTWARE PRODUCT:

- The replacement of the SOFTWARE PRODUCT, or the supply of an equivalent SOFTWARE PRODUCT;
- The repair of such SOFTWARE PRODUCT, or the payment of the cost of replacing the SOFTWARE PRODUCT, or of acquiring an equivalent SOFTWARE PRODUCT; or
- The payment of the cost of having the SOFTWARE PRODUCT repaired.

If the breach relates to services in relation to the SOFTWARE PRODUCT:

- The supplying of the services again; or
- The payment of the cost of having the services supplied again.

## TRADEMARKS

All names of products and companies used in this EULA, the SOFTWARE PRODUCT, or the enclosed documentation may be trademarks of their corresponding owners. Their use in this EULA is intended to be in compliance with the respective guidelines and licenses. Windows, Windows XP, Windows 2003 Server, Windows 2008 Server, Windows Vista, Windows 7 and Windows 8 are trademarks of Microsoft.

## GOVERNING LAW

This agreement shall be construed in accordance with the laws of the Commonwealth of AUSTRALIA, in the state of Victoria.

### 1.3.3 Trademarks

Sparx Systems acknowledge the following trademarks and registered trademarks, which are used throughout the Enterprise Architect documentation.

#### Acknowledgement

Company	Trademarks	Registered Trademarks
<b>Microsoft Corporation</b>	ActiveX™ Microsoft Office™ Microsoft Word™ VB.NET™	Access® Active Directory® Excel® IntelliSense® JScript® Microsoft® MS Access® MS Access 2007® SQL Server® SQL Server Express®

Company	Trademarks	Registered Trademarks
		Visio® Visual Basic® Visual SourceSafe® Visual Studio® Win32® Windows® (Windows XP, Windows 7, Windows 8) Windows Server® (Windows 2003 Server, Windows 2008 Server) Windows Vista®
<b>The Object Management Group (OMG)</b>	OMG™ Object Management Group™ The CORBA logo ORB™ Object Request Broker™ The CORBA Academy design OMG Interface Definition Language™ IDL™ CORBAservices™ CORBAfacilities™ CORBAmed™ CORBAnet™ Unified Modeling Language™ MOF™ CWM™ BPMN™ SoaML™ Model Driven Architecture™ MDA™ OMG Model Driven Architecture™ OMG MDA™	The OMG Object Management Group logo The Information Brokerage® CORBA® CORBA Academy® IIOP® XML® UML® The UML Cube logo
<b>The Open Group</b>		Archimate®
<b>Open Geospatial Consortium</b>	GML™	

Company	Trademarks	Registered Trademarks
<b>Methodologies Corporation</b>	SOMF™	
<b>Embarcadero Technologies Inc</b>	InterBase™	Delphi®
<b>Oracle Corporation</b>	MySQL™	Oracle® Java®
<b>Python Software Foundation</b>	Python™	
<b>Esri</b>	ArcGIS™ ArcCatalog™	
<b>Linux Mark Institute</b>	Linux™	
<b>Innodb Oy</b>	InnoDB™	
<b>Firebird Foundation Incorporated</b>		FireBird®
<b>Ingres Corporation</b>	Ingres™	
<b>IBM</b>	DB2™ Informix™	Rational® Rational® Rhapsody
<b>Sybase Inc</b>	Sybase Central™	Adaptive Server® Adaptive Server® Anywhere Adaptive Server® Everywhere SQL Anywhere® Sybase®
<b>Progress Software</b>		OpenEdge®

Company	Trademarks	Registered Trademarks
<b>Corporation</b>		Progress OpenEdge®

### 1.3.4 Acknowledgements

Some parts of this application include code originally written by various authors and modified for use in Enterprise Architect.

#### Acknowledgements

Name	Detail	Link
<b>Marquet Mike</b>	Print listview contents	<a href="mailto:mike.marquet@altavista.net">mike.marquet@altavista.net</a>
<b>Davide Pizzolato</b>	CXImage Library © 7-Aug-2001	<a href="mailto:ing.davide.pizzolato@libero.it">ing.davide.pizzolato@libero.it</a>
<b>Neil Hodgson</b>	Scintilla editor © 1998-2003	<a href="mailto:neilh@scintilla.org">neilh@scintilla.org</a>
<b>Others</b>	Many thanks to all those who have made suggestions, reported bugs, offered feedback and helped with the beta-testing of Enterprise Architect  Your help has been invaluable	



## 1.4 Order Enterprise Architect

Enterprise Architect is designed, built and published by Sparx Systems, and available through the Sparx Systems website

- Free, 30-day trial version (although you can extend the evaluation period)
- Registered purchase version that provides the facilities of the edition you require, as unlocked by the license keys that you purchase from Sparx Systems
- The trial version of Enterprise Architect is identical to the registered version with the exception that all diagrams are output to files with an embedded watermark. The trial software stops working after the trial period has elapsed.
- On purchase of a suitable license or licenses, the registered version is made available for download.
- The latest information on pricing and purchasing is available on the Purchase and Pricing page of the Sparx Systems website, and from Sparx Sales.

### Purchase Options

- On-line using a secure credit-card transaction
- Fax
- Check or equivalent
- Bank transfer

### Learn more

- [The Trial Version](#)<sup>[18]</sup>
- [Editions Available](#)<sup>[20]</sup>
- [sales@sparxsystems.com](mailto:sales@sparxsystems.com)
- [Pricing and Purchase Options](#)

## 1.5 Installation

Enterprise Architect is distributed as a single executable setup file (.exe). It is a standard Windows installer based file that will check and update previous installations of Enterprise Architect.

- After you install Enterprise Architect, you can immediately begin to create projects as .EAP files.
- The latest evaluation and registered versions of Enterprise Architect are always available from the Sparx Systems website.
- The registered version is available through the registered user area of the web site, which requires a username and password to access.
- These are provided upon purchase of a license.

### Installation

Topic	Detail	See also
<b>System Requirements</b>	The system requirements for installing Enterprise Architect are defined on the Enterprise Architect System Requirements page of the Sparx Systems website.	<a href="#">Enterprise Architect System Requirements</a>
<b>Windows Vista / Windows 7</b>	<p>Under Windows Vista and Windows 7 (with User Account Control turned on) an application starts with only Standard permissions, regardless of what level of authority the current user has.</p> <p>As a result, an installer run normally with an Admin account under Vista only has Standard privileges and either is not able to write to certain critical areas of the registry/file system, or redirects the write requests to a per-user virtualized registry/file system.</p> <p>Sparx Systems recommend that if you are installing on Windows Vista or Windows 7, always run the Enterprise Architect installer with Administrator privileges (right-click on the downloaded installer icon and select the <b>Run as administrator</b> menu option).</p>	
<b>Install Enterprise Architect</b>	<p>Run the Enterprise Architect setup program. Generally you can accept all the default options without change.</p> <p>To place Enterprise Architect in a directory other than the default, enter the name of the destination when prompted.</p> <p>You might be prompted to restart your computer when the installation completes; although this is not always necessary (if you already have the components Enterprise Architect requires installed on your computer), you should restart just to be certain.</p>	
<b>Installation on Linux and MacOS</b>	<p>If you intend to run Enterprise Architect on Linux, refer to the page <i>Installing Enterprise Architect inside Wine</i>, on the Sparx Systems website.</p> <p>To support the Scripting facility in the Corporate and extended</p>	<a href="#">Installation inside Wine</a>

Topic	Detail	See also
	editions of Enterprise Architect, you must also install Internet Explorer 6.0 or later revisions.	
<b>Using a Third Party DBMS as Model Repository</b>	<p>If you plan to use SQL Server, MySQL, PostgreSQL, Access 2007, Sybase Adaptive Server Anywhere or Oracle 9i, 10g or 11g as a model repository, then you must use the Corporate, Business and Software Engineering, Systems Engineering or Ultimate editions of Enterprise Architect.</p> <p>You also require additional files and supplementary installation processes; please note that installation and maintenance of these database management systems is not covered under the support agreement.</p> <p>Users planning to use SQL Server, MySQL, PostgreSQL, Sybase Adaptive Server Anywhere, Access 2007 or Oracle 9i, 10g or 11g as their model repository can access scripts that create the required data structures for the choice of DBMS, from the Sparx Systems website; to access these scripts, please follow the links at right.</p>	<p><a href="#">Corporate Edition Resources Page</a></p> <p><a href="#">Trial Corporate Edition Resources Page</a></p>

## 1.6 Register a Full License

The **Trial** version of Enterprise Architect available for download is an evaluation version only. For the **full** version you must first purchase one or more *licenses*.



The license key supplied determines which edition (Enterprise Architect Desktop, Professional, Corporate, Business and Software Engineering, System Engineering or Ultimate) is activated after installation.

The Corporate, Business and Software Engineering, System Engineering and Ultimate editions can be activated by either a private license key or a shared (floating) license key. Shared license keys, for a user population that includes temporary users, are checked out of and in to a Sparx Systems key store.

### Use to

- Obtain and activate the registered version of Enterprise Architect



### How to

Step	Action	See also
1	<p>Purchase one or more licenses.</p> <p>Once you have paid for a licensed version of Enterprise Architect, you receive (via email or other suitable means):</p> <ul style="list-style-type: none"> <li>• a license key or keys</li> <li>• a web site address from which to download the full version</li> <li>• if you have purchased a Sparx Systems key store, a web site address from which to download the key store</li> </ul>	<a href="#">Editions Available</a> 
2	Save the license key(s) and download the latest full install package from the address supplied.	
3	Run the setup program to install the full version.	
4	<p>Open Enterprise Architect from the Start Menu or desktop icon.</p> <p>The License Management dialog displays.</p>	
5	<p>Click on the <b>Add Key</b> button.</p> <p>The Add Registration Key dialog displays; register your key.</p>	<a href="#">Add License Key</a> 
6	<p>Click on the <b>OK</b> button.</p> <p>The full version is now activated on your PC, and Enterprise Architect displays the message:</p> <p><i>Registration succeeded! Thank you for purchasing Enterprise Architect &lt;type&gt; Edition</i></p>	

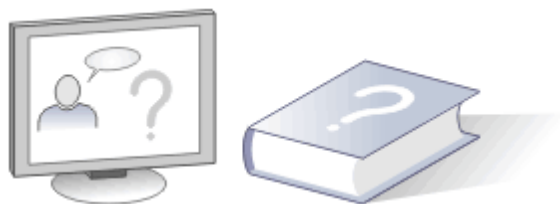
**Notes**

- Private license keys and shared license keys have different formats, so you cannot use one in place of the other

**Learn more**

- [Upgrade an Existing License](#) 
- [Register Add-In](#) 

## 1.7 Help and Support



Enterprise Architect has three main help and information systems to assist you in using the product:

- Learning Center
- Help
- The Sparx Systems website


In addition we recommend that you fully explore the sample project supplied with the Enterprise Architect installer. It assists you in learning to use Enterprise Architect and offers tips on getting the most out of the product features. Click on the **EAExample** option on the Start Page.

### Getting Help and Support

Topic	Detail	See also
<b>Learning Center</b>	<p>The Enterprise Architect Learning Center provides guidance and demonstrations to help you understand any area of Enterprise Architect that you are interested in.</p> <p>The Learning Center automatically displays on the right of the screen when you first open Enterprise Architect, showing the <i>Introduction To The Learning Center</i> topic. You can select other topics or task areas by clicking on the appropriate part of the topic path ('breadcrumb trail') at the top of the window, and selecting from the drop-down list.</p>	<a href="#">Learning Center</a> [74]
<b>Enterprise Architect Help</b>	<p>Enterprise Architect Help provides comprehensive documentation of Enterprise Architect and covers every aspect and facility of the product. To access Help within Enterprise Architect:</p> <ul style="list-style-type: none"> <li>• Click on the Help icon ( ? ) in the various toolbars</li> <li>• Select the <b>Help   Help Contents</b> menu option</li> <li>• Click on the <b>Help</b> button on a dialog (for Help specific to that dialog)</li> </ul> <p>Enterprise Architect Help is extensive; if you cannot quickly locate the topic you require in the online contents list, you can use one of two search facilities:</p> <ul style="list-style-type: none"> <li>• Click on the Index tab, type in a keyword or key phrase appropriate to the subject you require help for, and press ( <b>Enter</b> ); double-click on the appropriate index item</li> <li>• Click on the Search tab, type in a word or phrase to search for, and click on the <b>List Topics</b> button; double-click on the required topic.</li> </ul>	<a href="#">Available Help</a> <a href="#">File Formats</a> [39]

Topic	Detail	See also
	The Enterprise Architect Help is also available separately from the product, in different formats.	
<b>Sparx Systems Web Site</b>	<p>The Sparx Systems web site is also extensive, and provides information and announcements concerning the company and its full range of products, as well as tutorials, white papers, templates and solutions.</p> <p>It also provides a user forum, community site (see below) and support network; Sparx Systems are highly responsive to user feedback and requirements, and the web site enables rapid communication concerning problems, solutions and enhancements.</p> <p>You can access the web page and user forum within Enterprise Architect from the <b>View   Web Browser</b> menu option.</p>	<a href="#">Sparx Systems Web Site</a> <a href="#">Enterprise Architect User Forum</a>
<b>Community Site</b>	<p>The Sparx Systems website also hosts the Sparx Systems Enterprise Architect Community Site. This is a central location for the Enterprise Architect community to publish resources and share experiences.</p> <p>From this site you can download the latest news, tutorials, resources, best practices, tips, techniques and user-generated content for Enterprise Architect. You can also, as a registered author, contribute content and share your expertise with the wider community.</p>	<a href="#">Sparx Systems Enterprise Architect Community Site</a>

#### Learn more

- [Support](#)  (for contact details)

### 1.7.1 Available Helpfile Formats

You can access the latest Enterprise Architect help files from the following locations

Format	Link
.CHM	<a href="http://www.sparxsystems.com/bin/EA.chm">www.sparxsystems.com/bin/EA.chm</a>
.CHM inside a .ZIP file:	<a href="http://www.sparxsystems.com/bin/EAHelp.zip">www.sparxsystems.com/bin/EAHelp.zip</a>
.PDF	<a href="http://www.sparxsystems.com/bin/EAGuide.pdf">www.sparxsystems.com/bin/EAGuide.pdf</a>
.HTML	<a href="http://www.sparxsystems.com/EAGuide/index.html">www.sparxsystems.com/EAGuide/index.html</a>

**Version and release date information for the help files can be found at**

Topic	Link
Help file	<a href="http://www.sparxsystems.com/ea_downloads.htm#Helpfiles">www.sparxsystems.com/ea_downloads.htm#Helpfiles</a>
Help file(registered users).	<a href="http://www.sparxsystems.com/registered/reg_ea_down.htm#Helpfiles">www.sparxsystems.com/registered/reg_ea_down.htm#Helpfiles</a>

### **1.7.2 Support**

Technical support for Enterprise Architect is available to registered users. Responses to support queries are sent by email.

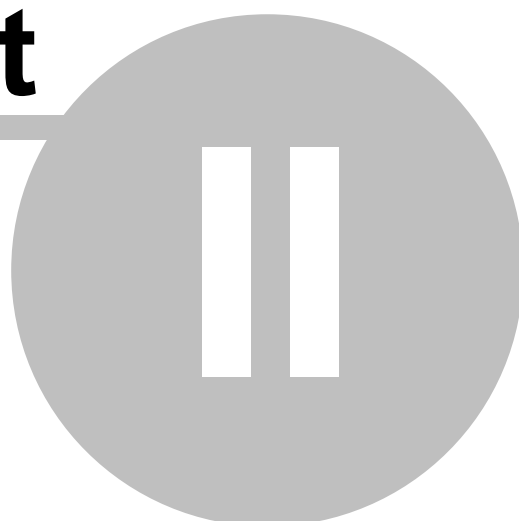
Sparx Systems endeavors to provide a rapid response to all product-related questions or concerns.

- Registered users can lodge a support request, by visiting: [http://www.sparxsystems.com/registered/reg\\_support.html](http://www.sparxsystems.com/registered/reg_support.html)
- Trial users can contact Sparx Systems with questions regarding their evaluation at: [support@sparxsystems.com](mailto:support@sparxsystems.com)
- An online user forum is also available for your questions and perusal, at <http://www.sparxsystems.com/cgi-bin/yabb/YaBB.cgi>

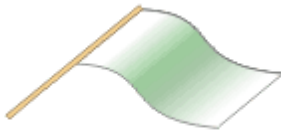


# Part

---



## 2 Getting Started



This guide helps you to understand the options available when you start Enterprise Architect, and to quickly gain an understanding of how to use these options to create models.

### Getting Started

Learning Activities	Link
After starting Enterprise Architect, use the Quick Start tutorial to immediately create a project	<a href="#">A Quickstart Tutorial</a> <sup>[47]</sup>
Read about work areas applicable to certain Project Roles	<a href="#">Project Roles</a> <sup>[183]</sup>
Review the Enterprise Architect User Interface, or workspace	<a href="#">Enterprise Architect User Interface</a> <sup>[67]</sup>

At various points throughout the *Enterprise Architect User Guide*, there are further Quick Start topics and sections to help you use the system immediately to experiment with a feature of the system. Use the Help **Index** tab and search for *Quick Start* to locate these topics.

### Learn more

- [Common Tasks](#) <sup>[45]</sup>
- [Basics](#) <sup>[43]</sup>

## 2.1 Basics

When you install Enterprise Architect on your system, two items are created:

- An Enterprise Architect icon on your Windows desktop
- A new program folder called *Enterprise Architect<version number>* in your Windows **Start > All Programs** menu; this folder contains the execution option **Enterprise Architect**

You can **start** Enterprise Architect by **clicking on** either of these objects. After a short pause, the **Start Page** displays, from which you can:

- Open a project file (.EAP file or .FEAP file)
- Create a new project (.EAP file or .FEAP file)
- Connect to a DBMS repository (Corporate and extended editions)
- Connect to a project via the Cloud
- Open the Learning Center, to read quick introductions to performing a range of basic tasks

### Notes

- If you changed the default system folder name during installation, the *Enterprise Architect<version number>* folder in the **All Programs** menu will have the name you provided
- By default, when you install Enterprise Architect, an empty 'starter' project called **EABase.EAP** is installed, as well as an example project named **EAExample.EAP**; we recommend that new users select the EAExample file and explore it in some detail while you become familiar with UML and software engineering using Enterprise Architect
- Enterprise Architect .EAP files default to use **JET 3.5** as the database engine, which does **not** support **unicode character sets**

If you want to use unicode character sets (for example, to provide user interface texts in languages other than your Windows-defined native language), you must either upsize to a **DBMS repository** or set **JET 4.0** as the database engine; resetting the database engine ensures compatibility with .EAP files that support unicode character sets and that are in turn compatible with versions of MS Access later than Access 97

If your .EAP project is not in a Jet 4.0 database, you should also download a copy of the **Jet 4.0 EABase** model from the Sparx Systems website, and do an **EAP to EAP transfer** of your model into the Jet 4.0 file

### Learn more

- [Installation](#)<sup>[34]</sup>
- [Start Page](#)<sup>[68]</sup>
- [A Quick Start Tutorial](#)<sup>[47]</sup>, to begin a guided exploration of Enterprise Architect immediately
- [Common Tasks](#)<sup>[45]</sup>, to get an overview of the kind of work you might perform with Enterprise Architect
- [Open a project file](#)<sup>[202]</sup>
- [Create a new project](#)<sup>[210]</sup>
- [Server Based Repositories](#)<sup>[214]</sup>
- [Connecting to Projects Via the Cloud](#)<sup>[262]</sup>
- [Learning Center](#)<sup>[74]</sup>

- [General Options](#),<sup>[605]</sup> to set JET 4.0 as database engine
- [The Jet 4.0 EABase Model](#) (Online Resource)
- [Perform a Data Transfer](#)<sup>[504]</sup>

## 2.2 Common Tasks

Enterprise Architect is a powerful modeling tool that can be used by project managers, business analysts, developers and testers alike to build business, real-time and software systems. Across this work there are several key areas in which the system can play a vital role.

### Common tasks

Tasks	Detail
<b>Manage Projects Effectively</b>	<p>Enterprise Architect provides a wide range of tools for planning, executing and successfully completing your projects. You can:</p> <ul style="list-style-type: none"> <li>• Create a shared vision of your project, improve communication and help team members agree upon the design of a proposed system</li> <li>• Evaluate project risk factors, report any changes or defects and develop strategies to tackle potential problems</li> <li>• Estimate the amount of time and effort required to complete a project and maximize the efficiency of use of staff</li> <li>• Use Auditing, Project Baselines and User Access Controls to better manage change in a collaborative environment</li> </ul>
<b>Analyze Strategic Business Needs</b>	<p>A Business Analyst is responsible for gathering requirements, modeling potential solutions and evaluating the business needs of an organization. You can:</p> <ul style="list-style-type: none"> <li>• Build a coherent picture of how a business operates through requirements gathering, Use Case analysis and modeling business rules</li> <li>• Help achieve business process improvement and implement change using tools such as Strategic Models, Use Case Scenarios, Business Rule Models, Flow Charts, Auditing and the Team Review window</li> </ul>
<b>Build and Develop Systems</b>	<p>The developer is responsible for implementing a solution using a range of tools. You can:</p> <ul style="list-style-type: none"> <li>• Bring the power and benefits of Enterprise Architect into your favorite Integrated Development Environment (IDE)</li> <li>• Use Enterprise Architect's in-built support for writing code, including syntax highlighting, line numbering, auto completion, bookmarking and automatic indentation</li> <li>• Create breakpoints to step through code execution, examine variables and view error messages via the Debug window</li> <li>• Apply Visual Execution Analysis to identify costly function calls, explain system behavior and establish the sequence of events that occur immediately prior to system failure</li> <li>• Use Enterprise Architect's support for round trip engineering and synchronizing source code with a corresponding model</li> <li>• In addition to generating code in many popular programming languages, create powerful database solutions that automatically generate DDL scripts for more than 10 different Database Management Systems</li> </ul>

Tasks	Detail
<b>Test and Investigate</b>	<p>Quality Assurance teams test software to identify defects, verify that it satisfies all requirements and ensure that it produces expected results. You can:</p> <ul style="list-style-type: none"><li>• In addition to applying the integrated JUnit and NUnit testing capabilities, create and manage test scripts for model elements, covering Unit, Integration, Scenario, System and Acceptance tests</li><li>• Visualize the execution of code to better understand how applications work and the sequence of events that leads to program failure</li></ul>

## 2.3 A Quick Start Tutorial



### Welcome to Enterprise Architect!

This quick-start tutorial helps you get up to speed with Enterprise Architect.

As you read through this tutorial, it is recommended that you have Enterprise Architect open so that you can try out the tasks described. By the end you should be able to begin modeling your own business/software projects with Enterprise Architect.

### Basic Tasks

Task	Link
Creating a new project	<a href="#">Creating a New Project</a> <sup>[47]</sup>
Adding Views to your model	<a href="#">Views</a> <sup>[49]</sup>
Adding Packages to your model	<a href="#">Packages</a> <sup>[50]</sup>
Adding diagrams	<a href="#">Adding Diagrams</a> <sup>[50]</sup>
Adding elements to your model Packages	<a href="#">Elements</a> <sup>[52]</sup>
Creating links between model elements	<a href="#">Links</a> <sup>[55]</sup>
Refining the model	<a href="#">Refining the Model</a> <sup>[57]</sup>

Throughout the descriptions there are hyperlinks to more detailed information on a range of topics.

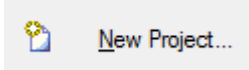
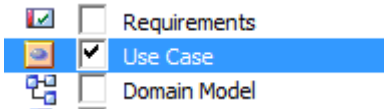
### 2.3.1 Create a Project

A project is a single file or repository-based store for one or more Models.

- The first step in getting started with Enterprise Architect is to either open an existing project, or create a new one
- In this example we create a new file-based project and add some template-based models to kick-start our modeling

- Once your new project has been created it will have a simple Use Case View added for you to explore and tailor to your requirements
- You can re-open your project at any time by double-clicking on it in a file browser
- It should also appear in your **Recent Projects** list on the Start page

### Create a New Project


Step	Action	Result
1	Start Enterprise Architect.	The Open Project dialog displays.  (If the dialog does not appear, press <b>Ctrl+O</b> to invoke it.)
2	Click the <b>New Project</b> button to select a suitable location and name for your new project.  	The standard Windows file browser dialog displays.  File-based Enterprise Architect projects are named with a .EAP extension.  You can also create projects on Firebird by clicking on the drop-down arrow in the <b>Save as type</b> field and selecting the .FEAP file extension.
3	Locate a suitable folder for your project and enter a distinctive name.  Click on the <b>Save</b> button to continue.	Enterprise Architect creates a new project file and places it in the specified location.  The project is then opened automatically and the New Model Wizard displays.
4	In the left hand column ensure that <b>Basic UML 2 Technology</b> is selected - and in the right hand side tick the <b>Use Case View</b> check box.    Click on the <b>OK</b> button.	The Model Wizard automatically creates a new Use Case model for you, with an initial diagram, some notes and default elements to help get you started.  At this stage all your changes have been saved to file and do not require further action.



### 2.3.2 Add a View to your Model

A View is a top level Package within a model. It can be categorized by purpose using different icon types, such as Use Case View, Component View or Deployment View. Views are used to contain Packages, diagrams and elements - the building blocks of your model.

#### Add a View

Step	Action	See also
1	<p>Select a model root node in the Project Browser, then either:</p> <ul style="list-style-type: none"> <li>Click the <b>New Package</b> button () on the Project Browser toolbar</li> <li>Select <b>Project   New Package</b></li> <li>Right-click the model root node and, from the context menu, select <b>New View</b>, or</li> <li>Press <b>Ctrl+W</b></li> </ul> <p>The Create New View dialog displays.</p>	<a href="#">Views</a> <sup>[769]</sup>
2	Enter an appropriate name for the new View.	
3	Click the appropriate radio button to select an icon for the View.	
4	<p>Click on the <b>OK</b> button.</p> <p>The Create New View dialog closes.</p> <p>The new View is created as a child of the selected model root node.</p>	

#### Notes

- There are six types of View, which represent conventional ways of categorizing the purpose of a Model View; these Views represent different structural or behavioral aspects of the same model

#### Learn more

- [Add a Package to your model](#) <sup>[50]</sup>
- [Create a new diagram](#) <sup>[50]</sup>
- [Add an element](#) <sup>[52]</sup>
- [Read more about views](#) <sup>[769]</sup>

### 2.3.3 Add New Packages

A **Package** is a container of model elements, which is represented in the Project Browser as a 'folder' icon.

- A Package holds the model building blocks, such as diagrams, elements and other Packages
- To begin developing your model, you create an initial Package to hold your first diagrams and model structures

**Access** [Project Browser Package context menu](#) | **Add a Package ( Ctrl+W )**

#### Add a Package to your model

Step	Action	See also
1	Select a Package or View in the Project Browser, and select the <b>Add a Package</b> option.  The New Model Package dialog displays.	
2	In the <b>Name</b> field, type a suitable name for the Package.	
3	Optionally, select the <b>Automatically add new diagram</b> checkbox, to automatically create a default diagram within your new Package, to speed your development.  If you select this checkbox, you can also select the <b>Open new diagram</b> checkbox to immediately open the new diagram when it is created.	
4	Click on the <b>OK</b> button.  The new Package is inserted into the tree at the current location and, if you selected to create a new diagram, the New Diagram dialog displays.  In the <b>Name</b> field, type an appropriate name for the diagram, and then select the diagram category and diagram type. Click on the <b>OK</b> button.  The diagram displays in the Diagram View, for you to work on.	<a href="#">New Diagrams</a> [822]

#### Learn more





- [Packages](#) [772]
- [Add a Package](#) [772]

### 2.3.4 Create a New Diagram

A diagram is a visual representation of

- The elements of your model
- Their attributes and characteristics
- How they are connected and/or interact with each other
- Different diagram types show different aspects of the model and the relationships between elements.

**Add new diagrams to your model**

Step	Action	See also
1	<p>Select a Package or View in the Project Browser, then either:</p> <ul style="list-style-type: none"> <li>Click on the <b>New Diagram</b> button  on the Project Browser toolbar</li> <li>Press <b>Ctrl+Insert</b></li> <li>Select <b>Project   New Diagram</b>, or</li> <li>Right-click the Package or View and, from the context menu, select <b>Add Diagram</b></li> </ul> <p>The New Diagram dialog displays.</p>	<a href="#">Add Elements</a>  <sup>52</sup> <a href="#">Move Elements</a>  <sup>57</sup> <a href="#">Learn more about UML diagrams</a>  <sup>118</sup>
2	<p>Enterprise Architect provides a default diagram name, that is the same name as the parent Package.</p> <p>Update the diagram name as required.</p>	
3	<p>Select a category of diagrams in the left hand pane.</p> <p>The list of available diagram types in the right hand pane is updated according to your selection.</p>	
4	Select the type of diagram to add from the right hand pane.	
5	<p>Click on the <b>OK</b> button.</p> <p>The New Diagram dialog closes.</p> <p>A new diagram is created as a child of the currently selected Package.</p> <p>The new diagram is opened in the Diagram View, ready for editing.</p>	


**Notes**

- When you create a Package, if you leave the **Automatically add new Diagram** option selected, the New Diagram dialog displays automatically

**2.3.5 Save Changes**

Throughout much of your work in Enterprise Architect, any changes you make are automatically saved when you close the *dialog* (data entry window) on which you made the changes. In some cases the dialog contains a **Save** or **Apply** button, which enables you to save your changes and then keep working on the dialog.

**If there is no specific dialog, such as when you create a diagram, you can save your work by**

- Clicking on the **Save** icon in the Diagram toolbar ()
- Pressing the **Ctrl+S** keyboard keys, or
- Selecting the **Diagram | Save** menu option

Often, Enterprise Architect does not let you close a screen without confirming that you want to save or discard your changes. You can also save your diagram changes automatically, by selecting the **Auto Save Changes** checkbox on the Diagram Behavior page of the Options dialog.

#### Learn more

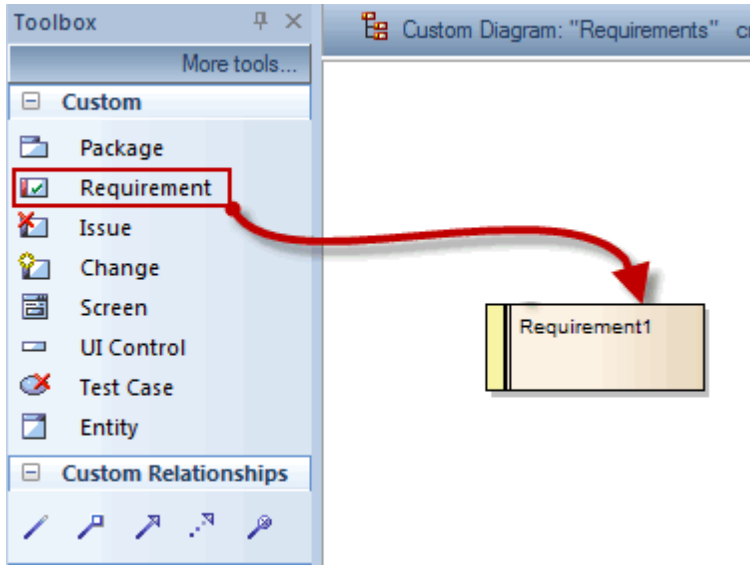
- So far you have been using the Project Browser and Diagram View to develop your project; at this point you should find out a bit more about the other facilities of the Enterprise Architect [User Interface](#)<sup>[67]</sup> or Application Workspace
- When you have finished exploring the User Interface topics, go to the [Summary of Typical Tasks](#)<sup>[183]</sup> to identify areas of Enterprise Architect that provide particular support for your job role
- Explore other options for managing the way your diagrams display and function, on the [Diagram Behavior](#)<sup>[625]</sup> page

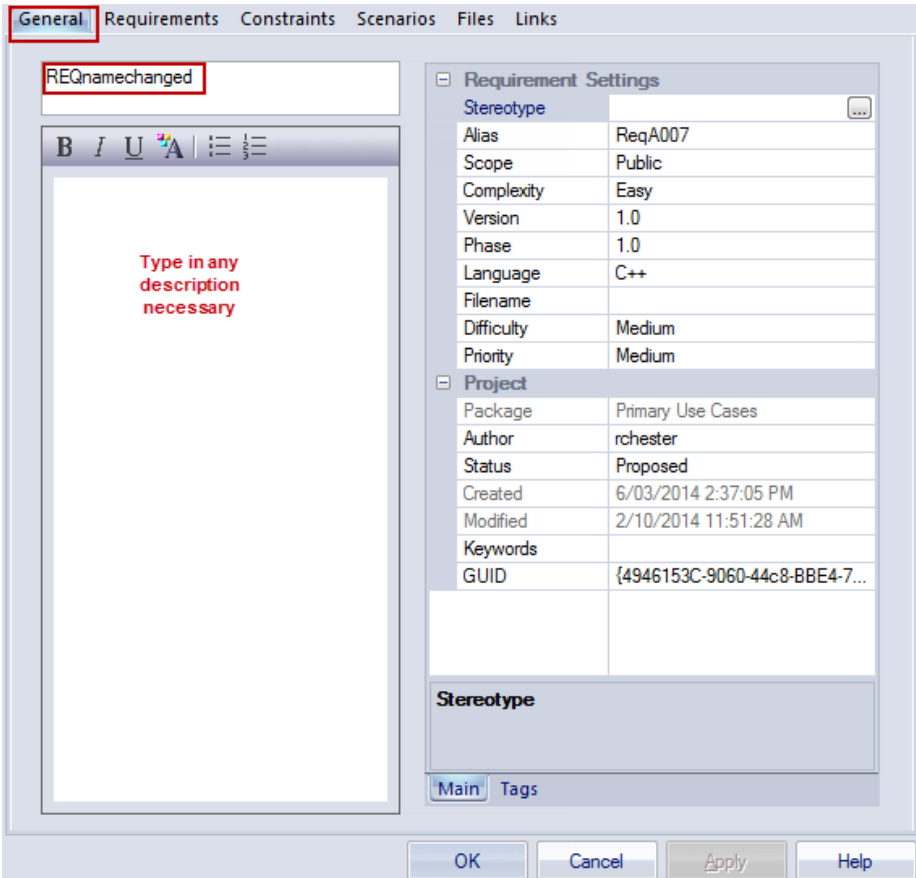
### 2.3.6 Create New Elements on a Diagram

Models are constructed from elements, each of which has its own meaning, rules and notation. Generally, when you create a new element in your model, you want to use that element on a diagram. The simplest way to do this is to create the element directly on the diagram. In the following procedure, we will use a Requirement element as the example.

#### Create a new element on a diagram in your model

Step	Action	See also
1	Display the Diagram Toolbox; either: <ul style="list-style-type: none"> <li>• Select <b>Diagram   Diagram Toolbox</b>, or</li> <li>• Press <b>Alt+5</b></li> </ul>	
2	Display the diagram on which the elements are to be created in the Diagram View. To open the diagram, double-click on the diagram name in the Project Browser.  The diagram opens and the Toolbox is updated to display the categories of elements and relationships that are applicable to that type of diagram. (In this example, the <b>Custom</b> toolbox.)	<a href="#">Diagram View</a> <sup>[784]</sup>
3	Click on the appropriate icon in the Toolbox to select the type of element to create (in this example, the <b>Requirement</b> element).  The element type is highlighted in the Toolbox.	
4	Click on the diagram at the location where you want to place the element.	

Step	Action	See also
	 <p>The new element is created as a child of the Package that contains the diagram, and is placed onto the diagram at the cursor position.</p> <p>The element Properties dialog displays.</p>	
5	<p>Use the Properties dialog to define the element properties that you want.</p>	<p><a href="#">Properties Dialog</a><sup>[956]</sup></p>

Step	Action	See also
		
6	<p>Click on the <b>OK</b> button.</p> <p>The Properties dialog closes.</p>	

### Notes

- You can also drag or paste existing elements onto a diagram from the Project Browser
- If you are creating several elements of one type, after creating the first just press **Shift+F3** or **Ctrl +Click** to create the next element of that type

### Learn more

- [Paste From the Project Browser](#)<sup>[833]</sup>
- [Add connectors](#)<sup>[55]</sup>
- [The Quick Linker](#)<sup>[896]</sup>
- [Move elements](#)<sup>[57]</sup>
- [UML elements](#)<sup>[1265]</sup>

### 2.3.7 Add Connectors

Connectors define specific relationships between specific elements, so you usually create them directly on the diagram by dragging the required relationship type from the Diagram Toolbox. As for elements, the Toolbox automatically presents the connector or relationship types appropriate to the type of diagram.

#### Define a relationship between two model elements on a diagram

Step	Action	See also
1	<p>Ensure that the Diagram Toolbox is visible.</p> <p>To display the Diagram Toolbox, either:</p> <ul style="list-style-type: none"> <li>• Select <b>Diagram   Diagram Toolbox</b> or</li> <li>• Press <b>Alt+5</b></li> </ul>	
2	<p>Ensure that the diagram containing the elements to be connected, is open in the Diagram View.</p> <p>To open the diagram, double-click on the diagram name in the Project Browser.</p> <p>The selected diagram opens and the Toolbox is updated to display the categories of elements and relationships that are applicable to that diagram type.</p>	<a href="#">Diagram View</a> <sup>[784]</sup> <a href="#">Creating Elements on Diagrams</a> <sup>[527]</sup>
3	<p>Click on the required connector in the Toolbox to select the type of connector to draw.</p> <p>The connector type is highlighted in the Toolbox.</p>	
4	<p>Click on the source element in the relationship, then drag across to the target element.</p> <p>The selected connector is drawn between the two elements.</p> <p>The connector Properties dialog displays.</p>	
5	<p>Use the Properties dialog to define the characteristics of the connector as required.</p>	<a href="#">Properties Dialog</a> <sup>[956]</sup>
6	<p>Click on the <b>OK</b> button.</p> <p>The Properties dialog closes.</p>	

#### Notes

- If you are creating several connectors of one type, after creating the first just click on the appropriate source element and press **F3** to create the next connector of that type
- As you drag a connector, you can press **Shift** to create a bend in the connector; if necessary, you can

put several bends in the connector line, pressing **Shift** every time you want to change direction

- To roll back the bends, keep holding the left mouse button down and press **Backspace** as many times as is necessary
- To find out more about the type of connector you have dragged on to a diagram, right-click on the connector and select the **UML Help** menu option, which displays a Help page on the connector type

#### Learn more

- [Quick Linker - Create Connectors](#) <sup>[898]</sup>
- [Create Connector in Project Browser](#) <sup>[1118]</sup>
- [UML Connections](#) <sup>[1389]</sup>

### 2.3.8 Modify Properties

When you create an element and connect it to another element, you usually have to define various characteristics of both the element and the connector to identify the purpose and function they represent. You do this using a Properties dialog.

When you create elements, Enterprise Architect automatically names and numbers them by type - for example, Class1, Class2 - so you should at least change the **Name** field to more easily identify each element. Enterprise Architect does not automatically name connectors, but for many connector types you should provide a name that describes the purpose of the connection.

#### Modify the characteristics of a model element or connector

Step	Action	See also
1	<p>Either:</p> <ul style="list-style-type: none"> <li>• Double-click on an element or connector in the diagram</li> <li>• Right-click an element in the Project Browser and, from the context menu, select <b>Properties</b>, or</li> <li>• Right-click a connector in a diagram and, from the context menu, select <b>Properties</b></li> </ul> <p>The appropriate Properties dialog displays.</p>	<a href="#">Element Properties Dialog</a> <sup>[956]</sup> <a href="#">Connector Properties Dialog</a> <sup>[1126]</sup>
2	Modify the element or connector characteristics as required.	<a href="#">Element Properties Dialog</a> <sup>[956]</sup> <a href="#">Connector Properties Dialog</a> <sup>[1126]</sup>
3	<p>Click on the <b>OK</b> button.</p> <p>The Properties dialog closes.</p> <p>The modifications are saved; the Project Browser and diagrams that contain the element are updated as necessary.</p>	



**Notes**

- Enterprise Architect is initially configured to display the Properties dialog automatically when you create an element or connector, but it is easy (and often convenient) to turn the dialog display off
- If the default display has been turned off, you can display the Properties dialog by either double-clicking on the element or connector in the diagram or by right-clicking on it in the Project Browser and selecting the **Properties** menu option

**2.3.9 Move Objects Around**

You have created a project containing Packages, diagrams and elements, and you have connected the elements.

How do you change where things are?

**Move diagrams, elements and connectors**

Task	Link
Change the Project Browser list order - re-arrange the order of items in the Project Browser	<a href="#">Move Elements Within a Package</a> <sup>[57]</sup>
Move elements between Packages - move diagrams, elements and child Packages from one parent Package to another	<a href="#">Move Objects Between Packages</a> <sup>[58]</sup>
Move elements on a diagram - re-arrange the position of diagram elements	<a href="#">Move Elements in a Diagram</a> <sup>[59]</sup>
Move elements between diagrams - cut an element from one diagram and paste it onto another	<a href="#">Move Elements Between Diagrams</a> <sup>[59]</sup>
Move connectors on a diagram - re-link a source element to a different target element	<a href="#">Move Connectors on a Diagram</a> <sup>[61]</sup>

**Learn more**

- [Delete Components](#) <sup>[64]</sup>
- [Arrange Connectors](#) <sup>[1112]</sup>
- [Order Package Contents](#) <sup>[672]</sup>



**2.3.9.1 Move Objects Within a Package**

In the Project Browser, the contents of a Package are listed in the order: diagrams, child Packages, elements.

- Elements are further arranged in order of type (but see *Notes*)
- Within a type, components are initially listed in alphabetical or numerical order

- You can change the order in which items are listed within their groups

#### Adjust the order in which items are listed in the Project Browser

Step	Action	See also
1	Click on an item in the Project Browser, then click on  or  in the toolbar at the top of the window.  The selected item is moved up or down in the Project Browser accordingly, but remains within its particular group of items.	
2	To revert to listing components in alphabetical order, in the Project Browser right-click on the Package and from the context menu select <b>Contents   Reset Sort Order</b> .	

#### Notes

- You can re-arrange elements in your preferred sequence **regardless** of type, by selecting the **Allow Free Sorting** checkbox on the General page of the Options dialog

#### Learn more

- [General Options](#) 

### 2.3.9.2 Move Objects Between Packages

When creating the various diagrams, elements and Packages that make up your model, at some point you might create one of these objects under the wrong parent Package. It is quite simple to relocate a model object from one Package to another, either to a higher level Package, a lower level Package or a different Package at the same level.

#### Move objects between Packages

Step	Action	See also
1	In the Project Browser, click and drag the model item to be relocated from its existing position, and drop it onto the new parent Package.  The dragged item and all of its child items are relocated to the target Package.	

#### Notes

- Moving elements in the Project Browser does not affect the use of elements in diagrams
- Moving an element or Package has no effect on any relationships that the element, Package, or elements within the Package have
- Moving a diagram generally does not affect the location of elements in Packages
- However, elements of certain types, such as Initial Node, Decision and Final Node, are used only

within one diagram and have no meaning outside that diagram; if you move a diagram containing these elements, they are moved to the new parent Package with the diagram

#### Learn more

- [Initial Node](#) <sup>[1313]</sup>
- [Decision](#) <sup>[1294]</sup>
- [Final Node](#) <sup>[1305]</sup>

### 2.3.9.3 Move Elements in a Diagram

To improve the understanding of a diagram or to simply improve the presentation, you might want to adjust the position of the elements on your diagram.

#### Adjust the position of the elements on your diagram

For coarse adjustments

Step	Action	See also
1	Click on a diagram element and drag it to its new position.	

For fine adjustments



Step	Action	See also
1	Click on a diagram element to select it.	
2	While holding down the <b>Shift</b> key, press the arrow keys to move the selected element one pixel at a time in the direction of the arrow ( <b>Shift+↑,↓,→,←</b> ).	

### 2.3.9.4 Move Elements Between Diagrams

If an element is present in one diagram but not in another, you can simply move or copy the element from one diagram into the other.



#### Move elements from one diagram to another

Step	Action	See also
1	In the Project Browser, double-click the name of the diagram that contains the element to be moved.  The diagram opens in the Diagram View.	

Step	Action	See also
2	Click on the element in the diagram to select it.	
3	<p>Either:</p> <ul style="list-style-type: none"> <li>Click on the <b>Cut</b> button () in the Default Tools toolbar, or</li> <li>Press <b>Ctrl+X</b></li> </ul> <p>The selected element is copied to the clipboard.</p> <p>However, the element remains visible on the current diagram until the clipboard content is pasted onto a different diagram.</p> <p>The model element itself is not affected in any way.</p>	
4	<p>In the Project Browser, double-click on the name of the diagram into which the element is to be moved.</p> <p>The diagram opens in the Diagram View.</p>	
5	<p>Either:</p> <ul style="list-style-type: none"> <li>Click on the <b>Paste</b> button () in the Default Tools toolbar, or</li> <li>Press <b>Ctrl+V</b> or <b>Shift+Insert</b></li> </ul> <p>The element is pasted from the clipboard onto the current diagram and removed from the original (source) diagram.</p> <p>Again, the model element itself is not affected in any way.</p>	

#### Copy elements from one diagram to another

Step	Action	See also
1	<p>In the Project Browser, double-click the name of the diagram that contains the element to be moved.</p> <p>The diagram opens in the Diagram View.</p>	
2	Click on the element in the diagram to select it.	

Step	Action	See also
3	<p>Either:</p> <ul style="list-style-type: none"> <li>Click on the <b>Copy</b> button (  ) in the Default Tools toolbar, or</li> <li>Press <b>Ctrl+C</b></li> </ul> <p>The selected element is copied to the clipboard.</p> <p>Neither the current diagram, nor the model element itself are affected in any way.</p>	
4	<p>In the Project Browser, double-click on the name of the diagram into which the element is to be copied.</p> <p>The diagram opens in the Diagram View.</p>	
5	<p>Either:</p> <ul style="list-style-type: none"> <li>Click on the <b>Paste</b> button (  ) in the Default Tools toolbar, or</li> <li>Press <b>Ctrl+V</b> or <b>Shift+Insert</b></li> </ul> <p>The element is pasted from the clipboard onto the current diagram.</p> <p>Again, neither the source diagram nor the model element itself are affected in any way.</p>	

### Notes


- By selecting more than one element at a time, you can move or copy multiple elements in the same operation
- You can remove an element from a diagram by selecting it, then pressing the **Delete** key
- You can also place an element onto an open diagram by locating the element in the Project Browser, then dragging it onto the diagram

### **2.3.9.5 Connect to a Different Element on a Diagram**

When creating connectors in your diagrams, at some point you might connect the wrong pair of elements. Enterprise Architect provides a simple method to enable you to move the end of a connector from one element to another; that is:

- Change the source element of a connector from the current element to another
- Change the target element of a connector from the current element to another

### Change a connected element

Step	Action	See also
1	In the diagram, click on the connector to select it. The connector is highlighted with 'handles' at each connector end.	
2	Move the cursor over the 'handle' at the end of the connector to reconnect. The cursor changes to a small arrow (  ) pointing upwards to the right.	
3	Click and drag the connector end to the other element to connect to. When you release the mouse button, the connector breaks from the original element and reconnects to the new element.	

#### Notes

- You can also tidy up a connection by dragging the end of the connector to a better position on the edge of the element, or move both ends at once by dragging the middle of the connector

### 2.3.10 Delete a Connector

It is possible to hide a connector on a diagram, or to delete it entirely from the model, removing the relationship between the previously connected elements.

Displaying all of the connectors on a complex diagram can make the diagram somewhat confused, so it can be useful to hide some of the connectors, to clarify a specific aspect of a more complex picture.

#### Hide or remove a connector between two elements

Step	Action	See also
1	Open the diagram containing the connector of interest. Either: <ul style="list-style-type: none"> <li>Double-click the diagram name in the Project Browser, or</li> <li>Make an already open diagram the 'active' diagram, by clicking on its tab in the Diagram View</li> </ul>	
2	Click on the connector in the diagram to select it. The selected connector is highlighted in the diagram.	
3	Either: <ul style="list-style-type: none"> <li>Press <b>Ctrl+Delete</b></li> <li>Right-click on the connector and, from the context menu, select <b>Delete Connector</b></li> <li>Press <b>Delete</b></li> </ul>	

Step	Action	See also
	<p>If you press <b>Ctrl+Delete</b>, a dialog displays prompting you to confirm that you want to delete the connector (and the relationship it describes) from the model.</p> <p>Otherwise, the Remove Connector dialog displays.</p>	
4	<p>On the Remove Connector dialog, choose one of the options:</p> <ul style="list-style-type: none"> <li>• <b>Hide the connector</b></li> <li>• <b>Delete the connector from the model</b></li> </ul> <p><b>Hide the connector</b> obscures the connector on the current diagram; you cannot see it but the relationship between the two elements still exists and is shown on other diagrams containing the two elements together, and on all reports of connectors between the elements.</p> <p><b>Delete the Connector from Model</b> removes the relationship that exists between the two elements; the connector is removed from the current diagram, from all other diagrams on which it is shown, and from all reports on connectors between the two elements.</p>	<a href="#">Hiding the Connector</a> <sup>[112]</sup> <a href="#">Relationships</a> <sup>[742]</sup>
5	<p>Optionally, tick the checkbox <b>Don't ask again</b>.</p> <p>Selecting this option prevents the Remove Connector dialog from being displayed next time you select the Delete Connector command; the command uses the setting you last used on the dialog.</p> <p>Make sure that you have selected the right option to use as your default.</p>	
6	<p>Click on the <b>OK</b> button.</p> <p>The connector disappears from the diagram.</p> <p>If you selected <b>Delete the connector from the model</b>, the relationship represented by the connector is also deleted from your model.</p>	

### Notes

- You cannot select multiple connectors - it is only possible to select one connector at a time
- If you select the **Don't ask again** checkbox on the Remove Connector dialog, you can reset this option on the Links page of the Options dialog (**Tools | Options | Links**)
- Selecting the **Hide the connector** option in the Remove Connector dialog, has the same effect as hiding the connector on the Links tab of the source element Properties dialog, or using the **Visibility | Hide Connector** context menu option

### Learn more

- To identify and reveal hidden connectors, see the [Links](#) <sup>[964]</sup> topic

### 2.3.11 Delete an Element

You can delete the elements of a model from a diagram or from the Project Browser.

#### Delete an element

Options	Link
Remove elements from a diagram: this is essentially 'hiding' the element in that particular view of the model.	<a href="#">Remove Elements from a Diagram</a> <sup>[64]</sup>
Delete elements from a model - the element, its properties and any child elements or diagrams it has are all deleted; the element is removed from all diagrams on which it appears.	<a href="#">Delete Elements from your Model</a> <sup>[65]</sup>

#### Learn more

- [Delete Connectors](#) <sup>[62]</sup>

#### 2.3.11.1 Remove Elements From a Diagram

Removing an element from a diagram does not delete the element from the model and therefore does not alter the structure of the underlying model. When you remove an element from a diagram, you are essentially 'hiding' the element in that particular view of the model.


#### Remove elements from a diagram

Step	Action	See also
1	Open the diagram that is to be modified. Either: <ul style="list-style-type: none"> <li>• Double-click on the diagram name in the Project Browser, or</li> <li>• Make an already open diagram the 'active' diagram, by clicking on its tab in the Diagram View</li> </ul>	
2	Either: <ul style="list-style-type: none"> <li>• Press <b>Ctrl+A</b> to select all of the elements in the diagram, or</li> <li>• Click on an element in the diagram to select it (and use <b>Ctrl+Click</b> to add further elements to your selection, or to remove elements from the selection)</li> </ul> <p>The selected elements are highlighted in the diagram.</p>	
3	Either: <ul style="list-style-type: none"> <li>• Press <b>Delete</b>, or</li> </ul>	



Step	Action	See also
	<ul style="list-style-type: none"> <li>Press <b>Ctrl+D</b>, or</li> <li>Right-click a selected element and from the context menu select <b>Delete &lt;element name&gt;</b> or <b>Delete Selected Elements</b></li> </ul> <p>The selected elements are removed from the diagram.</p>	

### Notes

- You can 'undo' the removal of a diagram element, by pressing **Ctrl+Z**, or by clicking the **Undo** button  in the Default Tools toolbar

### 2.3.11.2 Delete Elements From Your Model

You can delete elements from your model, working either from the Project Browser or from a diagram.

- When you delete an element from your model, the element, its properties and any child elements or diagrams it contains are all deleted.
- The deleted element is removed from all diagrams on which it appears
- All relationships that it was involved in are also deleted from the model

#### Delete one or more items from your model using the Project Browser

Step	Action	See also
1	<p>In the Project Browser, click on an item to select it.</p> <p>To select additional items, or to remove items from the selection, use <b>Ctrl+Click</b> or <b>Shift+Click</b>.</p> <p>The selected items are highlighted in the Project Browser.</p>	
2	<p>Either:</p> <ul style="list-style-type: none"> <li>Press <b>Ctrl+Delete</b> or</li> <li>Right-click on one of the selected elements, then select the <b>Delete selected item(s)</b> context menu option</li> </ul> <p>A confirmation dialog is displayed.</p>	
3	<p>Click on the <b>Yes</b> button.</p> <p>The selected items are deleted from your model.</p>	

**Delete one or more items from your model using a diagram**

Step	Action	See also
1	<p>To open the diagram containing the items to be deleted from your model, either:</p> <ul style="list-style-type: none"> <li>• Double-click on the diagram name in the Project Browser, or</li> <li>• Make an already open diagram the 'active' diagram, by clicking on its tab in the Diagram View</li> </ul>	
2	<p>Either:</p> <ul style="list-style-type: none"> <li>• Press <b>Ctrl+A</b> to select all of the elements in the diagram, or</li> <li>• Click on an element in the diagram to select it, then use <b>Ctrl+Click</b> to add or remove specific elements to and from the selection</li> </ul> <p>The selected elements are highlighted in the diagram.</p>	
3	<p>Press <b>Ctrl+Delete</b>.</p> <p>A confirmation dialog is displayed.</p>	
4	<p>Click on the <b>Yes</b> button.</p> <p>The selected elements are removed from the model.</p>	

**Notes**

- Deleting a Package completely removes the Package and all its contents - diagrams, child Packages and elements - from the model
- Deleting an element completely removes the element and its properties, connectors, child elements and child diagrams from the model, and from every diagram that contains it
- Deleting a diagram completely removes the diagram from the model, but not the diagram's component elements; they remain in the parent Package

## 2.4 User Interface Guide



The **Enterprise Architect Application Workspace** provides a powerful and flexible analysis, modeling, design and construction environment. The workspace is built from familiar components, such as toolbars, docking windows, dialogs and menus. Together these elements provide a **rich set of tools** for working with diagrams, models, **UML**, source code, executing applications (debugging and visualization), rich text documents, collaborative tools, extension technologies and much more.

### User Interface Tools

Tool	See
The Enterprise Architect <b>workspace</b> is the interface through which you create, open and display your models; the main workspace component is the central <b>Diagram View</b> , which is where you create model diagrams, view reports, scroll through lists of model elements, access the internet and even edit and debug source code.	<a href="#">The Start Page</a> <sup>[68]</sup> <a href="#">Diagram View</a> <sup>[784]</sup> <a href="#">The Web Browser</a> <sup>[170]</sup>
The <b>Main Menu</b> provides access both to high-level functions related to the project life cycle, and to project and system administration functions; at all levels of the system you can also access functions and operations using <b>context menus</b> .	<a href="#">The Main Menu</a> <sup>[77]</sup> <a href="#">Project Browser Context Menus</a> <sup>[648]</sup> <a href="#">Diagram Context Menu</a> <sup>[778]</sup>
The <b>windows</b> are used to enter and display information on your project and models; you can rearrange windows to adapt the screen space to your work habits.	<a href="#">Standard Windows</a> <sup>[125]</sup>
<b>Toolbars</b> are small sets of buttons that provide quick access to common operations, both on your project as a whole and on the individual modeling units.	<a href="#">Standard Toolbars</a> <sup>[134]</sup>
The <b>Diagram Toolbox</b> provides all the components and connectors that you use to create models using the medium of diagrams; the Toolbox automatically matches the kind of diagram you have open, from any technology that is currently active in your model.	<a href="#">The Diagram Toolbox</a> <sup>[792]</sup>
You can use keyboard <b>key combinations</b> to quickly initiate a wide range of actions throughout the system.	<a href="#">Keyboard Shortcuts</a> <sup>[171]</sup>
Enterprise Architect provides various <b>customization tools</b> that you can use to:	<a href="#">Customization</a> <sup>[151]</sup>

Tool	See
<ul style="list-style-type: none"> <li>Tailor the menus, toolbars and function keys you use in your work</li> <li>Set the visual style of the application, and</li> <li>Customize the <b>workspace layout</b></li> </ul>	<a href="#">Customize Command Sets</a> <sup>[165]</sup> <a href="#">Visual Styles</a> <sup>[168]</sup> <a href="#">Manage Workspace Layout</a> <sup>[163]</sup>
You can use several <b>navigation and search tools</b> to search for, track and locate information in your project, and filter the information so that you can explore particular views of your project rather than the entire contents.	<a href="#">Navigate, Search and Trace</a> <sup>[644]</sup> <a href="#">The Project Browser</a> <sup>[646]</sup>

#### Learn more

- [A Demonstration of Enterprise Architect in use](#) (Online Resource)

### 2.4.1 Start Page

The **Start Page** is the default jump-off point that displays when you first open Enterprise Architect. It provides a range of useful tools and resources such as commands to select from a list of **recently-opened projects**, **open** some other existing project, **create** a completely new project or copy a **template** project, connect to a **server repository**, or access a project via the **Cloud**. You can also access **online resources** and the **Enterprise Architect Community website** to further explore the possibilities of Enterprise Architect and broader aspects of modeling and the UML.

The options and commands on the Start Page are available through the **Side Buttons**, **Projects** options, **Recent** files list and web site **Resources**.

**Access** Opened by default on startup of Enterprise Architect; if not displayed, select **Window | Show Start Page**

#### Side Buttons

Option	Action	See also
<b>Learning Center</b>	Click on this icon to display the <b>Learning Center window</b> , providing a wealth of useful tutorials for a wide range of work areas, helping both new and experienced users become familiar with specific features and operations of the system.	<a href="#">Learning Center</a> <sup>[74]</sup>
<b>Help</b>	Click on this icon to display the <b>Contents</b> list, <b>Index</b> tab, <b>Search</b> tab, <b>Favorites</b> list and <b>Introduction</b> page of the Enterprise Architect integrated <b>Help</b> .	<a href="#">Help and Support</a> <sup>[38]</sup>
<b>Open Example Model</b>	Click on this icon to open the <b>EAExample</b> model provided with Enterprise Architect.  New users can explore the example model to immediately	

Option	Action	See also
	see a range of existing model structures and diagrams, several of which are constructed to help demonstrate many of Enterprise Architect's modeling features.	
<b>Keyboard Accelerators</b>	Click on this icon to open the <b>Help Keyboard</b> dialog to <b>view</b> the keyboard key combinations you can use in Enterprise Architect.  You cannot change these key combinations on this dialog, but you <b>can</b> customize them through the <b>Customize</b> dialog.	<a href="#">Keyboard Shortcuts</a> <sup>[167]</sup> <a href="#">Customize Keyboard Shortcuts</a> <sup>[167]</sup>
<b>Interface Customization</b>	Click on this icon to open the <b>Interface Customization Wizard</b> , which you can use to tailor and simplify your personal user interface to the system, and improve workflow. This can include: <ul style="list-style-type: none"> <li>• Customize the <b>window layout</b></li> <li>• Limit the available <b>menu options</b> based upon pre-defined <b>user roles</b></li> <li>• Specify which <b>technologies</b> and <b>toolboxes</b> are visible</li> </ul>	<a href="#">User Interface Customization</a> <sup>[167]</sup> <a href="#">Manage Workspace Layout</a> <sup>[163]</sup> <a href="#">Customize Command Sets</a> <sup>[165]</sup> <a href="#">Manage MDG Technologies</a> <sup>[1477]</sup>

## Projects

Option	Action	See also
<b>Manage Projects</b>	Click on this option to display the <b>Open Project</b> dialog, from which you can open an existing project, connect to a server repository, create a new project, select from a list of recent projects or remove a project from the <b>Recent</b> list.	<a href="#">Open a Project</a> <sup>[202]</sup> <a href="#">Remove Recent Projects</a> <sup>[73]</sup>
<b>Open a Project File</b>	Click on this option to display the <b>Select Enterprise Architect Project to Open</b> dialog to browse for and open an existing project file.	<a href="#">Open a Project</a> <sup>[202]</sup>
<b>Create a New Project</b>	Click on this option to create a new <b>EAP</b> or <b>FEAP</b> project file, and to add models to it using the <b>Model Wizard</b> dialog.	<a href="#">Create a New Project</a> <sup>[210]</sup> <a href="#">Model Wizard</a> <sup>[753]</sup>
<b>Copy a Base Project</b>	Click on this option to create a new project file based upon a copy of an existing project. The existing project can be one of your own, or the <b>EABase</b> file provided with the system.	<a href="#">Copy a Base Project</a> <sup>[217]</sup>
<b>Connect to Server</b>	Click on this option to display the <b>Data Link Properties</b> dialog, from which you begin to specify a <b>data source</b> to	<a href="#">Connect to a SQL Server Data Repository</a> <sup>[218]</sup>

Option	Action	See also
	<p>connect to, on one of a number of supported <b>repositories</b>.</p> <p>This feature is available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions.</p>	<p><a href="#">Connect to a MySQL Data Repository</a> <sup>[225]</sup></p> <p><a href="#">Connect to a PostgreSQL Data Repository</a> <sup>[233]</sup></p> <p><a href="#">Connect to an ASA Data Repository</a> <sup>[239]</sup></p> <p><a href="#">Connect to a Progress OpenEdge Repository</a> <sup>[248]</sup></p> <p><a href="#">Connect to an Oracle Data Repository (ODBC)</a> <sup>[255]</sup></p> <p><a href="#">Connect to an Oracle Data Repository (OLE DB)</a> <sup>[257]</sup></p>
<b>Connect to Cloud</b>	<p>Click on this option to open the <b>Cloud Connection</b> dialog, which you use to specify projects held at a remote location, and to access them through the internet or web. You work on the project using the Enterprise Architect installation on your machine.</p> <p>An advantage of working in this way is that you do not need to have the database drivers installed on your machine, as you would to work on projects held on a DBMS server.</p>	<p><a href="#">Connect to a Project Via the Cloud</a> <sup>[268]</sup></p>

### Recent

Option	Action	See also
<b>&lt;project names&gt;</b>	<p>This is a list of up to 10 Enterprise Architect projects (.EAP files, .FEAP files and DBMS server connections) that you have opened most recently.</p> <p>Click on a project name in the list to open that project.</p> <p>If necessary, you can remove projects from this list.</p>	<p><a href="#">Remove Recent Projects</a> <sup>[73]</sup></p>

### Resources

Option	Action	See also
<b>Resources Portal</b>	<p>Click on this option to open the Sparx Systems website <b>Help, Tutorials &amp; White papers</b> page and <b>Resources</b> page within the internal web browser. These pages provide access to a</p>	<p><a href="#">The Web Browser</a> <sup>[170]</sup></p>

Option	Action	See also
	wide range of Enterprise Architect and UML tutorials, demonstrations, examples, base files, Add-Ins and general information.	
<b>Quick Start Video Demos</b>	Click on this option to open the Sparx Systems website <b>Product Demonstrations</b> page within the internal web browser. This page provides a number of pre-recorded video demonstrations ranging from general overviews of Enterprise Architect to detailed instructions for using specific features.	
<b>EA &amp; UML Tutorials</b>	Click on this option to open the Sparx Systems website <b>Tutorials</b> page within the internal web browser. This page provides tutorials on general UML modeling and doing work with Enterprise Architect.	
<b>EA Overview Guide PDF</b>	Click on this option to open the <i>Enterprise Architect Reviewers Guide</i> PDF document within the internal web browser. This document provides a <b>comprehensive overview</b> of the capabilities of the most recent version of Enterprise Architect. Each section focuses on a particular aspect of the system. The document is very helpful to <b>trial users</b> who are currently evaluating Enterprise Architect.	
<b>Available MDG Technologies</b>	Click on this option to open the Sparx Systems website <b>MDG Products</b> page within the internal web browser. This page provides a brief description of, and the download links for, each supported MDG Add-In product for a selected edition of Enterprise Architect - select your installed edition from the drop-down list.  Enterprise Architect editions such as Systems Engineering, Business & Software Engineering and Ultimate will automatically register certain Add-In products; however these Add-Ins are not included in the <b>default</b> Enterprise Architect installation and must be downloaded separately.	
<b>3rd Party Extensions</b>	Click on this option to open the Sparx Systems website <b>Third Party Extensions</b> page within the internal web browser. This page provides listings of additional Enterprise Architect Add-Ins developed by partners and other members of the Enterprise Architect community.	
<b>DBMS Repository Scripts</b>	Click on this option to open the Sparx Systems website <b>Team Modeling Resources</b> page within the internal web browser. This page provides links to download the SQL scripts used to create new server-based repositories for Enterprise Architect projects.	<a href="#">Server Based Repositories</a> <sup>214</sup>

Option	Action	See also
<b>Download Latest Version</b>	<p>Click on this option to open a login portal for accessing the Sparx Systems website <b>Registered Downloads</b> page within the internal web browser. This page provides access to download the installer for the latest version of Enterprise Architect.</p> <p>When you purchase Enterprise Architect license keys, you are given a username and password for accessing this site and downloading the latest releases. These remain valid for 12 months from the date of purchase.</p>	
<b>Sparx Systems Newsletter</b>	<p>Click on this option to open the Sparx Systems website <b>Newsletters</b> page within the internal web browser. This page provides links to view newsletters previously released by Sparx Systems, or to subscribe to receive future newsletters as they are released.</p>	
<b>Webinar Registration</b>	<p>Click on this option to open the Sparx Systems website <b>Webinar Registration</b> page within the internal web browser. This page:</p> <ul style="list-style-type: none"> <li>• Describes the most recent webinar (a real-time seminar hosted via the internet) scheduled by Sparx Systems, and</li> <li>• Provides links to the registration form, for you to register your participation in the webinar</li> </ul>	
<b>Release History</b>	<p>Click on this option to open the Sparx Systems website <b>Recent Features</b> page on the internal web browser. This page provides a full history of changes in each build of Enterprise Architect.</p>	
<b>Enterprise Architect Community</b>	<p>Click on this option to open the <b>Sparx Systems Enterprise Architect Community Site</b> within the internal web browser.</p> <p>The Community Site contains a range of articles, tutorials, examples, case studies and resources provided by Sparx Systems and members of the Enterprise Architect user community.</p> <p>Registration for this site is free and provides full access to view and download published resources, and to submit comments. You can also register as an Author to submit your own articles and examples to be published on the website.</p>	<a href="#">Sparx Systems Enterprise Architect Community Site</a>

### Notes

- You can hide the Start Page, or show it again, by toggling the **Window | Show Start Page** option; this setting persists after shut down and restart of Enterprise Architect until you explicitly toggle the option again



### Learn more

- [A Quick Start Tutorial](#)<sup>[47]</sup>
- [The Web Browser](#)<sup>[170]</sup>
- [Diagram Tabs](#)<sup>[790]</sup>

## 2.4.1.1 Remove Recent Projects

On both the **File** menu and the Start Page **Recent** section, the projects that you have opened most recently are listed in the order of opening date/time up to a maximum of ten projects, with the most recently opened project at the top of the list. If the list identifies projects that you do not want to be so immediately accessible, you can remove them specifically from the list.

Removing a project from the list only removes the **hyperlink** to the project and does not remove the project from the file system or repository.

**Access**    **File | Open Project ( Ctrl+O )**  
              **Start Page | Projects | Manage Projects**

### Remove a project hyperlink from the Recent list

Step	Action	See also
1	On the Open Project dialog, in the Recent Projects panel, click on the project to be removed.	
2	Click on the <b>Remove Selection from List</b> button. The project link is immediately removed from the list.	

### Notes

- You can only remove one project link from the list at a time
- To remove a link there must be at least two projects in the list

### Learn more

- [Open a Project](#)<sup>[202]</sup>
- [Start Page](#)<sup>[68]</sup>
- [File Menu](#)<sup>[79]</sup>

### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Getting Started | User Interface | Remove Project From Recent**

## 2.4.2 Learning Center

The **Learning Center** window provides quick access to a range of context-specific task guides in a number of specific work areas, such as Getting Started, Modeling or Testing. Through the 'breadcrumb' path at the top of the Learning Center window you display the short task guides, which describe how to achieve specific objectives in a work area. The guides contain links to Help topics and other useful sources of information.

To open a guide in the Learning Center, click on the task name in the drop-down lists in the 'breadcrumb' path. The document immediately displays in the window.

The Learning Center facilities can be extended to present separate hierarchies of topics on working with any MDG Technologies loaded with Enterprise Architect. Your Technology Developers would create and incorporate these additional sets of topics.

**Access** **View | Learning Center** or  
**Alt+F1** or

Click on the **Learning Center** icon on the Start Page

(Also, when you open Enterprise Architect for the first time, the Learning Center automatically displays on the right of the screen)

### Switch between work areas

Click on the black arrow to the **right** of **Enterprise Architect** in the document path and select the required Enterprise Architect work area from the drop-down list. Click on the work area and select either a further division of the subject area or the title of a guide on that topic.

If you have additional sets of topics for in-house extensions of Enterprise Architect (Add-Ins and MDG Technologies), you can access them by clicking on the black arrow to the **left** of **Enterprise Architect** in the document path, and selecting the required product area from the drop-down list.

### Notes

- If an Add-In or MDG Technology loads its own Learning Center topics, when Enterprise Architect is shut down the Technology Learning Center topics are not retained and, in subsequent work sessions, must be reloaded; if the Technology is deactivated **during** a work session, its Learning Center topics are still available until Enterprise Architect is shut down

### Learn more

- [Create Learning Center Pages](#) 

### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Getting Started | Basic Tasks | Use the Learning Center**

### 2.4.2.1 Create Learning Center Pages

As a Technology Developer, it is possible for you to create a set of Learning Center pages to support any MDG Technology that you have developed and loaded into Enterprise Architect. You create each Learning Center page as the **Linked Document** of a **Document Artifact** element. These elements are organized into a hierarchy of **Packages**, where the Package name becomes the name of a **folder** of topics in the Learning Center. You then run two **scripts**; the first to generate the Learning Center folders and topics, and the second (as part of your Add-in or Technology activation) to import your Learning Center structure into the file location from which Enterprise Architect can load it.

The original process of developing a Learning Center Profile as part of an MDG Technology definition is no longer recommended.

### Set Up a Linked Document Template for Learning Center topics

Whilst not mandatory, it is recommended that you create one or more Linked Document templates for your Learning Center documents, to provide consistent standards for the layout, structure and content of your Learning Center pages.



Step	Action	See also
1	Plan the content of your Learning Center topics, and consider: <ul style="list-style-type: none"> <li>How you should present explanatory or descriptive text</li> <li>How you should lay out steps in direct-action tasks</li> <li>How you should refer and/or link to information in other documentation, web sites and file directories</li> <li>What corporate styles, standards and logos you might need to incorporate</li> </ul>	
2	Open the Resources window ( <b>Project   Resources</b> ) and create the appropriate Linked Document templates for your topics.	<a href="#">Create Linked Document Templates</a> <sup>[1098]</sup>

### Set up Package hierarchy to organize the Learning Center topics


Step	Action	See also
1	Create the root Package for your Learning Center documentation. The Package <b>name</b> will be the starting point for the documentation folder structure, as <b>Enterprise Architect</b> is for its product Learning Center topics.	
2	Display the Properties <b>window</b> for this Package element ( <b>Element   Properties Window</b> ) and, in the <b>Filename</b> field, type the file location into which to generate the Learning Center documentation.	<a href="#">Properties Window</a> <sup>[992]</sup>
3	In the Project Browser, create a child Package for each category of Learning Center documentation you intend to create, and within each of those create child Packages for any further subcategories of documentation. You can create child Packages to as many levels as necessary, but it is recommended that you try to limit the structure to fewer than five levels.  Consider also that you can have topics (Document Artifact elements) and folders (Packages) at the same level in your Learning Center document structure.	<a href="#">Add a Package</a> <sup>[772]</sup>


### Create the Learning Center Topics

You have a number of options for creating **Document Artifact** elements in a Package, but creating a **Document** diagram and adding the elements to it is a simple method that gives you an immediate visual impression of the content of a subject area.

Step	Action	See also
1	Working from the Project Browser, within each Package that is to contain Learning Center topics, create a Document diagram; that is: <ol style="list-style-type: none"> <li>1. Right-click on the Package name and select the <b>Add Diagram</b> option.</li> <li>2. On the New Diagram dialog, click on <b>Extended</b> in the <b>Select From</b> panel, and <b>Documentation</b> in the <b>Diagram Types</b> panel.</li> <li>3. Click on the <b>OK</b> button.</li> </ol>	
2	From the Diagram Toolbox, drag a Document Artifact icon onto the diagram and give it the name of the topic it will contain.  This name will display at the top of the document and in the Learning Center topic access path, where it will act as a link to open the topic.	
3	Double-click on the element and select your Learning Center document template (or any other appropriate template) from the New Linked Document from Template dialog.	
4	Add the description and/or procedure text to the document. When you have finished, save the document and repeat this procedure for the next Document Artifact element/ Learning Center topic.	<a href="#">Edit Linked Documents</a>  <sup>[1048]</sup>
5	In the Project Browser, check the order in which the Document Artifact elements are listed. This order determines the sequence of the Learning Center topics.  If you want to present the Learning Center topics in a different order, adjust the sequence of the corresponding Document Artifact elements in the Project Browser accordingly.	<a href="#">Move Objects Within a Package</a>  <sup>[57]</sup>

### Compile the Learning Center documentation

Step	Action	See also
1	In the Project Browser, click on the root-level Package for your Learning Center documentation.	
2	Select the <b>Tools   Scripting</b> menu option.  The Scripting window displays.	<a href="#">Scripting</a>  <sup>[279]</sup>

Step	Action	See also
3	Expand the <b>Local Scripts</b> folder and click on the <b>VBScript - Create Learning Center Books</b> script name.	
4	<p>Click on the <b>Run script</b> icon () in the Scripting window toolbar.</p> <p>The script generates the Learning Center documentation to the file path specified in the <b>Filename</b> field in the root Package properties.</p> <p>The script also generates:</p> <ul style="list-style-type: none"> <li>• An <b>order.txt</b> file from every Package in the document hierarchy, to define the order of display of child folders or topic names in the folder generated from that Package; you can directly edit this file to change the topic order in the Learning Center, if necessary</li> <li>• An <b>index.txt</b> file in the root folder that provides an index for every topic, derived from the GUID of the Document Artifact element from which the topic was generated; this index is used to identify the topic in references and hyperlinks to the topic, regardless of where the topic might be moved to</li> </ul>	

### Load the Learning Center documentation

The Learning Center documentation you have created for an Add-In or Technology is loaded into the Learning Center using the Repository method **AddDocumentationPath**. You would normally write this into the activation script you created to load the Technology itself, but you could also test the documentation using a simpler script that just loaded the documentation. The call resembles this:

```
{
  Repository.AddDocumentationPath('Tool Guide', 'C:\\Temp\\Tool LC', 0)
}
```

### Notes

- If the same Learning Center documentation is loaded more than once during a work session, each load is presented as a separate set of pages; the original documentation is not overwritten

### Learn more

- [Learning Center](#)<sup>74</sup>
- [Repository Class](#)<sup>2855</sup>

## 2.4.3 Main Menu

Enterprise Architect provides a wide range of high-level and administrative functions for setting up and managing your projects throughout the project life cycle. You can access these functions by mouse, through a set of **Main Menu** options at the top of the Enterprise Architect workspace.

### Main Menu Options

Option	See also
<b>File</b> - Lists options to create, open, close and save <b>projects</b> , and also to perform <b>print</b> tasks.	<a href="#">File Menu</a> <sup>[79]</sup>
<b>Edit</b> - Lists options to invoke a range of functions to apply to <b>diagram elements</b> in the currently open diagram or view.	<a href="#">Edit Menu</a> <sup>[80]</sup>
<b>View</b> - Lists options to <b>display</b> various project windows, <b>show or hide toolbars</b> , and set the <b>visual style</b> of your system workspace.	<a href="#">View Menu</a> <sup>[83]</sup>
<b>Project</b> - Provides access to tools to <b>manage</b> your project and to add to the <b>project structure</b> , such as importing and exporting data, generating documentation, version control and security.	<a href="#">Project Menu</a> <sup>[87]</sup>
<b>Diagram</b> - Lists options to configure <b>diagram properties</b> and options and to save diagram images to file.	<a href="#">Diagram Menu</a> <sup>[93]</sup>
<b>Element</b> - Lists options to configure and access <b>element features</b> and details, control element layout, generate documentation and work on source code for the element.	<a href="#">Element Menu</a> <sup>[97]</sup>
<b>Tools</b> - Provides access to a range of <b>tools</b> related to code engineering, performing transformations, creating technologies, spell checking, customization of features and setting operational options.	<a href="#">Tools Menu</a> <sup>[104]</sup>
<b>Analyzer</b> - Provides access to a range of functions to perform <b>build, debug, simulation</b> and <b>profiling</b> operations on your model code.	<a href="#">Analyzer Menu</a> <sup>[112]</sup>
<b>Extensions</b> - Lists options to connect to, display information on, work with, export and manage your <b>Add-Ins</b> .	<a href="#">Extensions Menu</a> <sup>[117]</sup>
<b>Settings</b> - Provides options to <b>configure</b> various properties for your overall project, such as stereotypes, Tagged Values, cardinality values, datatypes, language macros, local directories and image management.	<a href="#">Settings Menu</a> <sup>[119]</sup>
<b>Window</b> - Provides options for organizing and managing open <b>system windows</b> .	<a href="#">Window Menu</a> <sup>[121]</sup>
<b>Help</b> - Provides options for displaying the Enterprise Architect <b>Help</b> files, the Read Me file, the Enterprise Architect <b>End User License Agreement</b> , the <b>Example model</b> , and a number of <b>resources</b> on the Sparx Systems website; it also enables you to manage your <b>license keys</b> .	<a href="#">Help Menu</a> <sup>[123]</sup>

**Notes**

- If you right-click on the empty areas of the Main Menu ribbon or Toolbar ribbon, a composite context menu displays providing options to display the toolbars and all significant windows and views
- If you click and hold the vertical dotted line at the left of the Main Menu ribbon, you can drag the Main Menu to display above or below the Toolbar ribbon, on the bottom of the screen, vertically on either side of the screen, or as a floating Toolbar in any other screen position that is convenient to you, even on a separate screen from the Enterprise Architect work area

**2.4.3.1 File Menu**

When managing the project as a unit, you can create, open, close or save projects, provide desktop shortcuts to a project, or perform print tasks.

**Access** **File**

**File Management Options**

Option	Action	Shortcut	See also
<b>New Project</b>	Create a new Enterprise Architect project, using the <b>Model Wizard</b> to help you define the base technology, model types and patterns to use as a framework .	<b>Ctrl+N</b>	<a href="#">File Based Repositories</a> <sup>[210]</sup>
<b>Open Project</b>	Open an existing project, or create a new project from scratch or from a base <b>template project</b> .	<b>Ctrl+O</b>	<a href="#">Open a project</a> <sup>[202]</sup>
<b>Reload Project</b>	Reload the current project. (Use this in a multi-user environment to refresh the Project Browser).	<b>Ctrl+Shift+F1 1</b>	<a href="#">Refresh View of Shared Project</a> <sup>[308]</sup>
<b>Close Project</b>	Close the current project.		
<b>Save Project As</b>	Save the current project with a new name (as a .eap or .feap file).		<a href="#">Copy Existing Project</a> <sup>[212]</sup>
<b>Save Shortcut</b>	Create a desktop shortcut to the current project. (This option is also active in the 'Lite', read-only edition of Enterprise Architect.)		<a href="#">Project Shortcuts</a> <sup>[204]</sup> <a href="#">The Read-Only 'Lite' Edition</a> <sup>[23]</sup>
<b>Print Setup</b>	Configure your printer's settings.		
<b>Page Setup</b>	Configure the page settings to print the current diagram.		<a href="#">Set Up Diagram Page</a> <sup>[870]</sup>

Option	Action	Shortcut	See also
<b>Print Preview</b>	Preview <sup>[783]</sup> how the currently displayed diagram will print.		<a href="#">Print Preview</a> <sup>[783]</sup>
<b>Print</b>	Print the currently displayed diagram.  You can also make use of facilities to change the scale of the printed diagram (the number of pages it takes up) and to print or omit page headers and footers on the diagram.	<b>Ctrl+P</b>	<a href="#">Scale Image To Page Size</a> <sup>[871]</sup> <a href="#">Configure Diagram Display</a> <sup>[825]</sup>
<b>Print to PDF</b>	Save the currently displayed diagram as a PDF file. A browser displays to select the PDF file name and directory path.		
<b>&lt;Recent Files List&gt;</b>	Select from a list of up to ten of the most recently opened projects, to re-open the selected project.		
<b>Exit</b>	Exit from Enterprise Architect.		

### 2.4.3.2 Edit Menu

When working in a diagram or view, you often need to operate on the **elements** as **units** rather than, at one extreme, changing the **internal properties** of the elements or, at the other extreme, changing the properties of the **diagram**. For such work, you can select from a range of functions.


Access **Edit**

#### Editing Options

Option	Action	Shortcut	See also
<b>Undo</b>	<b>Undo</b> the last action performed; note that some actions cannot be undone.	<b>Ctrl+Z</b>	<a href="#">Undo Last Action</a> <sup>[873]</sup>
<b>Redo</b>	<b>Re-apply</b> the last undone action.	<b>Ctrl+Y</b>	<a href="#">Redo Last Action</a> <sup>[873]</sup>
<b>Find in Project</b>	<b>Search for elements</b> containing particular text strings, in the Find in Model view.	<b>Ctrl+Alt+A</b>	<a href="#">Model Search</a> <sup>[700]</sup>
<b>Find in Files</b>	<b>Search for text</b> in code files and scripts, in the Find in Files window.	<b>Ctrl+Shift+Alt+F</b>	<a href="#">Search In Files</a> <sup>[2164]</sup>



Option	Action	Shortcut	See also
<b>Copy</b>	<ol style="list-style-type: none"> <li>1. Copy selected <b>elements</b> to the MS Windows clipboard. (To <b>paste</b> the selected elements use the <b>Paste as Link</b> or <b>Paste as New</b> options, below.)</li> <li>2. Copy <b>an image</b> of selected elements to the clipboard. If no elements are selected, the entire diagram is copied.</li> </ol> <p>The image can be saved as a bitmap or a metafile; you set the format on the Options dialog.</p>	<b>Ctrl+C</b>	<a href="#">General Options</a> <sup>[60]</sup> (page of the Options dialog)
<b>Paste as Link</b>	<b>Paste</b> copied element(s) into the diagram as an <b>element-hyperlink</b> to the original element(s).	<b>Shift+Insert</b>	<a href="#">Paste Elements</a> <sup>[82]</sup>
<b>Paste as New</b>	<b>Paste</b> copied element(s) into the diagram as a new <b>instance</b> of the original element(s).  The Paste Element(s) as New dialog displays.	<b>Ctrl+Shift+V</b>	
<b>Project Clipboard</b>	<b>Add</b> copied elements to, or <b>clear</b> copied elements from, the Enterprise Architect clipboard; select the appropriate sub-option: <ul style="list-style-type: none"> <li>• <b>Add to Project Clipboard</b> - Add the currently-selected element to the clipboard</li> <li>• <b>Clear Project Clipboard</b> - Clear the clipboard</li> </ul>	<b>Ctrl+Space</b>	
<b>Paste Image From Clipboard</b>	<b>Paste</b> the element in the Enterprise Architect clipboard into the same diagram, or a different diagram, as a <b>metafile</b> (that is, a definition of the element) as many times as is necessary.  If you paste the element into a different diagram, its classifier identifies the source diagram for the element.	<b>Ctrl+Shift+Insert</b>	
<b>Select All</b>	Select <b>all</b> elements concurrently on the current diagram.		
<b>Select By Type</b>	Specify a particular <b>type</b> of element to select on the diagram.		
<b>Clear Selection</b>	<b>Deselect</b> all elements.		
<b>Bookmark Selected</b>	<b>Bookmark</b> the selected element(s).  If the selected element is already bookmarked, this option <b>removes</b> the bookmark.	<b>Shift+Space</b>	



Option	Action	Shortcut	See also
<b>Clear All Bookmarks</b>	<b>Clear</b> bookmarks from any bookmarked elements in the current diagram.		<a href="#">Manage Bookmarks</a> 
<b>Delete Selected Element(s)</b>	<b>Delete</b> the selected elements from the diagram (but not from the model).	<b>Ctrl+D</b>	

#### 2.4.3.2.1 Paste Elements

It is possible to paste an element into a diagram from the Enterprise Architect clipboard **or** the MS Windows <sup>TM</sup> clipboard, either as a hyperlink to the original element or as a new element. The Enterprise Architect clipboard takes precedence, so you must clear that clipboard before you can paste from the MS Windows clipboard.

**Access**    **Edit | Paste As Link**  
                  **Edit | Paste As New**

#### Paste Elements

Option	Action	Shortcut	See also
<b>as Link</b>	<p>Paste the element in the buffer as a link (that is, a reference) to the element.</p> <p>If there are images in the MS Windows clipboard and none in the Enterprise Architect clipboard, you can:</p> <ul style="list-style-type: none"> <li>• Paste an image from the MS Windows clipboard into a new element as the appearance of the new element, or</li> <li>• Paste an image from the MS Windows clipboard into the diagram as a new boundary's appearance</li> </ul>	<b>Shift+Insert</b>	<a href="#">Paste from Project Browser</a> 
<b>as New</b>	<p>Paste each element in the buffer as a completely new  element.</p> <p>The Paste Element(s) as New dialog displays, listing the copied elements and providing a default name for the new instance of each element.</p> <p>If you want to apply your own element name to the new element, overwrite the <b>New Element Name</b> field.</p> <p>Each listed element defaults to selected, to be pasted into the diagram. If you do not want to paste a new instance of an element held in the clipboard, <b>deselect</b> the checkbox against it.</p> <p>If you want to include the connectors between the copied elements, select the <b>Include Connectors</b> checkbox.</p>	<b>Ctrl+Shift+V</b>	

### 2.4.3.3 View Menu

During your work on your models, you might use any of a number of Enterprise Architect windows and toolbars. You can display these windows and toolbars individually or in defined configurations, as well as set the visual style of your workspace.

#### Access **View**

#### View Options

Option	Action	Shortcut	See also
<b>Project Browser</b>	Browse and explore your models in the <b>Project Browser</b> ; this shows the hierarchical arrangement of packages, elements and features within the project.	<b>Alt+0</b>	<a href="#">The Project Browser</a> <sup>[646]</sup>
<b>Package Browser</b>	Display the elements of the current package as a context-sensitive, editable table, in the <b>Package Browser</b> .	<b>Ctrl+A</b>	<a href="#">Package Browser</a> <sup>[673]</sup>
<b>Specification Manager</b>	Display the <b>Specification Manager</b> , a <b>document-based</b> interface to a selected Package in the model, providing the means of creating and reviewing elements as <b>text</b> representations of objects in the Project Browser.		<a href="#">Specification Manager</a> <sup>[1728]</sup>
<b>Model Views</b>	Create filtered views of selected areas or aspects of your project, using the <b>Model Views</b> window.	<b>Ctrl+Shift+5</b>	<a href="#">Model Views</a> <sup>[686]</sup>
<b>Traceability</b>	Trace the relationships of an element through the project, using the <b>Traceability</b> window.	<b>Ctrl+Shift+4</b>	<a href="#">The Traceability Window</a> <sup>[725]</sup>
<b>Team Review</b>	Engage in discussion and review of the project within your development team community, through the portal of the <b>Team Review</b> window, a company-internal discussion forum.	<b>Ctrl+Alt+U</b>	<a href="#">Team Review Tools</a> <sup>[343]</sup>
<b>Model Mail</b>	Exchange internal <b>mail messages</b> with project team members.		<a href="#">Model Mail</a> <sup>[565]</sup>
<b>Personal Tasks</b>	Record and manage the <b>work tasks</b> assigned to your identity as a defined Author on the project.		<a href="#">Personal Tasks</a> <sup>[554]</sup>
<b>Relationship Matrix</b>	Create and manage the relationships between elements, by connector type, through the		<a href="#">Relationship Matrix</a> <sup>[727]</sup>

Option	Action	Shortcut	See also
	<b>Relationship Matrix.</b>		
<b>Gap Analysis Matrix</b>	Analyze model artifacts for potential gaps in the solution, using the <b>Gap Analysis Matrix</b> .		<a href="#">Gap Analysis Matrix</a> <sup>[745]</sup>
<b>Web Browser</b>	Open the web browser page at the site you have specified on the Options dialog, in the <b>Web Home</b> field.	<b>Ctrl+Alt+W</b>	<a href="#">Internal Web Browser</a> <sup>[170]</sup> <a href="#">General Options</a> <sup>[605]</sup>
<b>Notes</b>	Create and review the explanatory text on a diagram, element, feature or connector, on the <b>Notes</b> window.	<b>Ctrl+Shift+1</b>	<a href="#">Notes</a> <sup>[1142]</sup>
<b>Tagged Values</b>	Add further properties to a modeling object beyond those supported by UML, by creating <b>Tagged Values</b> on the Tagged Values window.	<b>Ctrl+Shift+6</b>	<a href="#">Tagged Values</a> <sup>[1134]</sup>
<b>System Output</b>	Check the results of internal processing such as Execution Analysis, Validation, searching the model and scripting, on the <b>System Output</b> window. Add-Ins can also output information to this window.	<b>Ctrl+Shift+8</b>	<a href="#">The System Output Window</a> <sup>[169]</sup>
<b>Learning Center</b>	Display context-specific guides that explain common tasks you might perform in a number of work areas.	<b>Alt+F1</b>	<a href="#">Learning Center</a> <sup>[74]</sup>
<b>Toolbars</b>	Show or hide individual <b>toolbars</b> .		
<b>Visual Style</b>	Set the <b>Visual Style</b> and a number of other <b>User Interface</b> display options.		<a href="#">View Submenus</a> <sup>[86]</sup>
<b>Workspaces and Commands</b>	Set up specific configurations of the content and layout of the Enterprise Architect <b>workspace</b> , and customize the User Interface menus, using the Workspaces and Commands dialog.		<a href="#">Manage Workspace Layout</a> <sup>[163]</sup> <a href="#">Customize Command Sets</a> <sup>[165]</sup>

#### 2.4.3.3.1 View Submenus

In your work on your models, you might use various functions either individually or in combination. These functions are conveniently grouped by task type, on Toolbars that you can display if you wish to use them, or hide if you don't. You can also change certain other aspects of your work environment, such as the visual style of your Enterprise Architect user interface.


**Access** **View | Toolbars** and  
**View | Visual Style**

### Show and Hide toolbars

Option	Action	See also
<b>Default Tools</b>	Show or hide the <b>Default Tools</b> toolbar, which provides instant access to the most commonly used tools including creating and opening projects, editing and saving diagrams, printing and technology selection.	<a href="#">Default Tools Toolbar</a> <sup>[135]</sup>
<b>Project</b>	Show or hide the <b>Project</b> toolbar, which provides access to tools for creating packages, diagrams and elements, searching the Project Browser and model, managing issues and the Project Glossary, and generating model documentation.	<a href="#">Project Toolbar</a> <sup>[136]</sup>
<b>Diagram</b>	Show or hide the <b>Diagram</b> toolbar, which provides access to tools to improve the lay-out of elements and connectors on diagrams, including zooming in and out of the display.	<a href="#">Diagram Toolbar</a> <sup>[138]</sup>
<b>New Element</b>	Show or hide the <b>UML Elements</b> toolbar, through which you can add new elements to a diagram, including Notes, Text and Hyperlink elements	<a href="#">UML Elements Toolbar</a> <sup>[140]</sup>
<b>Element</b>	Show or hide the <b>Current Element</b> toolbar, through which you can view and edit the properties, operations, attributes and Tagged Values of existing elements. You can also select an element in a diagram and locate it in the Project Browser, and lock elements against editing.	<a href="#">Current Element Toolbar</a> <sup>[141]</sup>
<b>Connector</b>	Show or hide the <b>Current Connector</b> toolbar, through which you can modify the properties and style of a connector, show and hide labels and connectors, attach notes, reverse the direction of connectors, and pin each end of a connector.	<a href="#">Current Connector Toolbar</a> <sup>[143]</sup>
<b>Code Engineering</b>	Show or hide the <b>Code Generation</b> toolbar, which provides access to tools to import, generate, synchronize and open source code, and to set the default language and default database type.	<a href="#">Code Generation Toolbar</a> <sup>[144]</sup>
<b>Debug</b>	Show or hide the <b>Debug</b> toolbar, through which you can access the main windows for monitoring debug operations, and the simple commands for starting, pausing, continuing and stopping a debug run.	<a href="#">Debug Toolbar</a> <sup>[146]</sup>
<b>Build</b>	Show or hide the <b>Build</b> toolbar, through which you can configure,	<a href="#">Build Toolbar</a> <sup>[147]</sup>

Option	Action	See also
	compile and debug the last-selected build script, execute it, cancel the execution, and run the executable, test and deployment scripts.	
<b>Record</b>	Show or hide the <b>Record &amp; Analyze</b> toolbar, which provides access to the facilities for managing the recording of the execution of an application, including starting the recording, initiating both manual and automatic recording, stepping through a recording session and stopping the recording.	<a href="#">Record &amp; Analyze Toolbar</a> <sup>[148]</sup>
<b>Workspace Layouts</b>	Show or hide the <b>Workspace Layouts</b> toolbar, through which you can create, manage and apply workspace layouts.	<a href="#">Workspace Layouts Toolbar</a> <sup>[149]</sup>
<b>Status Bar</b>	Show or hide the <b>Status Bar</b> at the bottom of the screen, which provides feedback on current operations, the position of a selected element on a diagram and what keyboard controls are selected, and a zoom control for manipulating the scale of a diagram.	<a href="#">Status Bar</a> <sup>[150]</sup>

### Set the Visual Style

Option	Action	Shortcut	See also
<b>Select Visual Style</b>	Select different visual styles or <b>themes</b> for your <b>user interface</b> .		<a href="#">Visual Styles</a> <sup>[168]</sup>
<b>Animate Autohide Windows</b>	If you are using windows that have been automatically hidden, they simply appear in the workspace when you move the cursor over the window tab.  Select this option to <b>slide</b> the windows out into the workspace when you place the cursor on the window tab.		<a href="#">Auto Hide Windows</a> <sup>[132]</sup>
<b>Show Toolbar Customize Button</b>	Hide or show the down-arrow on the end of each toolbar that enables you to customize the toolbar buttons.  		<a href="#">Standard Toolbars</a> <sup>[134]</sup> <a href="#">The Customize Dialog</a> <sup>[152]</sup>
<b>Hide Diagram Caption Bar</b>	Hide or redisplay the diagram caption bar at the top or bottom of a diagram.		<a href="#">The Diagram View</a> <sup>[784]</sup>

### 2.4.3.4 Project Menu

As your project is developed, you need to **manage** that development by, for example, adding to the project structure, importing and exporting data, generating documentation, implementing version control and applying security. The **Project** menu provides access to a range of facilities for performing these management tasks and others.

Access **Project**

#### Project Management Options

Option	Action	Shortcut	See also
<b>Calendar</b>	Monitor work across the project in terms of the resources involved, the tasks being performed and the events scheduled, using the <b>Project Calendar</b> .		<a href="#">Project Calendar</a> <sup>[57]</sup>
<b>Resources</b>	Select a shortcut or re-use function to add stock elements to the current model, or patterns and elements for additional information, from the <b>Resources</b> window.	<b>Alt+6</b>	<a href="#">Resources</a> <sup>[1173]</sup>
<b>Glossary</b>	Display the glossary terms already defined in your model, in the <b>Glossary</b> view.	<b>Alt+2</b>	<a href="#">Project Glossary</a> <sup>[533]</sup>
<b>Project Status</b>	Review the tasks and issues that relate directly to the current project, and the statistics of the project, in the <b>Project Status</b> view.		<a href="#">The Project Status View</a> <sup>[525]</sup>
<b>Project Gantt View</b>	Review the allocation of work to elements in the project, focussing on either the elements that require work or the resources required to perform the work, using the <b>Project Gantt</b> view.		<a href="#">Project Task Allocation</a> <sup>[538]</sup> <a href="#">The Gantt View</a> <sup>[594]</sup>
<b>Package Baselines</b>	Store a model branch as a <b>snapshot</b> or <b>baseline</b> of the branch content at a given time.  Available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions.	<b>Ctrl+Alt+B</b>	<a href="#">Manage Baselines</a> <sup>[46]</sup>
<b>New Model</b>	Add further <b>models</b> to your project, using the <b>Model Wizard</b> .	<b>Ctrl+Shift+M</b>	<a href="#">Model Wizard</a> <sup>[753]</sup>
<b>New Package</b>	Create a new <b>Package</b> at any level of your model.	<b>Ctrl+W</b>	<a href="#">Add a Package</a> <sup>[772]</sup>
<b>New Diagram</b>	Create a new <b>diagram</b> in the current package.	<b>Ctrl+Inse</b>	<a href="#">Add new diagrams</a>

Option	Action	Shortcut	See also
		<b>rt</b>	<a href="#">[822]</a>
<b>Documentation</b>	Generate <b>document</b> and <b>web reports</b> on your model, using options on the <b>Documentation</b> submenu.		<a href="#">Documentation Submenu</a> <a href="#">[89]</a>
<b>Model Validation</b>	Set up and run <b>validation</b> operations, using options on the <b>Model Validation</b> submenu.		<a href="#">Model Validation Submenu</a> <a href="#">[89]</a>
<b>Model Import/Export</b>	<b>Import</b> model content from, and <b>export</b> content to, <b>XMI</b> and <b>CSV</b> files, using options on the <b>Import/Export</b> submenu.		<a href="#">Model Import/Export Submenu</a> <a href="#">[90]</a>
<b>Publish Model</b>	<b>Export</b> a package to any of a range of <b>XML</b> formats, using the <b>Publish Model Package</b> dialog.		<a href="#">Publish Model Package</a> <a href="#">[476]</a>
<b>Re-usable Asset Service</b>	Opens the <b>Reusable Asset Service (RAS)</b> through which you can connect to any repository using a cloud connection over the http protocol. The repository can hold a reusable intellectual property asset such as an industry standard, registry or framework.		<a href="#">Reuseable Asset Service</a> <a href="#">[282]</a>
<b>Version Control</b>	Switch on, configure and validate <b>version control</b> of your model packages, using options on the <b>Version Control</b> submenu.		<a href="#">Version Control Submenu</a> <a href="#">[91]</a>
<b>Security</b>	Switch on, set up and manage user <b>access permissions</b> in your project, using options on the <b>Security</b> submenu.		<a href="#">Security Submenu</a> <a href="#">[92]</a>
<b>Auditing</b>	Switch on, configure and monitor change <b>auditing</b> , using the <b>Audit View</b> .		<a href="#">The Audit View</a> <a href="#">[45]</a>
<b>QA Reports &amp; Metrics</b>	<p>Click on this option to compile and display a:</p> <ul style="list-style-type: none"> <li>• <b>Testing Details</b> report</li> <li>• <b>Implementation Details</b> report</li> <li>• <b>Dependency Details</b> report</li> <li>• Summary of <b>Maintenance item</b> properties, or</li> <li>• <b>Use Case Metrics</b> calculation</li> </ul> <p>These reports are all available through the <b>QA Reports</b> view.</p>		<a href="#">Testing Details Report</a> <a href="#">[2738]</a> <a href="#">Implementation Details Report</a> <a href="#">[2737]</a> <a href="#">Dependency Details Report</a> <a href="#">[2739]</a> <a href="#">Maintenance Report</a> <a href="#">[2740]</a> <a href="#">Estimating Project Size</a> <a href="#">[588]</a>



Option	Action	Shortcut	See also

#### 2.4.3.4.1 Documentation Submenu

When you need to generate **reports** on your project, you can do so in either printable document or web format, generating reports on Package content from standard or custom report templates, or you can generate reports on specific aspects of the project, such as Issues or defined Tests.

**Access** **Project | Documentation**

##### Documentation Options

Option	Action	Shortcut	See also
<b>Generate Documentation</b>	Generate a report on the contents of the currently selected package, in a number of document formats - PDF, RTF, and DocX.	<b>F8</b>	<a href="#">Document Reports</a> <sup>[2640]</sup>
<b>Publish as HTML</b>	Generate a report for the currently selected package in web format.	<b>Shift+F8</b>	<a href="#">Web Reports</a> <sup>[2744]</sup>
<b>Diagrams Only Report</b>	Generate a report on just the <b>diagrams</b> in a package.	<b>Ctrl+Shift+F8</b>	<a href="#">Diagrams Only Report</a> <sup>[2741]</sup>
<b>Testing Report</b>	Generate a report on the project's existing <b>tests</b> .		<a href="#">Test Documentation</a> <sup>[2617]</sup>
<b>Issues</b>	Generate a report on the project's <b>issues</b> .		<a href="#">Project Issues</a> <sup>[528]</sup>
<b>Glossary</b>	Generate a report on the project's <b>Glossary</b> .		<a href="#">Generate a (Glossary) Report</a> <sup>[536]</sup>

#### 2.4.3.4.2 Model Validation Submenu

As you develop your model, you can **validate** its component elements, diagrams and packages against sets of **validation rules** that you can configure to suit your requirements.

**Access** **Project | Model Validation**

##### Model Validation Options

Option	Action	Shortcut	See also
<b>Validate Selected</b>	Validate a selected element, diagram or package from the Project Browser.	<b>Ctrl+Alt+V</b>	<a href="#">Model Validation</a> <sup>[2594]</sup>
<b>Cancel Validation</b>	Cancel a currently-executing validation process.		
<b>Configure</b>	Configure the rules against which a validation checks the model data, selecting from the list of available rules.		<a href="#">Configure Model Validation</a> <sup>[2596]</sup>

#### Learn more

- [Rules Reference](#) <sup>[2597]</sup>

#### 2.4.3.4.3 Model Import/Export Submenu

In developing your model, it is possible to **export** and store modeling data from it in either **XMI** or **CSV** format, to work on in another tool, transfer to another project or hold as a snapshot of the data at a particular time. You can also make a copy of the standard or **reference** data **used** by the model to transfer to another model, and **import** both model and reference data from an XMI or CSV file into your own model.

**Access**    **Project | Model Import/Export**

#### Model Import/Export Options

Option	Action	Shortcut	See also
<b>Import Package from XMI</b>	<b>Import</b> a Package from an <b>XMI</b> (XML based) file.	<b>Ctrl+Alt+I</b>	<a href="#">Import From XMI</a> <sup>[478]</sup>
<b>Export Package to XMI</b>	<b>Export</b> the currently selected Package to an <b>XMI</b> (XML based) file.	<b>Ctrl+Alt+E</b>	<a href="#">Export To XMI</a> <sup>[475]</sup>
<b>Batch XMI Export</b>	<b>Export</b> a <b>group</b> of controlled Packages in one action.		<a href="#">Batch XMI Export</a> <sup>[492]</sup>
<b>Batch XMI Import</b>	<b>Run</b> a <b>batch import</b> of multiple Packages.		<a href="#">Batch XMI Import</a> <sup>[493]</sup>
<b>CSV Import/Export</b>	<b>Import</b> or <b>export</b> information on model elements in <b>CSV</b> format.	<b>Ctrl+Alt+C</b>	<a href="#">CSV Import</a> <sup>[503]</sup> <a href="#">CSV Export</a> <sup>[501]</sup>
<b>CSV Import/Export Specifications</b>	Set up <b>CSV</b> import and export <b>Specifications</b> .		<a href="#">CSV Specifications</a> <sup>[498]</sup>

Option	Action	Shortcut	See also
<b>Export Reference Data</b>	<b>Export reference data</b> to XML files for convenient model updating.		<a href="#">Export Reference Data</a> <sup>[376]</sup>
<b>Import Reference Data</b>	<b>Import reference data</b> from XML files for convenient model updating.		<a href="#">Import Reference Data</a> <sup>[380]</sup>

#### 2.4.3.4.4 Version Control Submenu

If you want to apply a **version control system** to your project, you can set it up and add Packages to it , and:

- Apply version control to a Package
- Modify or check version control settings
- Validate settings for all version controlled Packages
- Resynchronize the status of all version controlled Packages, with their associated version control system.
- Allow working when not connected to version control

Once the system is configured, you can apply further version control options using the **Package** context menu in the **Project Browser**.

**Access**   **Project | Version Control**

#### Version Control Options

Option	Action	Shortcut	See also
<b>Configure Current Package</b>	Specify whether this Package (and its children) is version controlled and, if so, which version control <b>configuration file</b> applies, using the <b>Package Control Options</b> dialog.	<b>Ctrl+Alt+P</b>	<a href="#">Configure Controlled Package</a> <sup>[424]</sup>
<b>Version Control Settings</b>	Specify the options required to connect to a <b>Source Code Control (SCC)</b> provider, using the Version Control Settings dialog.		<a href="#">Version Control Settings</a> <sup>[415]</sup>
<b>Validate Package Configurations</b>	Test the <b>validity</b> of the version control settings associated with each version controlled Package within the current model.		<a href="#">Validate Package Configurations</a> <sup>[443]</sup>
<b>Re-synch Statuses of All Packages</b>	<b>Resynchronize</b> the version control <b>status</b> of Packages as recorded in your project when they are out of synchronization with the version control status reported by your version control		<a href="#">Resynchronize the Status of Version Controlled Packages</a> <sup>[444]</sup>

Option	Action	Shortcut	See also
	<p>provider.</p> <p>The function acts on all version controlled Packages within the project, updating the values recorded in the project to match the values reported by the version control provider, without performing any XML import or export.</p>		
<b>Work Offline</b>	<p>Work <b>independently</b> of the version control server.</p> <p>By selecting Work Offline before loading your project, you can prevent the system from attempting to connect to a version control server that is unavailable, thereby avoiding the delays of waiting for the server to respond and any associated error messages.</p>		<a href="#">Offline Version Control</a> [392]

#### Learn more

- [Package Version Control Options](#) [427]

#### 2.4.3.4.5 Security Submenu

In managing your project, you might want to limit the users' **access** to **update functions** across the model, to prevent concurrent editing and limit the possibility of inadvertent model changes by users not designated as model authors.

#### Access **Project | Security**

#### Security Options

Option	Action	Shortcut	See also
<b>Manage Users</b>	Add, modify and remove users, including maintaining permissions to perform the various update tasks.		<a href="#">Maintain Users</a> [323]
<b>Manage Groups</b>	Add, modify and remove security groups, including maintaining permissions to perform the various update tasks.		<a href="#">Maintain Groups</a> [320]
<b>Manage Locks</b>	View and manage the element locks set by other users across the system.		<a href="#">View and Manage Locks</a> [332]
<b>Require User Lock to Edit</b>	Toggle the security policy in force.		<a href="#">Set Security Policy</a> [319]

Option	Action	Shortcut	See also
<b>Manage My Locks</b>	View and delete the locks that you have set.	<b>Ctrl+Shift+L</b>	<a href="#">Manage Your Own Locks</a> <sup>[342]</sup>
<b>Login as Another User</b>	Switch the current login to a different user ID.		
<b>Change Password</b>	Change a user's current security password.		<a href="#">Change Password</a> <sup>[334]</sup>
<b>Encrypt Password</b>	Add encryption to your password (for models created in releases of Enterprise Architect prior to Release 7.1).		<a href="#">Password Encryption</a> <sup>[333]</sup>
<b>Enable Security</b>	Enable or disable user security across the model.		<a href="#">Enable/Disable Security</a> <sup>[318]</sup>

#### Notes

- Project Security is available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions

#### Learn more

- [User Security](#) <sup>[316]</sup>

### 2.4.3.5 Diagram Menu

In creating your **diagrams**, you can make use of a wide range of facilities to add elements and connectors to each diagram, control the diagram size and layout or presentation format, configure diagram properties and save diagram images to file.

#### Access **Diagram**

#### Diagram Options

Option	Action	Shortcut	See also
<b>Diagram Toolbox</b>	Display the <b>Diagram Toolbox</b> , a panel of icons for creating elements and connectors on a diagram; you can select pages of icons for each UML, Extended and MDG Technology diagram type available in your project.	<b>Alt+5</b>	<a href="#">Diagram Toolbox</a> <sup>[792]</sup>
<b>Diagram Filters</b>	<b>Filter the display</b> of elements on a diagram, selecting the elements to show or hide, using the Diagram		<a href="#">Diagram Filters</a> <sup>[718]</sup>

Option	Action	Shortcut	See also
	Filters window.		
<b>Diagram Layout</b>	<b>Reformat</b> your diagram in one of a range of layouts, using the Layout Tools window.		<a href="#">Layout Diagrams</a> [874]
<b>Pan and Zoom</b>	<b>Pan</b> across the current diagram to display sections at greater magnification, using the Pan & Zoom window.	<b>Ctrl+Shift+N</b>	<a href="#">The Pan &amp; Zoom Window</a> [698]
<b>Find in Project Browser</b>	Locate the current diagram in the <b>Project Browser</b> window.	<b>Shift+Alt+G</b>	
<b>Show Diagram As</b>	Change the presentation of the diagram to the <b>Diagram List</b> format or the <b>Gantt Chart</b> format.		<a href="#">Diagram List</a> [684] <a href="#">The Gantt View</a> [594]
<b>Advanced</b>	Lock the current diagram, make it the user or model default, hide or reveal connectors, or save the diagram as a UML Pattern, from the <b>Diagram   Advanced</b> menu.		<a href="#">Diagram Advanced Menu</a> [95]
<b>Save</b>	Save the current <b>position</b> of all diagram <b>elements</b> .	<b>Ctrl+S</b>	
<b>Save as Image</b>	Save the diagram as an image in a <b>graphics file</b> in one of several formats, such as bitmap (.BMP), Graphics Interchange Format (.GIF) or Windows Metafile (.WMF).	<b>Ctrl+T</b>	
<b>Copy Image to Clipboard</b>	Copy an <b>image</b> of the current diagram to the <b>clipboard</b> . The image can be in metafile or bitmap format; you set the format on the Options dialog.	<b>Ctrl+B</b>	<a href="#">General Options</a> [605] (page of Options dialog)
<b>Swimlanes, Matrix and Kanban</b>	Add, modify and delete <b>swimlanes</b> or the <b>swimlanes matrix</b> for the current diagram, or apply the Kanban facilities.		<a href="#">Swimlanes</a> [852] <a href="#">Swimlanes Matrix</a> [854] <a href="#">Kanban Facilities</a> [857]
<b>Repeat Last Element</b>	Create an <b>element</b> of the same type as the last element created.	<b>Shift+F3</b>	
<b>Repeat Last Connector</b>	Create a <b>connector</b> of the same type as the last connector created.	<b>F3</b>	

Option	Action	Shortcut	See also
<b>Add Diagram Property Note</b>	Display the <b>properties note</b> for the current diagram on screen.		<a href="#">Insert Diagram Properties Note</a> <sup>[848]</sup>
<b>Snap To Grid</b>	<p>Show or hide the <b>grid</b> on the diagram, and select one of two options to position elements on the grid:</p> <ul style="list-style-type: none"> <li>• <b>Standard Grid</b> - constrains elements to the grid when they are added to diagrams</li> <li>• <b>Smart Placement</b> - places elements even distances away from other elements and spaces elements evenly; for Sequence diagrams, this helps position Notes on Messages</li> </ul> <p>If neither of these options are enabled, the elements can be placed freely on the diagram.</p>		
<b>Layout Diagram</b>	<b>Automatically lay out</b> the current diagram.		<a href="#">Layout a Diagram Automatically</a> <sup>[891]</sup>
<b>Zoom</b>	Change the <b>zoom</b> factor on the current diagram.		<a href="#">Pan and Zoom a Diagram</a> <sup>[868]</sup>
<b>Properties</b>	View and edit the <type> Diagram: <name> <b>properties</b> dialog for the current diagram.	<b>F5</b>	<a href="#">Set Diagram Properties</a> <sup>[823]</sup>

#### 2.4.3.5.1 Diagram Advanced Menu

As you develop some of your diagrams you might need to perform special operations on them, such as making them the default diagram for the model or for just your own work, or locking them to prevent change by yourself or by other users. You can also generate a Pattern from a diagram, or hide all connectors on the diagram.

**Access** **Diagram | Advanced**

#### Advanced Diagram Options

Option	Action	Shortcut	See also
<b>Lock Diagram</b>	<p>Prevent the diagram from being edited, or <b>release</b> the locked diagram for editing.</p> <p>This does <b>not</b> apply in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions, <b>if security</b> is enabled.</p>		<a href="#">Lock Model Elements</a> <sup>[336]</sup>

Option	Action	Shortcut	See also
<b>Make Model Default</b>	<p>Make the current diagram the default diagram displayed when the model is re-opened (unless you set a <b>User Default diagram</b>, which overrides the model default; see below).</p> <p>This option is also overridden by a <b>project shortcut</b>, which defines displays to present on opening the project.</p> <p>To cancel a Model Default diagram, either:</p> <ul style="list-style-type: none"> <li>• Create a dummy diagram, set it as the Model Default and delete it, or</li> <li>• Delete the original diagram (if it is no longer relevant)</li> </ul> <p>In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, <b>if security</b> is enabled you must have <b>Manage Project Settings</b> permission to set the current diagram as the model default.</p>		<a href="#">Set the Default Diagram</a> <sup>[844]</sup>  <a href="#">Project Shortcuts</a> <sup>[204]</sup>
<b>Make User Default</b>	<p>(Use in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, <b>if security</b> is enabled).</p> <p>Make the current diagram the default diagram displayed when you re-open this model.</p> <p>The User Default diagram overrides the <b>Model Default</b> diagram (see <b>Make Model Default</b>, above); this option is itself overridden by a project shortcut, which defines displays to present on opening the project.</p> <p>To cancel a User Default diagram, either:</p> <ul style="list-style-type: none"> <li>• Create a dummy diagram, set it as the User Default and delete it, or</li> <li>• Delete the original diagram (if it is no longer relevant)</li> </ul> <p>This still blocks the Model Default diagram, whilst security is enabled; to re-establish the Model Default diagram, set it as the User Default.</p>		<a href="#">Project Shortcuts</a> <sup>[204]</sup>
<b>Visible Relations</b>	Hide or show individual <b>connectors</b> for the current diagram.	<b>Ctrl+Shift t+l</b>	<a href="#">Relationship Visibility</a> <sup>[1119]</sup>
<b>Sequence Messages</b>	Change <sup>[1430]</sup> the order of the <b>communication messages</b> in the current diagram.		<a href="#">Message (Communication Diagrams)</a> <sup>[1428]</sup>



Option	Action	Shortcut	See also
<b>Change Type</b>	Change the <b>type</b> of the current diagram.		<a href="#">Change Diagram Type</a> [840]
<b>Save UML Pattern</b> (Not available for Sequence diagrams)	Save the current diagram as a <b>UML Pattern</b> .		<a href="#">Create a Pattern</a> [1464]

#### Learn more

- [Diagram Menu](#) [93]

### 2.4.3.6 Element Menu

In creating your **elements**, you can make use of a wide range of facilities to add features and properties to each element, control each element's appearance and generate code from the element (if a Class or Interface).

#### Access **Element**

#### Element Management Options

Option	Action	Shortcut	See also
<b>Properties Window</b>	Edit the common properties of the selected element in a table, in the <b>Properties window</b> .	<b>Alt+1</b>	<a href="#">Properties Window</a> [992]
<b>Element Browser</b>	Explore the components of the selected element, in the <b>Element Browser window</b> .	<b>Alt+9</b>	<a href="#">The Element Browser</a> [989]
<b>Relationships</b>	List the selected element's connectors in the <b>Relationships window</b> .	<b>Ctrl+Shift+2</b>	<a href="#">The Relationships Window</a> [742]
<b>Scenarios &amp; Rules</b>	List and edit the Scenarios, internal Constraints and Requirements for the selected element, on tabs of the <b>main work space</b> .	<b>Ctrl+Shift+3</b>	<a href="#">Element Scenarios &amp; Requirements</a> [992]
<b>Testing</b>	Create and manage Unit, Integration, System, Acceptance and Scenario test records for the selected element, using the <b>Testing window</b> .	<b>Alt+3</b>	<a href="#">Working on Test Records</a> [2605]

Option	Action	Shortcut	See also
<b>Maintenance</b>	Create and manage defect, change, issue and task records for the selected element, using the <b>Maintenance window</b> .	<b>Alt+4</b>	<a href="#">Working on Maintenance Items</a> [2623]
<b>Project Management</b>	Create and manage resource allocation, effort, risk and metric records for the selected element, using the <b>Project Management window</b> .	<b>Ctrl+Shift+7</b>	<a href="#">The Project Management Window</a> [510]
<b>Discussions</b>	Create, review and contribute to discussion threads on a selected element, using the <b>Element Discussions window</b> .		<a href="#">Element Discussions</a> [365]
<b>Linked Document</b>	Create or open a <b>rich text document</b> attached to the selected element.	<b>Ctrl+Alt+D</b>	<a href="#">Create Linked Document on an Element</a> [1047]
<b>Appearance</b>	Set up an alternative appearance for the element, or change the text font size or element size, using the <b>Appearance</b> submenu.		<a href="#">Appearance Submenu</a> [99]
<b>Advanced</b>	Add Tagged Values to the selected element, and manage how the element interacts with related elements, using the <b>Advanced</b> submenu.		<a href="#">Advanced Submenu</a> [100]
<b>Find in Project Browser</b>	Find the currently selected element in the <b>Project Browser</b> .  If no <b>element</b> is selected, find the current <b>diagram</b> in the Project Browser.	<b>Alt+G</b>	
<b>Find in Diagrams</b>	List all <b>occurrences</b> of the currently selected element in any <b>diagram</b> .	<b>Ctrl+U</b>	<a href="#">Show Element Use</a> [910]
<b>Attributes</b>	View and edit the <b>attributes</b> for the selected element.	<b>F9</b>	<a href="#">General Properties of Attributes</a> [1001]
<b>Operations</b>	View and edit the <b>operations</b> (methods) for the selected element.	<b>F10</b>	<a href="#">General Properties of Operations</a> [1015]
<b>Feature Visibility</b>	Select which features and characteristics of the selected element are visible on a diagram, using the <b>Feature and Compartment Visibility</b> window.	<b>Ctrl+Shift+Y</b>	<a href="#">Feature Visibility</a> [845]

Option	Action	Shortcut	See also
<b>Inline Features</b>	Make changes to the <b>properties</b> and <b>features</b> of the element <b>within the element shape</b> on the diagram, through the element <b>Inline Features</b> submenu.		<a href="#">Inline Features Submenu</a> <sup>[103]</sup>
<b>Source Code Engineering</b>	<b>Generate source code</b> and, subsequently, synchronize the element against the code, through the element <b>Source Code Engineering</b> submenu.		<a href="#">Source Code Engineering Submenu</a> <sup>[103]</sup>
<b>Show Source Viewer</b>	View and edit the <b>source code</b> that has already been generated for the element, or write new code, in the <b>Source Code window</b> .	<b>Alt+7</b>	<a href="#">Editing Source Code</a> <sup>[2146]</sup>
<b>Open Source in External Editor</b>	Open the source code of the selected Class in an editor <b>external</b> to Enterprise Architect (defined in the <b>Editor</b> field on the Language Options page for the programming language of the Class).  Source code must have been generated, and the selected element must be a Class.	<b>F12</b>	<a href="#">Editing Source Code</a> <sup>[2146]</sup> <a href="#">Language Options</a> <sup>[2262]</sup>
<b>Properties</b>	View and edit the <b>properties</b> of the selected element, using the element <b>Properties dialog</b> .	<b>Alt+Enter</b>	<a href="#">Properties Dialog</a> <sup>[956]</sup>

#### 2.4.3.6.1 Appearance Submenu

You can **customize** the appearance of an individual element in a number of ways, such as changing the color of its background, borders and/or text. These changes are applied to the selected element only, on each diagram on which it is used. For example, if the Class element *Customer* is used on three diagrams, and you set its background color to yellow on one diagram, **all three** diagrams will show the *Customer* Class in yellow.

Access **Element | Appearance**

#### Element Appearance Options

Option	Action	Shortcut	See also
<b>Default Appearance</b>	Set the border, font and background colors and the border thickness for the selected element, as its default appearance.	<b>Ctrl+Shift+E</b>	<a href="#">Set an Element's Default Appearance</a> <sup>[927]</sup>

Option	Action	Shortcut	See also
<b>Autosize</b>	Automatically size one element or a group of elements in a diagram to the best fit.	<b>Alt+Z</b>	<a href="#">Autosize Elements</a> <sup>[85]</sup>
<b>Set Font</b>	Change the font of the text displayed on the element in a diagram.		<a href="#">Set Element Font</a> <sup>[94]</sup>
<b>Alternate Image</b>	Select an alternative image for the selected element.	<b>Ctrl+Shift+W</b>	<a href="#">Using the Image Manager</a> <sup>[86]</sup>
<b>Apply Image From Clipboard</b>	Insert the image currently held on the clipboard. (If the image is to be applied from elsewhere in Enterprise Architect, turn off the <b>Diagram Frames On Clipboard Images</b> option in the Options dialog before you capture the image, to avoid having the image applied as a labeled frame.)		<a href="#">Diagram Options</a> <sup>[60]</sup> (page of the Options dialog)

### Notes

- You can also define the default appearance of:
  - The element on the **current diagram only**, using the **Format Toolbar**
  - All** new elements, using the **Diagram | Appearance** page of the Options dialog
  - All** new elements **of a type**, using an **Element Template**
  - Various** new elements **of a type**, by defining the appearance in a **stereotype** that you apply to the element as you create it
  - A **group** of selected elements on a diagram, applying the same change to all elements in the group

### Learn more

- [Format Toolbar](#) <sup>[78]</sup>
- [Diagram Appearance Options](#) <sup>[62]</sup>
- [Standard Colors](#) <sup>[61]</sup>
- [Set Element Templates Package](#) <sup>[92]</sup>
- [UML Stereotypes](#) <sup>[145]</sup>
- [Operations on Multiple Elements](#) <sup>[95]</sup>

#### 2.4.3.6.2 Advanced Submenu

As you develop the elements in a diagram or package, you can:

- Model more complex interactions between elements, such as adding structured (embedded) elements or cross-references to other elements

- Add more specific properties, in the form of Tagged Values
- Make fundamental changes such as changing the element type
- Generate document reports on individual elements rather than on the entire contents of a package

**Access**   **Element | Advanced**

#### **Advanced Element Management Options**

Option	Action	Shortcut	See also
<b>Add Tagged Value</b>	Add a Tagged Value to the currently selected element.	<b>Ctrl+Shift+T</b>	<a href="#">Quick Start - Add Tagged Value To Elements</a> <sup>[1136]</sup>
<b>Custom References</b>	List the model elements or diagrams that refer to the element you have selected and, if you wish, access one of these elements or diagrams.	<b>Ctrl+J</b>	<a href="#">Set Up Cross References</a> <sup>[916]</sup>
<b>Rich Text Format (RTF) Report</b>	Generate a rich text format report on the currently selected <b>element</b> .	<b>F8</b>	<a href="#">RTF Documents</a> <sup>[2640]</sup>
<b>Set Parents and Interfaces</b>	Set the parent elements for the selected element, or set an interface that the selected element realizes.	<b>Ctrl+I</b>	<a href="#">Set Element Parent</a> <sup>[908]</sup>
<b>Embedded Elements</b>	Attach structural elements such as Ports and Parts to the currently selected element.	<b>Ctrl+Shift+B</b>	<a href="#">Manage Structural Elements</a> <sup>[935]</sup>
<b>Overrides &amp; Implementations</b>	Automatically override methods from parent Classes and from realized interfaces.	<b>Ctrl+Shift+O</b>	<a href="#">Override Parent Operations</a> <sup>[1023]</sup>
<b>Change Type</b>	Change the element type of the selected element.		<a href="#">Change Element Type</a> <sup>[918]</sup>

#### **2.4.3.6.3 Inline Features Submenu**

When you are editing elements and their features, and if you know what to enter without being prompted by field names, you can quickly complete your work by using **inline editing**; that is, typing in and editing the information **within the element image** on a diagram. To support this, you can also use menu options to add and delete selected features, display their properties and add linked notes to the element.

**Access**   **Element | Inline Features**  
**(Or, to display the options on a context menu, right-click on the selected feature)**

### Inline Editing Options

Click on the element and then click on the attribute, operation, element name, Maintenance item or Test item on the element before displaying the menu.

Option	Action	Shortcut	See also
<b>Edit Selected</b>	Type over or delete the text of the highlighted feature.	<b>F2</b>	
<b>View Properties</b>	Open the Properties dialog containing details of the selected feature, or of the element itself if no feature is selected.		
<b>Insert New After Selected</b>	Insert a new item in the element, of the same type as the selected feature, and immediately following the selected feature.  Type in the feature name and parameters, and press <b>Enter</b> .		
<b>Create Linked Note</b>	Add a Note element linked to the selected feature, reflecting the content of that feature.		<a href="#">Link Note to Internal Documentation</a> <sup>[924]</sup>
<b>Add Attribute</b>	Add an attribute to the current element.	<b>Ctrl+Shift+F9</b>	<a href="#">Attributes</a> <sup>[999]</sup>
<b>Add Operation</b>	Add an operation to the element.	<b>Ctrl+Shift+F10</b>	<a href="#">Operations</a> <sup>[1014]</sup>
<b>Add Other</b>	Insert other features such as Maintenance and Testing items.  To display these features on the element, select to show the appropriate compartment on the element.	<b>Ctrl+F11</b>	<a href="#">Compartment s</a> <sup>[955]</sup>  <a href="#">Define Element Characteristics</a> <sup>[828]</sup>
<b>Delete Selected from Model</b>	Delete the selected feature from the model.	<b>Ctrl+Shift+Delete</b>	

### Active keys after selecting the above options

Key	Action
<b>Enter</b>	Save current changes.

Key	Action
<b>Ctrl+Enter</b>	Save current changes and open a new slot to add a new item.
<b>Esc</b>	Cancel editing.
<b>Shift+F10</b>	Display a context menu with possible parameter values.

#### Learn more



- [In-place Editing Options](#) 

#### 2.4.3.6.4 Source Code Engineering Submenu

If you are modeling a software application developed in a particular language, you can generate the code in that language from the elements of your model and, in turn, update the model from changes to the generated code. That is, perform forward and reverse engineering at element level.

**Access**    **Element | Source Code Engineering**

#### Element Source Code Engineering Options

Option	Action	Shortcut	See also
<b>Generate Current Element</b>	Generate source code  for the currently selected element.	<b>F11</b>	<a href="#">Generate a Single Class</a> 
<b>Synchronize Current Element</b>	Synchronize the selected Class with the source code. If a problem arises, an error message displays. Otherwise the synchronization just proceeds in the background until complete.	<b>F7</b>	
<b>Batch Generate Selected Element(s)</b>	Batch generate source code for the currently selected element(s). The operation proceeds in the background until complete. If any problems arise, an error message displays.	<b>Shift+F11</b>	
<b>Batch Synchronize Selected Element(s)</b>	Batch synchronize the currently selected element(s) with source code. The operation proceeds in the background until complete. If any problems arise, an error message displays.	<b>Ctrl+R</b>	
<b>Open Source Directory</b>	Open the directory containing the source for this element. A browser displays, through which you locate the source	<b>Ctrl+Alt+Y</b>	

Option	Action	Shortcut	See also
	code file for the element. Double-click on the file to open the code in the default external viewer for the code language.		

#### Learn more

- [Software Engineering](#)<sup>[2073]</sup>

### 2.4.3.7 Tools Menu

In developing and managing your models, you can make use of a wide range of **tools** for code engineering, performing transformations, creating technologies, spell checking the model, customizing features and setting operational options.

#### Access **Tools**

#### Modeling Tool Options

Option	Action	Shortcut	See also
<b>Open Source File</b>	Open any type of external source file (code, XML, DDL) for editing.  A file browser displays, through which you locate and open the file in the default viewer/editor for the language. This could be an external source code editor or the internal Source Code viewer.	<b>Ctrl+Alt+O</b>	<a href="#">Compare Editors</a> <sup>[2148]</sup>
<b>Source Code Engineering</b>	Generate or import source code files, using options on the <b>Source Code Engineering</b> submenu.		<a href="#">Source Code Engineering Submenu</a> <sup>[106]</sup>
<b>Source Code Generation Templates</b>	Modify code generation templates using the Code Templates Editor.	<b>Ctrl+Shift+P</b>	<a href="#">The Code Template Editor</a> <sup>[1641]</sup>
<b>Document Template Designer</b>	Display the <b>Document Template Designer</b> for creating and reviewing <b>document templates</b> .		<a href="#">Design Custom Document Templates</a> <sup>[2684]</sup>
<b>Database Engineering</b>	Import database schema or generate DDL for a package, using options on the <b>Database Engineering</b> submenu.		<a href="#">Database Engineering Submenu</a> <sup>[107]</sup>
<b>Scripting</b>	Develop scripts for querying and updating the model, using the Scripting window.		<a href="#">Scripting</a> <sup>[2791]</sup>
<b>Web Services</b>	Import or generate Web Service Definition		<a href="#">Web Services</a>



Option	Action	Shortcut	See also
	Language (WSDL) files, using the <b>Web Services</b> submenu.		<a href="#">Submenu</a> <sup>[108]</sup>
<b>XML Schema</b>	Import or generate W3C XML Schema (XSD) files, using the <b>XML Schema</b> submenu.		<a href="#">XML Schema Submenu</a> <sup>[108]</sup>
<b>Model Transformation (MDA)</b>	Perform Model Driven Architecture (MDA) Transformations on elements in a diagram or in a package, using options on the <b>Model Transformation (MDA)</b> submenu.		<a href="#">Model Transformation (MDA) Submenu</a> <sup>[109]</sup>
<b>MDA Transformation Templates</b>	Modify the templates used in Model Transformations, using the Transformation Template Editor.	<b>Ctrl+Alt+H</b>	<a href="#">Edit Transformation Templates</a> <sup>[2048]</sup>
<b>Spelling Tools</b>	Configure and run the spelling checker, to check the spelling of text in a single package or across the whole model.		<a href="#">Spelling Tools</a> <sup>[109]</sup>
<b>Data Management</b>	Move, compare, manage and verify the integrity of the data in your models, using the <b>Data Management</b> submenu.		<a href="#">Data Management Submenu</a> <sup>[110]</sup>
<b>MDG Technology Import</b>	Make an MDG Technology available to: <ul style="list-style-type: none"> <li>• Yourself only, by importing the technology files into the %APPDATA% folder, or</li> <li>• Your project team members by importing the files into the model</li> </ul> (Not available in the Desktop edition.)		<a href="#">Import MDG Technologies to Model</a> <sup>[1480]</sup>
<b>Generate MDG Technology File</b>	Create an MDG Technology file using the MDG Technology Wizard.  (Not available in the Desktop edition.)		<a href="#">Create MDG Technology File</a> <sup>[1545]</sup>
<b>&lt;Customized Options&gt;</b>	Provide access to external tools that you might use during your modeling work, using options that you have defined through the Tools tab of the Customize dialog (below). For example: <ul style="list-style-type: none"> <li>• <b>ODBC Data Sources</b> - Open the Windows system ODBC Data sources administration tool to define connections to Enterprise Architect repositories, or when setting up</li> </ul>		

Option	Action	Shortcut	See also
	<p>connections to data sources for reverse engineering</p> <ul style="list-style-type: none"> <li>• <b>System Information</b> - Open the Windows System Information tool, to explore the various configurations and settings active on your current system</li> </ul>		<a href="#">The Customize Dialog</a> <sup>[152]</sup>
<b>Customize</b>	Customize your operation of the system, such as modifying the main menu, toolbars, tools and other user interface behavior, to your particular preferences using the Customize dialog.		
<b>Options</b>	Customize your general settings through the Options dialog. These options govern how a number of processes within Enterprise Architect are executed, and how various elements are displayed.	<b>Ctrl+F9</b>	<a href="#">Local Options</a> <sup>[604]</sup>

#### 2.4.3.7.1 Source Code Engineering Submenu

As you develop your model, you can **generate code** from the elements in a package (forward engineering) and create and update model elements from source code that you **import into the project** (reverse engineering) using any of a range of programming languages. That is, you perform forward and reverse engineering at **package** level.

**Access**    **Tools | Source Code Engineering**

#### Source Code Engineering Options

Option	Action	Shortcut	See also
<b>Generate Package Source Code</b>	Generate source code for the currently selected package.	<b>Ctrl+Alt+K</b>	<a href="#">Generate a Package</a> <sup>[2115]</sup>
<b>Synchronize Package Contents</b>	Synchronize the selected package with the source code.	<b>Ctrl+Alt+M</b>	<a href="#">Update Package Contents</a> <sup>[2117]</sup>
<b>Import Source Directory</b>	Import and reverse engineer an entire directory structure.	<b>Ctrl+Shift+U</b>	<a href="#">Import a Directory Structure</a> <sup>[2142]</sup>
<b>Import Binary Module</b>	Import a binary module from Java Archive, .NET PE or Intermediary Language files.		<a href="#">Import Binary Module</a> <sup>[2143]</sup>

Option	Action	Shortcut	See also
<b>Import ActionScript Files</b>	Import code written in ActionScript from a <b>.as</b> code file.		<a href="#">Import ActionScript</a>  <sup>[2139]</sup>
<b>Import C Files</b>	Import code written in ANSI C from a <b>.c</b> or <b>.h</b> code file.		<a href="#">Import ANSI C</a>  <sup>[2139]</sup>
<b>Import C# Files</b>	Import code written in C# from a <b>.cs</b> code file.		<a href="#">Import C#</a>  <sup>[2139]</sup>
<b>Import C++ Files</b>	Import code written in C++ from a <b>.h</b> , <b>.hpp</b> or <b>.hh</b> code file.		<a href="#">Import C++</a>  <sup>[2139]</sup>
<b>Import Delphi Files</b>	Import code written in Delphi from a <b>.pas</b> code file.		<a href="#">Import Delphi</a>  <sup>[2139]</sup>
<b>Import Java Files</b>	Import code written in Java from a <b>.java</b> code file.		<a href="#">Import Java</a>  <sup>[2140]</sup>
<b>Import PHP Files</b>	Import code written in PHP from a <b>.inc</b> , <b>.php</b> or <b>.php4</b> code file.		<a href="#">Import PHP</a>  <sup>[2140]</sup>
<b>Import Python Files</b>	Import code written in Python from a <b>.py</b> code file.		<a href="#">Import Python</a>  <sup>[2140]</sup>
<b>Import Visual Basic Files</b>	Import code written in Visual Basic from a <b>.frm</b> , <b>.cls</b> , <b>.bas</b> or <b>.ctl</b> code file.		<a href="#">Import Visual Basic</a>  <sup>[2140]</sup>
<b>Import VB.NET Files</b>	Import code written in VB.NET from a <b>.VB</b> code file.		<a href="#">Import VB.NET</a>  <sup>[2140]</sup>

#### Notes

- Source Code Engineering is available in the Professional, Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect

#### 2.4.3.7.2 Database Engineering Submenu

If you are modeling databases, you can import existing **database schemas** into your model (reverse engineering the schemas) or **generate data definition language (DDL) scripts** for a package (forward engineering).

**Access** **Tools | Database Engineering**

**Database Engineering Options**

Option	Action	Shortcut	See also
<b>Import DB Schema from ODBC</b>	Import a database schema from an ODBC data source.		<a href="#">Import Database Schema</a> <sup>[2376]</sup>
<b>Generate Package DDL</b>	Generate a Package DDL script to create the tables in the currently selected package.		<a href="#">Generate DDL for a Package</a> <sup>[2381]</sup>

**2.4.3.7.3 Web Services Submenu**

If you are modeling a web service with a **WSDL-stereotyped UML model**, you can import **Web Service Definition Language (WSDL) files** into the model (reverse engineering), or forward-engineer the model into **WSDL 1.1** files.

**Access** **Tools | Web Services**

**Web Services Options**

Option	Action	Shortcut	See also
<b>Import WSDL</b>	Reverse engineer a Web Service Definition Language (WSDL) file into a UML Class model.		<a href="#">Import WSDL</a> <sup>[2447]</sup>
<b>Generate WSDL</b>	Forward engineer a UML Class model into a WSDL 1.1 file.		<a href="#">Generate WSDL</a> <sup>[2445]</sup>


**2.4.3.7.4 XML Schema Submenu**

If you have developed a **UML Class model**, you can forward-engineer a package from it into a W3C **XML Schema (XSD) file**, or reverse engineer a W3C XML Schema (XSD) file into a package of your UML Class model.

**Access** **Tools | XML Schema**

**XML Schema Options**

Option	Action	Shortcut	See also
<b>Import XML Schema</b>	Reverse engineer a W3C XML Schema (XSD) file as a package of a UML Class model.		<a href="#">Import XSD</a> <sup>[2420]</sup>
<b>Generate XML</b>	Forward engineer a package in a UML Class model to a		<a href="#">Generate</a>


Option	Action	Shortcut	See also
<b>Schema</b>	W3C XML Schema (XSD) file.		<a href="#">XSD</a> 

#### 2.4.3.7.5 Model Transformation (MDA) Submenu

Whilst you might develop your model to reflect a particular domain, you can **transform** the model elements and model fragments to reflect a different domain or, typically, convert generic Platform-**Independent** Model (PIM) elements to Platform-**Specific** Model (PSM) elements, using **Model Driven Architecture (MDA) Transformations**. You can perform a transformation on elements either in a diagram or in a Package.

**Access** [Tools | Model Transformation \(MDA\)](#)

##### Model Transformation Options



Option	Action	Shortcut	See also
<b>Transform Selected Elements</b>	Perform an MDA-style transformation on the currently selected elements in a <b>diagram</b> .	<b>Ctrl+H</b>	<a href="#">Transform Elements</a> 
<b>Transform Current Package</b>	Perform an MDA-style transformation on the currently selected <b>Package</b> .	<b>Ctrl+Shift+H</b>	

#### 2.4.3.7.6 Spelling Tools

As you develop your model, you can use the system **Spell Checker** to check the spelling of text in a single package or across the whole model. You can also select from a number of options to determine what types of text are checked or ignored by the Spell Checker.

**Access** [Tools | Spelling Tools](#)

##### Spell-Checking Options

Option	Action	Shortcut	See also
<b>Spell Check Project</b>	Check the spelling across the whole project.	<b>Ctrl+F7</b>	<a href="#">Using the Spell Checker</a> 
<b>Spell Check Current Package</b>	Check the spelling in the current package, selected in the Project Browser.	<b>Ctrl+Shift+F7</b>	
<b>Spelling Options</b>	Specify a range of options to configure how the Spell Checker operates.		<a href="#">Select Spell Checker Options</a> 

Learn more

- [Spell Checking](#)<sup>[549]</sup>
- [Use Languages Other Than English](#)<sup>[552]</sup>

**2.4.3.7.7 Data Management Submenu**

As you develop your project, you might need to perform management tasks on the project data, such as transferring a complete project, sharing data between projects, comparing the data of two instances of a project or two separate projects, checking the integrity of the data in a project, updating the index statistics on your Firebird project file, or performing various tasks specific to .eap files. Such tasks are not necessarily routine, and would be performed only as the need arises.

Access    **Tools | Data Management**

Data Management Options

Option	Action	Shortcut	See also
<b>Project Transfer</b>	<b>Move a complete project</b> from one repository to another.  You cannot move a project from a source .EAP file from a version of Enterprise Architect earlier than <b>3.5.0</b> .		<a href="#">Perform a Project Data Transfer</a> <sup>[504]</sup>
<b>Shared Repository</b>	Generate an SQL script to link the current repository to selected data tables belonging to a <b>shared repository</b> , rather than using data specific to this project only.		<a href="#">Link Reference Data to a Shared Repository</a> <sup>[375]</sup>
<b>Reset IDs</b>	Reset the <b>table auto increment</b> column values.		<a href="#">Reset Table Auto Increment or Identity Columns</a> <sup>[598]</sup>
<b>Project Compare</b>	<b>Compare</b> the total <b>project sizes</b> of two projects after a major operation such as a project transfer or backup recovery.		<a href="#">Compare Projects</a> <sup>[507]</sup>
<b>Project Integrity Check</b>	Check the <b>integrity of the data</b> in your project.	<b>Shift+F9</b>	<a href="#">Check Project Data Integrity</a> <sup>[597]</sup>
<b>Run Patch</b>	Execute an <b>SQL Patch</b> .		<a href="#">Run SQL Patches</a> <sup>[607]</sup>
<b>Manage .EAP File</b>	<b>Repair, compact</b> or <b>replicate</b> your .eap files.		<a href="#">Manage .EAP File Submenu</a> <sup>[117]</sup>
<b>Manage Firebird</b>	Update Index statistics on your Firebird project		

Option	Action	Shortcut	See also
<b>File</b>	<p>file, so that the indices perform better. This is particularly useful after major changes have been made to the project.</p> <p>Select this option, identify the project file on the Browser dialog, and click on the <b>Open</b> button to process the file. Click on the <b>OK</b> button to clear the 'Completed' prompt.</p>		

#### Learn more

- [Project Maintenance](#) <sup>696</sup>

#### 2.4.3.7.7.1 Manage .EAP File Submenu

If you have created your project as a **.eap** file (and not in a DBMS repository) you occasionally might need to perform some **data management** tasks that are specific to that file format, such as repairing, compacting or replicating your .eap file.

**Access**    **Tools | Data Management | Manage .EAP File**

#### Manage .EAP Files

Option	Action	See also
<b>Repair .EAP File</b>	<p><b>Repair</b> a project that has not been closed properly and is now not opening correctly (this event rarely happens).</p> <p>Copy the project to a local drive before selecting this option. All users must be logged off the project while it is being repaired.</p>	<a href="#">Repair a Project</a> <sup>602</sup>
<b>Compact .EAP File</b>	<p><b>Compact</b> the current project, if you need to conserve space.</p> <p>All users must be logged off the project before you select this option.</p>	<a href="#">Compact a Project</a> <sup>607</sup>
<b>Make Design Master</b>	Make a <b>Design Master project</b> as the base for creating and updating <b>replicas</b> of the project.	<a href="#">Design Masters</a> <sup>317</sup>
<b>Create New Replica</b>	Create a <b>replica project</b> from the Design Master project.	<a href="#">Replication</a> <sup>310</sup> <a href="#">Design Masters</a> <sup>317</sup>
<b>Synchronize Replicas</b>	Copy changes from a replica of a project to the Design Master, or to <b>another replica</b> of the same project.	<a href="#">Synchronize Replicas</a> <sup>313</sup>
<b>Remove</b>	<b>Remove all replication features</b> , should you no longer require a	<a href="#">Remove Replication</a>

Option	Action	See also
<b>Replication</b>	model to be replicable.	<a href="#">313</a>
<b>Resolve Replication Conflicts</b>	Resolve any conflicts caused when users have separately changed the same object in two replicas that have been synchronized.	<a href="#">Resolve Conflicts</a> <a href="#">315</a>

### 2.4.3.8 Analyzer Menu

Whilst you are modeling an application, you can generate code from the model and then **analyse** the way in which it **executes**, both monitoring and recording the execution behavior using the **Execution Analyzer**. This facility builds on and processes the structures and operations of the **Model Driven Development Environment (MDDE)**, which provides tools to design, build and debug an application.

Access   **Analyzer**

#### Analyzer Options

Option	Action	Shortcut	See also
<b>Execution Analyzer</b>	Manage the <b>Analyzer scripts</b> for your model, using the <b>Execution Analyzer</b> window.	<b>Shift+F12</b>	<a href="#">Managing Analyzer Scripts</a> <a href="#">[2175]</a>
<b>Debugger</b>	Control a <b>debugging</b> session and review the <b>output</b> , using the <b>Debug</b> window.	<b>Alt+8</b>	<a href="#">Debugging</a> <a href="#">[2222]</a>
<b>Recorder</b>	Review the information <b>recorded</b> during a debug session, using the <b>Record &amp; Analyze</b> window.		<a href="#">The Recording History</a> <a href="#">[2533]</a>
<b>Simulator</b>	<b>Simulate</b> the debugging of an application during the design of that application, using the <b>Simulation</b> window.		<a href="#">Model Simulation</a> <a href="#">[2463]</a>
<b>Simulation Events</b>	Manage the <b>triggers</b> and <b>sets of events</b> in a simulation, using the <b>Simulation Events</b> window.		<a href="#">Simulation Events Window</a> <a href="#">[2504]</a>
<b>Profiler</b>	Create <b>execution profiles</b> to report on runtime aspects such as: <ul style="list-style-type: none"> <li>• What functions are most-frequently called in an application</li> <li>• Which tasks are taking longer than expected, and</li> </ul>		<a href="#">Profiling</a> <a href="#">[2555]</a>



Option	Action	Shortcut	See also
	<ul style="list-style-type: none"> <li>Which tasks are taking the longest time to execute</li> </ul> <p>This information is displayed on the <b>Profiler</b> window.</p>		
<b>Testpoint Manager</b>	Execute <b>test scripts</b> and monitor or control the output, using the <b>Testpoints</b> window.		<a href="#">Testpoint Management</a> <sup>[2577]</sup>
<b>Analyzer Workbench</b>	Create your own Class <b>workbench instances</b> and invoke methods on them, using the <b>Analyzer</b> - or <b>Object - Workbench</b> .		<a href="#">Object Workbench</a> <sup>[2567]</sup>
<b>Grammar Editor</b>	Create, edit and debug <b>context-free grammars</b> to define the <b>syntax</b> of programming languages (in Backus-Normal Form / Backus Naur Form), using the <b>Grammar Editor</b> view.		<a href="#">Grammar Framework</a> <sup>[1705]</sup>
<b>Breakpoints &amp; Events</b>	Manage <b>breakpoints</b> and <b>markers</b> for <b>debugging</b> and <b>Simulation</b> , using the <b>Breakpoints &amp; Markers</b> window.		<a href="#">Breakpoint and Marker Management</a> <sup>[2224]</sup> <a href="#">Simulation Breakpoints</a> <sup>[2477]</sup>
<b>Call Stack</b>	Review the currently-running <b>threads</b> in a <b>debugging session</b> , using the <b>Call Stack</b> .		<a href="#">Capture Current Work Environment</a> <sup>[207]</sup>
<b>Locals</b>	Review the <b>local variables</b> and their values for the current local thread, using the <b>Locals</b> window.		<a href="#">View the Local Variables</a> <sup>[2234]</sup>
<b>Watches</b>	Evaluate data items that are <b>not</b> available as <b>local variables</b> - such as data items with static Class member items - using the <b>Watches</b> window.		<a href="#">View Variables in Other Scopes</a> <sup>[2237]</sup>
<b>Memory Viewer</b>	Review a <b>process memory</b> debugging session, using the <b>Memory Viewer</b> window.		<a href="#">Inspect Process Memory</a> <sup>[2242]</sup>
<b>Modules</b>	Review details of the loaded <b>modules</b> , using the <b>Modules</b> window.		<a href="#">Show Loaded Modules</a> <sup>[2243]</sup>
<b>Debug</b>	Execute a sequence of <b>debugging commands</b> ,		<a href="#">Debug Menu</a> <sup>[114]</sup>

Option	Action	Shortcut	See also
	initiated through the <b>Debug</b> menu.		
<b>Record</b>	Execute a sequence of commands for <b>recording a debug session</b> ; these commands are initiated through the <b>Record</b> menu.		<a href="#">Record Menu</a> <sup>[115]</sup>
<b>Build</b>	Execute the <b>Build script</b> for the selected package; the script would have been previously created in the <b>Execution Analyzer</b> dialog.  The output from the script is displayed in the Build tab of the System Output window.	<b>Ctrl+Shift+F1 2</b>	<a href="#">Add Build Commands</a> <sup>[2180]</sup>
<b>Run</b>	Execute the <b>Run script</b> for the selected package; the script would have been previously created in the <b>Execution Analyzer</b> dialog.  The output from the script is displayed in the Build tab of the System Output window.	<b>Ctrl+Alt+N</b>	<a href="#">Add Run Command</a> <sup>[2217]</sup>
<b>Test</b>	Execute the <b>Test script</b> for the selected package; the script would have been previously created in the <b>Execution Analyzer</b> dialog.  The output from the script is displayed in the Build tab of the System Output window.	<b>Ctrl+Alt+T</b>	<a href="#">Add Testing Command</a> <sup>[2182]</sup>
<b>Deploy</b>	Execute the <b>Deploy script</b> for the selected package; the script would have been previously created in the <b>Execution Analyzer</b> dialog.  The output from the script is displayed in the Build tab of the System Output window.	<b>Ctrl+Shift+Alt +F12</b>	<a href="#">Add Deploy Command</a> <sup>[2218]</sup>

#### Learn more

- [Execution Analyzer](#) <sup>[2527]</sup>

#### 2.4.3.8.1 Debug Menu

As you model and test software applications, you can execute complete **debugging sessions** on the code and observe the outcome as **Sequence diagrams** generated from the Call Stack.

**Access**    **Analyzer | Debug**

#### Debugging Options








Option	Action	Shortcut	See also
<b>Debug Run</b>	Start or resume the <b>debugger</b> .  If no debug session is in progress, the active Analyzer Script is used to determine which application to run or whether the debugger should attach to an already running process.	<b>F6</b>	<a href="#">Run the Debugger</a> <small>[223]</small>
<b>Debug Pause</b>	<b>Suspend</b> execution of process being debugged.		
<b>Step Into</b>	<b>Step into</b> the current function.	<b>Shift+F6</b>	
<b>Step Over</b>	<b>Step over</b> the current function.	<b>Alt+F6</b>	
<b>Step Out</b>	<b>Step out</b> of the current function.	<b>Ctrl+F6</b>	
<b>Debug Stop</b>	<b>Stop</b> the current debug session.	<b>Ctrl+Alt+F6</b>	<a href="#">Create Sequence Diagram Of Call Stack</a> <small>[224]</small>
<b>Generate Sequence Diagram From Stack</b>	Create a <b>Sequence diagram</b> from the <b>Stack Trace History</b> .		
<b>Register as Just-In-Time Debugger</b>	Register Enterprise Architect as the <b>operating system debugger</b> , saving the currently registered debugger.  When an exception or crash occurs in an application running <b>outside</b> Enterprise Architect, the Enterprise Architect debugger is opened so that you can debug the application.  When this option is de-selected again, the previously-registered debugger is restored.		

#### 2.4.3.8.2 Record Menu

As you model, test and debug software applications, you can **record** the execution of the debugging sessions on the code, and generate **Sequence**, **Test Point** and **Collaboration Class** diagrams from the recording.

**Access**    **Analyzer | Record**

#### Debug Recording Options

Option	Action	Shortcut	See also
<b>Start Debug Recording</b>	Enter <b>manual</b> record mode; each time you step into a function, the call is captured.  When you step out, the return is captured.  This option is available when the target program is at a breakpoint.		<a href="#">The Recording History</a>  <sup>[2533]</sup> <a href="#">Recorder Toolbar</a>  <sup>[2544]</sup>
<b>Stop Debug Recording</b>	Stop the recording.  This option is available when recording is in progress.		
<b>Auto Record Thread</b>	Record the thread that is halted, and resume execution of the program; whenever the thread makes a function call or a function call exits, this information is captured to the <b>History window</b> .  The Stack Trace History, Stack tab and Source Code Editor dynamically update to reflect the current execution sequence for the thread. Stack Trace Recording ends when the thread ends or when you click on the <b>Stop</b> button.  This option is available when the target program is at a breakpoint.		
<b>Show/Hide Execution</b>	Display the executing code while the recording is in progress. The <b>Call Stack</b> is updated in real time, even while recording is in progress.  Watching this window while recording can give a quick picture of what is going on, even without producing a Sequence diagram.  This option slows recording down considerably, and is not necessary when recording is being controlled by recording markers.		<a href="#">View the Call Stack</a>  <sup>[2239]</sup>
<b>Generate Sequence Diagram</b>	Create a Sequence diagram from the <b>Record &amp; Analyze</b> window.		<a href="#">Generating Sequence Diagrams</a>  <sup>[2549]</sup>
<b>Generate Testpoint Diagram</b> <b>Generate Class Diagram</b>	Create a Testpoint or Collaboration Class diagram from the stack trace in the <b>Record &amp; Analyze</b> window.		<a href="#">Testpoint Management</a>  <sup>[2577]</sup> <a href="#">Combine Testpoints</a>  <sup>[2586]</sup> <a href="#">Recorder Toolbar</a>  <sup>[2544]</sup>

### 2.4.3.9 Extensions Menu

Whilst modeling in Enterprise Architect, you can extend and specialize the product's capabilities by using Add-Ins (extensions written in COM that interact with the Enterprise Architect user interface), and by importing XML files from specific technologies, working with them and publishing model information in XML format back to the technologies.

Menu options are provided for:

- An Add-In when the product has been loaded into the system on start-up using the Manage Add-Ins dialog (below)
- ODM, ArcGIS, BPMN 2.0 and GML when they have been enabled through the MDG Technologies dialog (**Settings | MDG Technologies**)

#### Access   **Extensions**

#### Technology Options

Option	Action	Shortcut	See also
<b>ArcGIS</b>	Display a menu of options for the built-in MDG Technology for ArcGIS, a technology supporting geospatial data modeling. Using this menu you can: <ul style="list-style-type: none"> <li>• Create new model packages using patterns</li> <li>• Import ArcGIS data models</li> <li>• Export models to schema and</li> <li>• Configure the ArcGIS technology</li> <li>• Validate your ArcGIS model</li> </ul>	<b>Ctrl+Shift +M</b>	<a href="#">Geodatabase Design for ArcGIS</a> <sup>[1944]</sup>
<b>ODM</b>	Display a menu of options for the built-in MDG Technology for Ontology Definition Metamodel (ODM), a technology for developing large-scale ontologies for project domains. Using this menu you can: <ul style="list-style-type: none"> <li>• Import OWL/RDF XML files</li> <li>• Export packages as OWL/RDF XML files</li> <li>• Define new namespaces for ODM packages and</li> <li>• Define new labels for OWL and RDF elements</li> </ul>		<a href="#">MDG Technology for ODM</a> <sup>[1971]</sup>
<b>GML</b>	Display a menu of options for the built-in MDG Technology for GML, a technology supporting geographical feature modeling. Using this menu you can: <ul style="list-style-type: none"> <li>• Generate GML Application Schema from models</li> </ul>		<a href="#">MDG Technology For GML</a> <sup>[1983]</sup>
<b>Import</b>	Display a menu of options for importing XML/XML from a variety of formats. ArcGIS, OWL/RDF and EMX are specifically supported, but a wide range of other tools' XML output can be imported into Enterprise Architect		<a href="#">ODM Commands</a> <sup>[1980]</sup> <a href="#">Import ArcGIS XML Workspace</a> <sup>[1968]</sup>

Option	Action	Shortcut	See also
	using the <b>Other Tools/Formats</b> menu option.		
<b>Publish</b>	<p>Display a menu of options for Publishing (exporting) Enterprise Architect model information in a specific format for consumption by other tools or processes.</p> <p>A wide range of output formats is supported, and you can further tailor the output by including or excluding diagram information, Enterprise Architect specific extensions and so on. You would tailor your published file for the capabilities of the tool or process expected to consume it.</p> <p>The <b>Normative XMI 2.4.1</b> option provides a quick means of exporting model information in the latest UML/XMI combination with no Enterprise Architect extension information or diagram information. This format is most useful for standards development and for base interoperability with other UML based tools.</p> <p>Note that you should not use this menu to Publish model information for consumption by other users of Enterprise Architect. For that purpose, use the <b>Project   Model Import/Export</b> options.</p>		<a href="#">Publish Model Package</a> <sup>[478]</sup> <a href="#">Model Import/Export Submenu</a> <sup>[90]</sup> <a href="#">ODM Commands</a> <sup>[1980]</sup> <a href="#">Export ArcGIS XML Workspace</a> <sup>[1961]</sup>

#### Dynamic/Configuration options for Add-Ins such as TOGAF or UPDM

Option	Action	Shortcut	See also
<b>&lt;Add-In Name(s)&gt;</b>	<p>Access the facilities of the selected Add-In, through a submenu (below).</p> <p>For example, if you have TOGAF enabled on your system, you could click on the <b>TOGAF</b> menu option and display the options available to you as a TOGAF user.</p>		
<b>Add-In Windows</b>	<p>Display a window, or list of windows, provided by any Add-Ins you have installed and enabled.</p> <p>If no windows are provided, an empty, docked Add-Ins window displays.</p>		
<b>Manage Add-Ins</b>	<p>Display the Manage Add-Ins dialog, which you use to enable or disable Add-Ins for use.</p> <p>Any technology loaded by an Add-In is automatically enabled; if you do not want to use it, you can disable it on the dialog.</p>		<a href="#">The Add-In Manager</a> <sup>[8018]</sup>

**Add-In Submenu**









Option	Action	Shortcut	See also
<b>&lt;Add-In specific options&gt;</b>	List options to perform various functions specific to the Add-In.  For example, the MDG Technology For Zachman Framework, as an Add-In, has the options <b>Open Example Model</b> and <b>Insert New Framework Model</b> .		
<b>Help</b>	Display the Help subsystem for the Add-In (if one has been provided).		
<b>About</b>	Display information on the Add-In installation, such as version, registration details and copyright statement.		

**2.4.3.10 Settings Menu**

When you set up a project, you need to configure several types of reference data, default settings and directory paths for the project, such as stereotypes, Tagged Values, cardinality values, data types, language macros, local directories and image library content. You access the facilities for configuring this data from the **Settings** menu.

**Access **Settings******Configure Project Settings**

Option	Action	Shortcut	See also
<b>UML Types</b>	Set up the stereotypes, Tagged Value Types and the cardinality list for your project, on the corresponding tabs of the UML Types dialog.		<a href="#">Stereotype Settings</a> [1147] <a href="#">Tagged Value Types</a> [1150] <a href="#">Cardinality</a> [1151]
<b>Project Types</b>	Display the options to register the people involved in the project, and the categories or status types for a number of object properties.		<a href="#">Project Types Submenu</a> [120]
<b>Namespace Roots</b>	Locate and delete model namespaces.		<a href="#">Namespaces</a> [2120]
<b>Project Template Package</b>	Configure or change the default element template directory.		<a href="#">Set Element Templates Package</a> [929]


Option	Action	Shortcut	See also
<b>Auto Names and Counters</b>	Set up automatic naming and counters for elements. The conventions you define can be applied to names or aliases, or both.		<a href="#">Set Auto Naming and Auto Counters</a> 
<b>MDG Technologies</b>	Activate and connect to MDG Technology files.		<a href="#">Manage MDG Technologies</a> 
<b>Local Directories and Paths</b>	Configure local directories and paths for each of your programming languages.		<a href="#">Local Paths</a> 
<b>Code Engineering Datatypes</b>	Add, modify and delete programming language datatypes.		<a href="#">Data Types</a> 
<b>Preprocessor Macros</b>	Add and delete preprocessor macros that are skipped during reverse engineering.		<a href="#">Language Macros</a> 
<b>Database Datatypes</b>	Add, modify and delete database datatypes.		<a href="#">Add New Datatypes</a> 
<b>Images</b>	Import and configure custom images to be applied to elements in the model.		<a href="#">Using the Image Manager</a> 
<b>Colors</b>	<p>Configure the custom colors for the project.</p> <p>There are two options:</p> <ul style="list-style-type: none"> <li>• <b>Get Project Custom Colors</b> (import from the Project file)</li> <li>• <b>Set Project Custom Colors</b> (export to the Project file)</li> </ul> <p>Custom colors are used in setting the default appearance of an element in a diagram.</p>		<a href="#">Set an Element's Default Appearance</a> 

#### 2.4.3.10.1 Project Types Submenu

The **Project Types** submenu enables you to create and manage project variables and definitions for use across the project.

Access    **Settings | Project Types**

##### Set Project Types

Option	Action	Shortcut	See also
<b>People</b>	Configure the authors, clients, resources and roles for		<a href="#">People</a> 



Option	Action	Shortcut	See also
	your project.		
<b>General Types</b>	Configure requirement types, status types, constraints and scenarios for your project.		<a href="#">General Types</a> <sup>[1158]</sup>
<b>Project Indicators</b>	Define the project indicators (risks, efforts and metrics) used in Resource Management.		<a href="#">Risk Types</a> <sup>[524]</sup> <a href="#">Effort Types</a> <sup>[522]</sup> <a href="#">Metric Types</a> <sup>[523]</sup>
<b>Maintenance</b>	Define problem types and test types.		<a href="#">Maintenance</a> <sup>[1169]</sup>
<b>Estimation Factors</b>	Configure estimation factor types (Technical Complexity Factors, Environmental Complexity Factors, and Default Hour Rate) for your project.		<a href="#">Use Case Estimation</a> <sup>[584]</sup> <a href="#">Technical Complexity Factors</a> <sup>[585]</sup> <a href="#">Environment Complexity Factors</a> <sup>[586]</sup> <a href="#">Default Hours</a> <sup>[588]</sup>

### 2.4.3.11 Window Menu

The **Window** menu provides a range of options for controlling how currently-open system windows and views are displayed and used.

Access **Window**

#### Window Control options

Option	Action	Shortcut	See also
<b>Full Screen</b>	<p>Reset the display to full screen so that only the work area and main menu are shown - no toolbars or windows.</p> <p>To return to your normal working display, either click on the <b>Full Screen</b> option again or click on the <b>Close Full Screen</b> pop-up menu option.</p> <p>You can also restore individual menus and toolbars using the <b>View</b> menu options.</p>	<b>Shift+Alt+Enter</b>	<a href="#">View Submenus</a> <sup>[84]</sup> <a href="#">Manage Workspace Layout</a> <sup>[163]</sup>

Option	Action	Shortcut	See also
<b>Save All Modified</b>	Save all changes that you have made.		
<b>Reload Current View</b>	Refresh the current view.		<a href="#">Refresh View of Shared Project</a> [308]
<b>Set Focus to Current View</b>	Make the current view the active one, so that key strokes immediately act on that view.	<b>Ctrl+Shift+0</b>	
<b>Float Current View</b>	<p>Switch the current view into a floating window that you can move around your screen(s) and dock in any convenient location.</p> <p>This option is very useful for displaying two or more diagrams or views at the same time, which makes it easier to copy or drag and drop elements between them. Other points to consider are:</p> <ul style="list-style-type: none"> <li>• The view windows are not affected by the <b>Full Screen</b> option (above); other window management options do apply to them</li> <li>• A working set that includes floating or docked views will, when activated, re-open the views at their most recent location</li> <li>• Floating or docked diagrams display the Caption Bar, but not the Format toolbar that is attached to the tabbed diagrams in the Diagram View</li> </ul> <p>You can also make a tabbed view into a floating view by simply clicking on the tab and dragging it off the row of Diagram Tabs.</p> <p>To restore a floating view to a tab in the Diagram View, right-click on its Caption Bar and select the <b>Tabbed Document</b> option.</p>		<a href="#">Working Sets</a> [56↑]  <a href="#">Diagram View</a> [78↓]
<b>Close Current View</b>	Close the current view.		
<b>Close All Except Current</b>	Close all views except the currently selected view.		
<b>Close All</b>	Close all opened windows in the main tab view.		
<b>Auto Hide Active Window</b>	Autohide [132] the window that currently has focus.	<b>Ctrl+Shift+F4</b>	<a href="#">Autohide</a>

Option	Action	Shortcut	See also
<b>Auto Hide All Docked Windows</b>	Autohide <sup>[132]</sup> all windows that are docked.		<a href="#">Windows</a> <sup>[132]</sup>
<b>Close Active Window</b>	Close the window that currently has focus.	<b>Ctrl+F4</b>	
<b>Show Start Page</b>	Show or hide the Start Page.  The setting of this option persists through shut down and restart, until you specifically change the option again.		<a href="#">The Start Page</a> <sup>[68]</sup>
<b>Always on Top</b>	Force the main Enterprise Architect window to be on top of all other windows.		

### 2.4.3.12 Help Menu

The **Help** menu provides access to the Enterprise Architect Help files, the Read Me file, the Enterprise Architect End User License Agreement and a range of features on the Sparx Systems website.

Access   **Help**

#### Help Options

Option	Action	Shortcut	See also
<b>About EA</b>	Display information about Enterprise Architect, including your registration details.		
<b>Help Contents</b>	Display the Enterprise Architect Help, starting at the <i>Introduction to Enterprise Architect</i> page.		
<b>On-Line Resources</b>	Display a submenu of options for accessing Online Resources; see below.		
<b>Open Example Model</b>	Open the <i>EAExample</i> model provided with Enterprise Architect.  New users can explore the example model to immediately see a range of existing model structures and diagrams, several of which are constructed to help demonstrate many of Enterprise Architect's modeling features.		

Option	Action	Shortcut	See also
<b>Register and Manage License Key(s)</b>	Configure and manage the license keys used to register Enterprise Architect and its Add-Ins.		<a href="#">License Management</a> <sup>[3162]</sup>
<b>View License Agreement</b>	Display the Enterprise Architect End User License Agreement.		
<b>Ordering Information</b>	Display information on how to purchase Enterprise Architect.		<a href="#">Order Enterprise Architect</a> <sup>[33]</sup>
<b>Read Me</b>	Display the <i>Readme.txt</i> file, which details the changes and enhancements in Enterprise Architect, build by build.		
<b>Keyboard Accelerator Map</b>	Display the keyboard accelerator map. You can customize your keyboard shortcuts, if required.		<a href="#">Customize Keyboard Shortcuts</a> <sup>[16]</sup>
<b>EA on the Web</b>	Open the Sparx Systems website.		<a href="#">Sparx Systems website</a>

#### On-Line Resources

Option	Action	Shortcut	See also
<b>User Forum and News</b>	Access the Enterprise Architect user forum.		<a href="#">User forum</a>
<b>Request-a-Feature</b>	Open the form for submitting a request for a feature you would like to see in Enterprise Architect.		<a href="#">Request a feature</a>
<b>Bug Report Page</b>	Open the form for reporting the details of a bug you have found in Enterprise Architect.		<a href="#">Report the details of a bug</a>
<b>Latest Version Details</b>	Open the <b>Announcements</b> folder of the user forum, providing details of the latest Enterprise Architect build.		<a href="#">Details of the latest Enterprise Architect build</a>
<b>Automation Interface</b>	Display the Enterprise Architect Automation Interface pages on the Sparx Systems website.		<a href="#">Automation Interface</a>
<b>Introducing UML</b>	Access the Sparx Systems online UML tutorials.		<a href="#">UML tutorials</a>

Option	Action	Shortcut	See also
<b>Pricing and Purchase Options</b>	Display details on purchasing or upgrading Enterprise Architect over the internet.		<a href="#">Purchase or upgrade Enterprise Architect</a>







## 2.4.4 Standard Windows









As you work on your project, creating and reviewing model structures, you can make use of a wide range of system windows that help you locate and focus on particular types of object or object properties. To make your modeling tasks even easier, you can fix these windows in a convenient place on your screen, move them around, set them as callable tabs or autohide them.

You can access these windows through:

- **Main Menu** options, such as those on the **View**, **Project**, **Diagram** or **Element** menus, or
- Options on the context menu that displays when you right-click anywhere on the **Main Menu** bar

### The Standard windows

Window	See also
<b>Project Browser</b> - The Project Browser is the primary mechanism for navigating through and exploring your model and is the starting point for many of the most important features in Enterprise Architect. It lists the model packages, diagrams, elements and element features in a hierarchical structure, reflecting the arrangement of packages and elements within your model.	<a href="#">The Project Browser</a> 
<b>Properties window</b> - The Properties window provides a convenient, concise way to view (and in some cases edit) the common properties of a selected element.	<a href="#">Properties Window</a> 
<b>Testing window</b> - The Testing window, or workspace, lists the <b>test records</b> for a selected element, ready for modification or addition. The window provides several facilities for managing the test records.	<a href="#">Working on Test Records</a> 
<b>Maintenance window</b> - The Maintenance window, or workspace, lists the <b>maintenance records</b> (defects, changes, issues and tasks) for a selected element, ready for modification or addition. The window provides several facilities for managing the maintenance records.	<a href="#">Working on Maintenance Items</a> 
<b>Diagram Toolbox</b> - The Diagram Toolbox is a panel of icons that you drag onto a diagram to create elements and connectors. Related elements and connectors are organized into Toolbox <b>pages</b> , each page containing the elements or connectors used for a particular type of diagram.	<a href="#">Diagram Toolbox</a> 
<b>Resources window</b> - The Resources window provides useful shortcuts and re-use functions for creating and applying stock components in the current model. The	<a href="#">Resources</a> 

Window	See also
<p>window contains a tree structure of Document templates, Relationship Matrix profiles, Stylesheets, UML Patterns and commonly-used model elements. The window can also provide access to UML Profiles and MDG Technologies, although this is not the recommended route to these facilities.</p>	
<p><b>Source Code Viewer</b> - The viewer displays the generated source code for a selected Class element, and provides a number of facilities for <b>editing</b> that source code.</p>	<a href="#">Editing Source Code</a> 
<p><b>Scripting window</b> - The Scripting window hosts a flexible and easy to use scripting capability to create scripts in Javascript, Microsoft JScript or VBScript. Using a built in 'Repository' object, you can programmatically inspect and/or modify elements within the currently open model. The window provides tools to edit, run, debug and manage your scripts.</p>	<a href="#">Scripting</a> 
<p><b>Debug window</b> - The Debug window provides a set of facilities for starting, managing and halting a debug session on the code generated from elements of your model.</p>	<a href="#">Run the Debugger</a> 
<p><b>Learning Center</b> - The Learning Center provides quick access to a range of context-specific task guides in a number of work areas of the system, such as Getting Started, Modeling or Testing.</p>	<a href="#">Learning Center</a> 
<p><b>Notes window</b> - Notes are the main documentation feature you use to describe an element, diagram, feature or connector; in the documentation that Enterprise Architect generates, notes feature prominently. You can record and view notes on a modeling object using the Notes window.</p>	<a href="#">Notes</a> 
<p><b>Traceability window</b> - Using the Traceability window you can quickly see what elements are immediately linked to the selected element and in what relationships, and what elements are indirectly linked via a <b>chain</b> of relationships. You can also select a diagram and explore the relationships of all elements in that diagram.</p>	<a href="#">The Traceability Window</a> 
<p><b>Tagged Values window</b> - Tagged Values are a convenient method of adding additional information to an element or connector, beyond what is directly supported by UML. A Tagged Value is the <b>value</b> of a property of a modeling item, the property being called a <b>tag</b>.</p> <p>You can create, view and edit Tagged Values for the currently-selected modeling object using the Tagged Values window.</p>	<a href="#">Tagged Values</a> 
<p><b>Project Management window</b> - You use the Project Management window to record and manage the resources, effort, risk and metrics associated with the work in developing the area of the model represented by the selected element.</p>	<a href="#">The Project Management Window</a> 

Window	See also
<b>Model Views</b> - The Model Views window provides a mechanism for representing areas of your model in any of a number of different ways, to capture a particular aspect of the structure or purpose of the model.	<a href="#">Model Views</a> <sup>[686]</sup>
<b>Element Browser</b> - You can use the Element Browser to summarize, review and manage a wide range of added-on properties of the selected element. These added-on properties include methods, attributes, Tagged Values, relationships and responsibilities.	<a href="#">The Element Browser</a> <sup>[989]</sup>
<b>Relationships window</b> - The Relationships window lists all the relationships of the currently-selected element, and shows the properties of each relationship.	<a href="#">The Relationships Window</a> <sup>[742]</sup>
<b>Scenarios &amp; Requirements window</b> - The Scenarios & Requirements window is a three-tabbed window through which you can quickly add, view, edit and delete rules applied to the selected element. The entities that impose such rules are scenarios, constraints and requirements applied to the element.	<a href="#">Element Scenarios &amp; Requirements</a> <sup>[992]</sup>
<b>Pan &amp; Zoom window</b> - The Pan and Zoom window enlarges the current diagram by a percentage you define, and displays a thumbnail view of it with a shaded rectangle. As you move the shaded rectangle with the mouse, the diagram shows the magnified items of the structure underneath the rectangle.	<a href="#">The Pan &amp; Zoom Window</a> <sup>[698]</sup>
<b>Layout Diagrams</b> - The Layout Diagrams window provides a set of <b>diagram layout formats</b> that you can apply to a diagram you have created, to automatically reorganize the layout to make it more readable.	<a href="#">Layout Diagrams</a> <sup>[874]</sup>
<b>Team Review</b> - The Team Review window provides a structured internal <b>forum</b> that your team can use to discuss aspects of the model content and development.	<a href="#">Team Review Tools</a> <sup>[343]</sup>
<b>Diagram Filters</b> - You use the Diagram Filters window to modify the display of diagram components so that relevant items are immediately identified for the reader's attention, without damaging the structure and integrity of the model.	<a href="#">Diagram Filters</a> <sup>[718]</sup>

### Notes

- On the **Testing**, **Maintenance** and **Project Management** windows, any descriptive text, history, input or results for a selected item are also displayed in the **Notes** window; you cannot edit this text in the Notes window

### Learn more

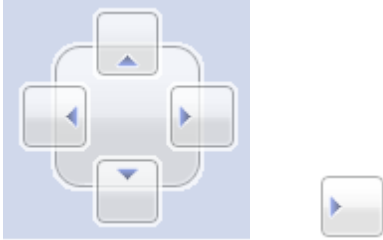
- [Code Editors](#) <sup>[133]</sup>
- [Docking Windows](#) <sup>[128]</sup>
- [Autohide Windows](#) <sup>[132]</sup>

### 2.4.4.1 Dock Windows

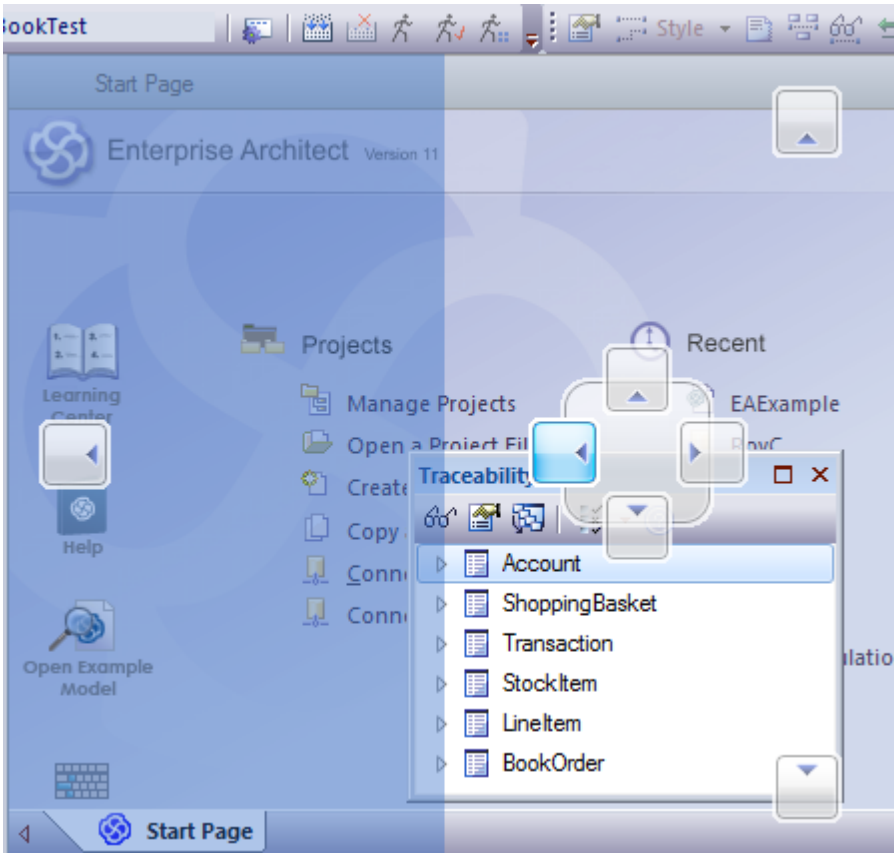
When you are doing work using one or more of the Enterprise Architect **windows**, you can re-position and/or group the windows to make them easier to use. You can easily move any window around the **application workspace** and either **dock** (anchor) it against the top, bottom, left or right edge of the workspace for long-term use, or just drop it where it is most convenient for the moment (that is, leave it **floating**).

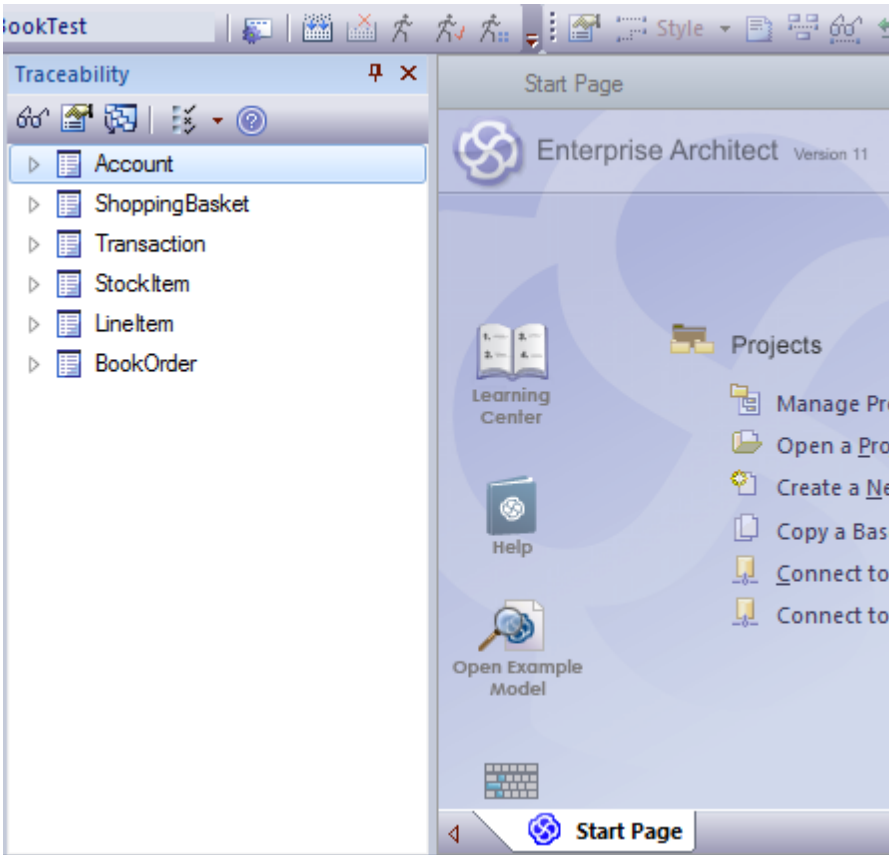
All the windows are **resizable**, so you can also drag the window margins to make the window a convenient size and shape to display either just a column or row, or a large section of the information it contains.

#### Dock a Window Against an Edge

Step	Action	See also
1	<p>Click on the title bar of the window and start to drag it towards the appropriate edge of the workspace. A set of positioning icons display, consisting of a <b>navigation compass</b> in the body of the application workspace and a <b>compass pointer</b> in the middle of each edge.</p> 	
2	<p>Drag the window onto a compass point.</p> <p>The screen display shades the area that the window will fill, once you release the mouse button.</p>	



Step	Action	See also
	 <p>The screenshot shows the Enterprise Architect Version 11 Start Page. On the left, there are icons for Learning Center, Help, and Open Example Model. In the center, there is a 'Projects' section with a list of actions: Manage Projects, Open a Project File, Create, Copy, Connect, and another Connect. On the right, a 'Recent' window is open, displaying a list of recent projects: Account, ShoppingBasket, Transaction, StockItem, LineItem, and BookOrder. The 'Account' project is currently selected. The background of the Start Page is a light blue gradient with a large, faint 'E' logo.</p>	
3	Release the mouse button to dock the window.	

Step	Action	See also
	 <p>The docked window initially fills the previously-highlighted space, but you can drag the margins to resize the window, if you wish.</p>	

Notice that the window docking is an **insert**, not an **overlay**; in the illustration, the left border of the Start Page now sits just under the 'ticked man' icon in the toolbar.

The docking behavior varies depending on whether you have already docked other windows on the screen, whether you use the navigation compass or the compass point icons, and where you drag the docked window before you release the mouse button. If you drag and release the window over:

- A point of the **navigation compass** in the **center** of the screen, the window sits **inside** existing docked windows (for example, in the illustration a window released on the bottom, top or left icons of the navigation compass would be docked to the **right** of the Traceability window)
- A **separate compass point** icon on the edge of the screen, the window is docked on the **outside** edge of the screen and existing docked windows are pushed inwards
- A point of the **navigation compass** at the **edge** of the screen (only when there is **already** a docked window there) the window is docked **within** the area occupied by the previously docked window (for example, in the illustration you could dock your new window as the left, right, top or bottom segment of the Traceability window space; the Traceability window itself is reduced in size to accommodate the new docked window, but you can then adjust the margins of the two docked windows to improve

readability)

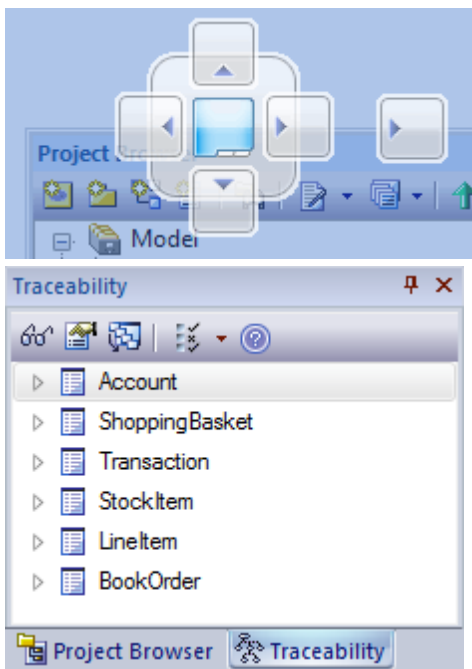
- The **center** of the **navigation compass** at the **edge** of the screen (only when there is **already** a docked window there) the window is docked as a **tab** of a window **frame**; see below

The highlighted area always indicates exactly where the moved window will be docked, so you have an indication of the effect before you release the window.

### Dock Windows into a Frame

A docked window does not overlap any other window, so if you are docking several windows you progressively fill up the workspace; however, you can avoid this by **combining** the docked windows in a single **tabbed frame** by either:

- Dragging the title bar of each window up to the title bar of the first docked window, or
- Dragging each window over the 'tabbed frame' icon in the middle of the navigation compass, when the compass is over the previously-docked window that you want to combine the dragged window with; when you release the mouse button, the window becomes a tab of the frame



The Project Browser becomes a tab -

You can move and re-dock a frame of windows as if they were a single window.

To **separate** a window from a tabbed frame, click on the window's tab at the bottom of the frame and drag it away.

### Floating Windows

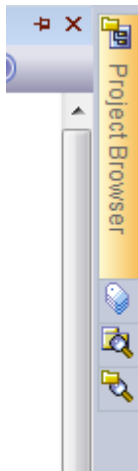
To float a window anywhere on the screen, click on its **title bar** and drag it to where you want it to sit. The navigation compass displays, but you ignore it in this situation. Floating windows **can overlay** each other, which is convenient if you want to compare information displayed in two windows. You can drag one window under or over another so that the relevant pieces of information are adjacent, do your check, then move the windows apart again.

Learn more

- [Standard Windows](#) <sup>[125]</sup>
- [Autohide Windows](#) <sup>[132]</sup>

**2.4.4.2 Autohide Windows**



If you are using docked windows, you can have them available for use but kept minimized or **hidden** until you actually need to use them. The windows are represented by tabs in the top, bottom, left or right margins of the work area.



When you move the cursor over the tab, the window displays and you can work within it. As soon as you move the cursor out of the window, it **automatically** contracts into the margin tab again. This means that you have the maximum work area for performing your main task, with the convenience of still being able to immediately use any of a wide range of other windows.

If you have 'hidden' a **tabbed frame** of docked windows, the 'open' hidden window is represented by the tab, whilst the others are indicated by an icon. In the illustration, the Project Browser is the open hidden window in the frame containing (continuing downwards) the Tagged Values window, Package Browser and Model Views window. You move the cursor over an icon to expand both the tab and the associated window.

**Autohide a Window or Set of Windows**

- To set the displayed window to 'autohide', click on the  button in the top right corner of the window; any other windows in the same frame as the open window are set to 'autohide' as well
- To turn off 'autohide' for a particular window, or set of windows within a frame, click on the  button

**Notes**

- You can only autohide docked windows; if you want to autohide a floating window, dock it against a margin first
- When you select an autohidden window, it immediately displays against the nearest work area margin; you can make the autohidden window **slide** out of and into the margin by selecting the **View | Visual Style | Animate Autohide Windows** menu option

Learn more

- [Dock Windows](#) <sup>[128]</sup>
- [Standard Windows](#) <sup>[125]</sup>

**2.4.4.3 Code Editors**

Enterprise Architect provides a number of **editors** that you can use across the development environment to maintain scripts, code and templates. Each editor has its own features, but they are all based on a common **Code Editor control**.

You can have several code editors (or files within a code editor) open at the same time, as separate tabs in the Enterprise Architect central work area (the 'Diagram View'). You can also close the editors individually or all together, leaving views of other types (such as diagrams or Document reports) still open.

Code Editors

Editor	Detail	See also
<b>Shape Script Editor</b>	When you create a <b>Shape Script</b> to alter the appearance of an element or connector through the use of a stereotype, you create and edit the script using the Shape Script Editor, which provides a number of facilities including <b>Intelli-sense</b> for Shape Script attributes and functions.	<a href="#">Shape Script Editor</a> <sup>[1584]</sup>
<b>Code Generation Template Editor</b>	You use the Code Generation Template Editor to create and edit templates that control the generation of code from your model elements. The facilities include Intelli-sense for a wide range of macros.	<a href="#">Code Generation Template Editor</a> <sup>[1641]</sup>
<b>Transformation Template Editor</b>	You use the Transformation Template Editor to create and edit templates that control the transformation of model structures from one domain to another (typically a generic domain to a code language specific domain).	<a href="#">Edit Transformation Templates</a> <sup>[2048]</sup>
<b>Custom SQL Search Filter Editor</b>	The Custom SQL Search Filter Editor is available through the <b>Model Search</b> facility, to define your own SQL searches for locating information in your model.	<a href="#">Custom SQL Search Filter Editor</a> <sup>[711]</sup>
<b>Database View Editor</b>	The Database View editor is available within the <b>View definition</b> field of a database View element Properties dialog. You create the View definition in SQL, using Intelli-sense for basic SQL keywords and functions.	<a href="#">Database View Editor</a> <sup>[2372]</sup>
<b>HTML and CSS Style Editor</b>	You use the HTML and CSS Style Editor to create web style templates from which to generate HTML reports. These templates incorporate a large number of HTML Template Fragments, which you edit to customize the reports for your specific purposes.	<a href="#">Create Web Style Templates</a> <sup>[2747]</sup>

Editor	Detail	See also
<b>Source Code Viewer</b>	You use the Source Code Viewer to both display and edit the source code files that you have generated or imported for an element. The Viewer/Editor is quite adaptable, and you can access it and use it in different ways for different types of source code file.	<a href="#">Editing Source Code</a> <sup>[2148]</sup>
<b>Script Editor</b>	Enterprise Architect provides a flexible and easy to use <b>scripting capability</b> to create scripts in Javascript, Microsoft JScript or VBScript for programmatically inspecting and/or modifying elements within the currently open model. You use the <b>Script Editor</b> to create and edit scripts, and to run, debug and stop an open script.	<a href="#">Script Editor</a> <sup>[2798]</sup>

#### Learn more

- [Diagram Tabs](#)<sup>[790]</sup>

## 2.4.5 Standard Toolbars


Enterprise Architect provides a set of **toolbars** containing convenient **shortcuts** to a wide range of common functions that you can perform on your project as a whole through to the individual modeling units. You can also **customize** the toolbars by deleting and reordering the default button set.

The toolbars are initially displayed in the toolbar banner at the top of the workspace, but you can display them at the foot of the workspace, drag and dock them within the workspace, or float them over the application; this is useful when you are using a certain set of functions a lot in a particular area.

You can also dock toolbars to the edge of the Enterprise Architect workspace by dragging them by the title bar and placing them against the appropriate edge.

#### Available toolbars

- [Default Tools Toolbar](#)<sup>[135]</sup>
- [Project Toolbar](#)<sup>[136]</sup>
- [Code Generation Toolbar](#)<sup>[144]</sup>
- [UML Elements Toolbar](#)<sup>[140]</sup>
- [Diagram Toolbar](#)<sup>[138]</sup>
- [Current Element Toolbar](#)<sup>[141]</sup>
- [Current Connector Toolbar](#)<sup>[143]</sup>
- [Build Toolbar](#)<sup>[147]</sup>
- [Record & Analyze Toolbar](#)<sup>[148]</sup>
- [Debug Toolbar](#)<sup>[146]</sup>
- [Workspace Layouts Toolbar](#)<sup>[149]</sup>
- [Status Bar](#)<sup>[150]</sup>
- [Format Toolbar](#)<sup>[783]</sup>
- [Notes Toolbar](#)<sup>[1143]</sup>

Each toolbar has a drop-down arrow at the right-hand end, , which can be enabled or hidden using the **Visual Style** sub-menu. If you click on this drop-down arrow, the **Add or Remove Buttons** option displays. Select this option to show a context menu listing the toolbars that are displayed on the same line as the current toolbar, and an option to customize toolbars - both your own and the system-provided toolbars.

You can select one of the toolbars identified on the context menu to list the icons available through that toolbar. Click on the icons as necessary to hide or show them in the toolbar.

#### Learn more

- [Customize Commands](#)<sup>[153]</sup>
- [The Visual Style Sub-menu](#)<sup>[86]</sup>
- [The Customize Dialog](#)<sup>[152]</sup>

### 2.4.5.1 Default Tools Toolbar

The **Default Tools** toolbar provides instant access to the most commonly used tools in Enterprise Architect, including file management, printing and technology facilities.



**Access** [View | Toolbars | Default Tools](#)

#### Default Tools Options

Icon/Option	Action	Shortcut	See also
<b>New Project</b>	Click on this icon to create a new Enterprise Architect project (as a .EAP or .FEAP file).	<b>Ctrl+N</b>	<a href="#">File Based Repositories</a> <sup>[210]</sup>
<b>Open Project</b>	Click on this icon to open an existing project.	<b>Ctrl+O</b>	<a href="#">Open a project</a> <sup>[202]</sup>
<b>Save Current Diagram</b>	Click on this icon to save changes to the current diagram.	<b>Ctrl+S</b>	<a href="#">Save Changes</a> <sup>[51]</sup>
<b>Cut</b>	Click on this icon to cut the selected element(s) from the diagram and copy them to the Enterprise Architect clipboard.	<b>Ctrl+X</b>	<a href="#">Move Elements Between Diagrams</a> <sup>[59]</sup>
<b>Copy</b>	Click on this icon to copy the selected element(s) from the diagram to the Enterprise Architect clipboard.	<b>Ctrl+Space</b> <b>Ctrl+C</b>	
<b>Paste</b>	Click on this icon to paste elements from the Enterprise Architect clipboard as a reference on the current diagram.	<b>Shift+Insert</b>	

		<b>Ctrl+V</b>	
<b>Undo</b>	Click on this icon to undo the last action.	<b>Ctrl+Z</b>	<a href="#">Undo Last Action</a> <sup>[873]</sup>
<b>Redo</b>	Click on this icon to redo the last undone action.	<b>Ctrl+Y</b>	<a href="#">Redo Last Action</a> <sup>[873]</sup>
<b>Print Preview</b>	Click on this icon to display the current diagram as it will appear when printed.		<a href="#">Print Preview</a> <sup>[783]</sup>
<b>Page Setup for Current Diagram</b>	Click on this icon to configure print settings for the current diagram.		
<b>Print Diagram</b>	Click on this icon to print the current diagram.	<b>Ctrl+P</b>	
<b>Package Browser</b>	Click on this icon to display the content of the current package as a list of editable elements.		<a href="#">Package Browser</a> <sup>[673]</sup>
<b>Project Search</b>	Click on this icon to display the Model Search window, to define, manage and run model searches.	<b>Ctrl+Alt+A</b>	<a href="#">Model Search</a> <sup>[700]</sup>
<b>Technology List</b>	Click on the down-arrow and select the active technology, to provide the appropriate sets of Toolbox pages and the diagram Space Bar context element menu options.		<a href="#">Manage MDG Technologies</a> <sup>[1477]</sup>
<b>Help Contents</b>	Click on this icon to access the Enterprise Architect Help.	<b>F1</b>	

### Notes

- The **Cut** and **Copy** buttons are only enabled when an item is selected on the current diagram
- You can move this toolbar to any dockable position; it retains that position in subsequent sessions
- You can hide or show the toolbar by clicking on the **View | Toolbars | Default Tools** menu option

### 2.4.5.2 Project Toolbar

The **Project** toolbar provides tools for performing routine tasks such as creating packages, diagrams and elements, searching the model and generating documentation.



**Access** **View | Toolbars | Project**



**Project Toolbar Options**

Icon/Option	Action	Shortcut	See also
<b>Reload Project</b>	Click on this option to close and reload the current project, to bring in changes made by other users sharing the project.	<b>Ctrl+Shift+F11</b>	<a href="#">Refresh View of Shared Project</a> <sup>[308]</sup>
<b>New Diagram</b>	Click on this option to create a new diagram under the currently selected package, in the Project Browser.	<b>Ctrl+Insert</b>	<a href="#">Add New Diagrams</a> <sup>[822]</sup>
<b>New Package</b>	Click on this option to create a new child package under the currently selected package, in the Project Browser.	<b>Ctrl+W</b>	<a href="#">Add a Package</a> <sup>[772]</sup>
<b>New Element</b>	Click on this option to create a new model element under the currently selected package or element, in the Project Browser.	<b>Ctrl+M</b>	<a href="#">Add Elements Directly To Packages</a> <sup>[903]</sup>
<b>Find in Project Browser</b>	Click on this option to run a simple search within the Project Browser.	<b>Ctrl+Shift+F</b>	
<b>Find in Project</b>	Click on this option to search the entire project or the selected package using pre-defined and customized searches.	<b>Ctrl+F</b>	<a href="#">Model Search</a> <sup>[700]</sup>
<b>Generate Documentation</b>	Click on this option to define and generate Document reports on the currently-selected package.	<b>F8</b>	<a href="#">Generate Documentation</a> <sup>[2644]</sup>
<b>Project Issues</b>	Click on this option to display the Project Issues dialog to review and create project issues.		<a href="#">Project Issues</a> <sup>[528]</sup>
<b>Project Glossary</b>	Click on this option to display the Glossary dialog to review and create glossary terms.		<a href="#">The Glossary Dialog</a> <sup>[535]</sup>
<b>Set Options</b>	Click on this option to display the Options dialog, through which you can personalize project display and behaviour options for your own login to the project.	<b>Ctrl+F9</b>	<a href="#">General Options</a> <sup>[605]</sup>

**Notes**

- You can move this toolbar to any dockable position; it retains that position in subsequent sessions
- You can hide or show the toolbar by clicking on the **View | Toolbars | Project** menu option

### 2.4.5.3 Diagram Toolbar

The **Diagram** toolbar provides tools to lay out and improve the appearance of diagrams, including zoom controls to better manage large diagrams and quickly identify key areas of interest.



Access    **View | Toolbars | Diagram**

#### Diagram Toolbar Options

Icon/Option	Action	Shortcut	See also
<b>Align Left</b>	Click on this option to align the <b>left</b> edges of the selected elements.	<b>Ctrl+Alt+Left Arrow</b>	
<b>Align Right</b>	Click on this option to align the <b>right</b> edges of the selected elements.	<b>Ctrl+Alt+Right Arrow</b>	
<b>Align Tops</b>	Click on this option to align the <b>top</b> edges of the selected elements.	<b>Ctrl+Alt+Up Arrow</b>	
<b>Align Bottoms</b>	Click on this option to align the <b>bottom</b> edges of the selected elements.	<b>Ctrl+Alt+Down Arrow</b>	
<b>Bring to Top</b>	Click on this option to bring the selected elements to the <b>top</b> of the Z order, in <b>front</b> of other elements on the diagram.		<a href="#">Diagram Context Menu</a> [778]
<b>Send to Back</b>	Click on this option to move the selected elements to the <b>bottom</b> of the Z order, <b>behind</b> other elements on the diagram.		
<b>Previous diagram</b>	Click on this option to display the previous open diagram.	<b>Alt+Left Arrow</b>	
<b>Next Diagram</b>	Click on this option to display the next open diagram.	<b>Alt+Right Arrow</b>	
<b>Home Diagram</b>	Click on this option to display the default Model Diagram		<a href="#">Diagram</a>

	(or default User Diagram) if one has been defined.		<a href="#">Advanced Menu</a> [95]
<b>Zoom In</b>	Click on this option to zoom into ( <b>magnify</b> ) the diagram by 10 percent.		
<b>Zoom Out</b>	Click on this option to zoom out of ( <b>shrink</b> ) the diagram by 10 percent.		
<b>Fit to View</b>	Click on this option to adjust the diagram size to fit within the Diagram View screen.		
<b>Size View to Page</b>	Click on this option to adjust the diagram size to fit within the page margins shown on the screen.		
<b>Zoom to 100%</b>	Click on this option to display the diagram content at <b>100% of default size</b> , regardless of the Diagram View screen size.		
<b>Layout Diagram</b>	(Not applicable to Behavioral diagrams) Click on this option to automatically layout the diagram in the <b>Digraph</b> layout format.		<a href="#">Digraph Layout</a> [88]
<b>Diagram Properties</b>	Click on this option to display the diagram Properties dialog, on which you can adjust the diagram display and behaviour properties.	<b>F5</b>	<a href="#">Set Diagram Properties</a> [82]
<b>Paste Appearance</b>	Click on this option to apply to the selected element a set of appearance style settings as defined for an element in the Default Appearance dialog and loaded into the Painter.		<a href="#">Set an Element's Default Appearance</a> [92] <a href="#">Changing Element Appearance</a> [94]
<b>Delete Selected</b>	Click on this option to delete the selected element(s) from the diagram.	<b>Ctrl+D</b>	

**Notes**

- Any actions that result in a change in diagram content and appearance (including Zoom) are saved as **changes** to the diagram
- The Diagram toolbar can be docked to any main window edge or floated freely in a convenient location for quick access

- You can move this toolbar to any dockable position and it retains that position in subsequent sessions
- You can hide or show the toolbar by clicking on the **View | Toolbars | Diagram** menu option

#### 2.4.5.4 UML Elements Toolbar




The **UML Elements** toolbar provides facilities for quickly and easily inserting UML and Technology **elements** into a **diagram**, as well as Notes, text, hyperlinks, diagram notes and diagram legends. You can also add **connectors** to a diagram, using this toolbar.



**Access** **View | Toolbars | New Element**

#### UML Elements Toolbar Options

Icon/Option	Action	Shortcut	See also
<b>New Element</b>	Click on this option to add a new element or connector to the current diagram.  The Toolbox Shortcut menu displays, from which you select the UML or Technology element or connector to insert.	<b>Space bar</b>	<a href="#">Toolbox Shortcut Menu</a> <sup>[799]</sup>
<b>New Boundary</b>	Click on this option and click on the current diagram to add a simple Boundary element to the diagram.		<a href="#">System Boundary</a> <sup>[1347]</sup>
<b>New Note</b>	Click on this option and click on the current diagram to add a new Note element to the diagram.  When you add the Note to the diagram the Notes dialog displays, into which you type the Note text.		<a href="#">Create Notes and Text</a> <sup>[923]</sup>
<b>New Text Element</b>	Click on this option and click on the current diagram to add a simple text element to the diagram.  When you add the Text element to the diagram the Notes dialog displays, into which you type the text.		
<b>Diagram Note Element</b>	Click on this option and click on the current diagram to add a Diagram Properties Note to the diagram.		<a href="#">Insert Diagram Properties Note</a> <sup>[848]</sup>

<b>New Diagram Legend</b>	<p>Click on this option and click on the current diagram to add a diagram Legend element to the diagram.</p> <p>The Legend dialog displays, on which you define the appearance of the Legend element itself and the appearance of the legends it defines.</p>		<a href="#">Create Legends</a>  <sup>849</sup>
<b>New Hyperlink</b>	<p>Click on this option and click on the current diagram to add a Hyperlink element to the current diagram. You can use this element as a hyperlink to, for example, a web page, file or help topic.</p> <p>The Hyperlink Details dialog displays.</p>		<a href="#">Hyperlinks</a>  <sup>2002</sup>
<b>New Note Link</b>	<p>Click on this option, click on the current diagram and 'drag' between a Note element and the element it annotates, to link the Note to the element.</p>		<a href="#">Notelink</a>  <sup>1435</sup>

## Notes

- You can move this toolbar to any dockable position and it retains that position in subsequent sessions

#### 2.4.5.5 Current Element Toolbar

The **Current Element** toolbar provides a number of options for performing common operations on a selected element, such as viewing the element's properties, operations or attributes. You can also use it to specify the visibility of the element's features and compartments, and to lock that element.



**Access**   **View** | **Toolbars** | **Element**

### Current Element Toolbar Options

Icon/Option	Action	Shortcut	See also
<b>Edit Properties</b>	Click on this option to display the element Properties dialog, through which you can view and edit the properties of the selected element.	<b>Alt+Enter</b>	<a href="#">Properties Dialog</a> <sup>[95]</sup>
<b>Set Parent</b>	Click on this option to display the Set Parents and Interfaces dialog, through which you can set the parent or implement interfaces for the selected element.	<b>Ctrl+I</b>	<a href="#">Set Element Parent</a> <sup>[90]</sup>

<b>Operations</b>	Click on this option to display the Operations dialog, through which you can create and edit operations for the selected element.	<b>F10</b>	<a href="#">General Properties of Operations</a> <small>[1015]</small>
<b>Attributes</b>	Click on this option to display the Attribute Properties dialog, through which you can create and edit attributes for the selected element.	<b>F9</b>	<a href="#">General Properties of Attributes</a> <small>[1001]</small>
<b>Show Features</b>	Click on this option to display the Feature and Compartment Visibility dialog, through which you can specify which compartments and information to display for the selected element.	<b>Ctrl+Shift+Y</b>	<a href="#">Feature Visibility</a> <small>[845]</small>
<b>Set Runstate/Initialization</b>	Click on this option to display the Override Attribute Initializers dialog, through which you can set the element run state, attribute initializers or - for Parts - property value.	<b>Ctrl+Shift+R</b>	<a href="#">Run-time State</a> <small>[1380]</small> <a href="#">Add Property Value</a> <small>[1383]</small>
<b>Find Element Usage</b>	Click on this option to find all other diagrams in which the selected element is used: <ul style="list-style-type: none"> <li>• If the element is not used in any other diagram, there is no further action</li> <li>• If the element is included in other diagrams, the Element Usage dialog displays, listing those diagrams, and you can chose which diagram to explore.</li> </ul>	<b>Ctrl+U</b>	<a href="#">Show Element Use</a> <small>[910]</small>
<b>Find in Project Browser</b>	Click on this option to highlight the selected element in the Project Browser.	<b>Alt+G</b>	
<b>Configure References</b>	Click on this option to display the Custom References dialog, on which you can view the properties of other elements and diagrams in the model to which the selected element has custom cross-references.	<b>Ctrl+J</b>	<a href="#">Set Up Cross Reference</a> <small>[916]</small>
<b>Lock Element</b>	Click on this option to apply or release a lock on the selected element. The element is then blocked against or made available for editing, respectively, depending on which locking policy your system is operating under.		<a href="#">Lock Model Elements</a> <small>[336]</small>
<b>Add Tagged Value</b>	Click on this option to display the Tagged Value dialog,	<b>Ctrl+Shift</b>	<a href="#">Assign a</a>

	through which you can add a Tagged Value to the selected element.	+T	<a href="#">Tagged value to an Item</a> <small>[1137]</small>
--	---	----	--

#### Notes

- You can move this toolbar to any dockable position and it retains that position in subsequent sessions
- You can hide or show the toolbar by clicking on the **View | Toolbars | Current Element** menu option

### 2.4.5.6 Current Connector Toolbar

The **Current Connector** toolbar provides a number of options for quickly modifying the properties, style and direction of a connector, showing and hiding a connector and/or its labels, and pinning each end of the connector.



**Access** **View | Toolbars | Connector**

#### Current Connector Toolbar Options

Icon/Option	Action	Shortcut	See also
<b>Connector Properties</b>	Click on this option to display the <connector type> Properties dialog, on which you can view and edit the properties of the selected connector.	<b>Enter</b>	<a href="#">Connector Properties</a> <small>[1126]</small>
<b>Connector Style</b>	Click on the drop-down arrow and select the appropriate connector line style from the list. The options are: <ul style="list-style-type: none"> <li>Direct Route</li> <li>Auto Routing</li> <li>Custom Routing</li> </ul>	<b>Ctrl+Shift+D</b> <b>Ctrl+Shift+A</b> <b>Ctrl+Shift+C</b>	<a href="#">Connector Styles</a> <small>[1114]</small>
<b>Attach Note or Constraint</b>	Click on this option to display the Link Relations dialog, which you use to attach a note or constraint to the selected connector and to any other existing connectors in the current diagram.  Double-click on the Note element to display the Notes dialog on which to type in the Notes text.		<a href="#">Add a Note to a Connector</a> <small>[1111]</small>
<b>Show or Hide Connector Labels</b>	Click on this option to display the Label Visibility dialog, which you use to specify which of the eight possible label		<a href="#">Hide/Show Labels</a> <small>[1122]</small>

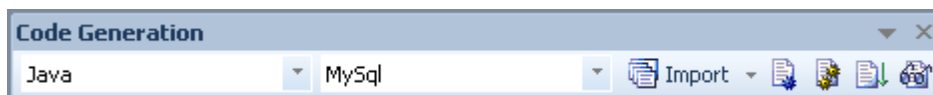
	positions around the selected connector are to be displayed.		
<b>Show and Hide Relationships</b>	Click on this option to display the Set Visible Relations dialog, through which you can show or hide each of the relationships between elements within the current diagram.	<b>Ctrl+Shift+I</b>	<a href="#">Relationship Visibility</a> <sup>[1115]</sup>
<b>Reverse Direction</b>	Click on this option to simply reverse the direction of the selected connector.		<a href="#">Reverse Connector</a> <sup>[1123]</sup>
<b>Pin Connector Ends</b>	Click on the down-arrow to select the end of the selected connector to pin to the source and/or target elements. The pinned end cannot be dragged off the point on the element border to which it has been fixed.		

#### Notes

- You can move this toolbar to any dockable position and it retains that position in subsequent sessions
- You can hide or show the toolbar by clicking on the **View | Toolbars | Current Connector** menu option

### 2.4.5.7 Code Generation Toolbar

The **Code Generation** toolbar provides options to import, generate, synchronize and open source code. Some of these options display convenient menus to quickly set, for example, the default language and default database type for the project.



Access **View | Toolbars | Code Engineering**

#### Code Generation Toolbar Options

Icon/Option	Action	Shortcut	See also
<b>Default Code Language</b>	Click on the drop-down arrow and click on a coding language in the list, to set that language as the default applied to all newly created Class elements.		<a href="#">Language Options</a> <sup>[2262]</sup>
<b>Default Database Type</b>	Click on the drop-down arrow and click on a database type in the list, to set that database type as the default applied to all newly created database elements.		<a href="#">Physical Data Model</a> <sup>[2335]</sup>



<b>Import</b>	<p>Click on the drop-down arrow to locate and select source files to import for reverse engineering code into new Class and Interface elements in the current diagram.</p> <p>Firstly, click on a coding language in the drop-down list to identify the file types to locate; this selection overrides the project's default language for this particular import.</p> <p>When you have selected a language, a browser dialog displays through which you browse for the file location and select from files with the extensions appropriate to the coding language you selected.</p> <p>When you click on a file, the system immediately imports it and generates elements from the file, in the current diagram. The progress of the reverse engineering is shown on a Progress dialog; click on the <b>Close</b> button when the process is complete.</p>		<a href="#">Notes on Source Code Import</a> <sup>[2139]</sup>
<b>Generate Source Code</b>	Click on this option to generate source code for the currently selected single Class element, through the Generate Code dialog.	<b>F11</b>	<a href="#">Generate a Single Class</a> <sup>[2113]</sup>
<b>(Batch) Generate Source Code</b>	<p>Click on this option to batch-generate source code for multiple selected Class elements.</p> <p>A Progress dialog displays and, if no target file is specified in the element, a prompt for the location of a file to generate the code into.</p>	<b>Shift+F11</b>	<a href="#">Generate a Group of Classes</a> <sup>[2114]</sup>
<b>Synchronize Element with Source</b>	<p>Click on this option to synchronize the selected Class element with its associated source code.</p> <p>The synchronization completes as a background task.</p>	<b>F7</b>	
<b>Open Source Code</b>	<p>Click on this option to open the source code of the selected Class, in the default code editor.</p> <p>The source code must have been generated, and the selected element must be a Class. Otherwise no action is taken.</p>	<b>F12</b>	<a href="#">Editing Source Code</a> <sup>[2146]</sup>

**Notes**

- You can move this toolbar to any dockable position and it retains that position in subsequent sessions
- You can hide or show the toolbar by clicking on the **View | Toolbars | Code Generation** menu option

### 2.4.5.8 Debug Toolbar

The **Debug** toolbar provides access to tools for debugging program source code. From this toolbar you can:

- Attach to running processes
- Open a number of debugger windows
- Run, pause or terminate program execution
- Step over, step into or step out of program statements



**Access** [View | Toolbars | Debug](#)

#### Debug Toolbar Options

Icon/Option	Action	Shortcut	See also
<b>Drop-down Command List</b>	Click on the drop-down arrow and select options from the menu for attaching to a running process and accessing a range of debugging windows.		<a href="#">Debug Another Process</a> <a href="#">Breakpoint and Marker Management</a> <a href="#">View the Call Stack</a> <a href="#">View the Local Variables</a> <a href="#">View Variables in Other Scopes</a> <a href="#">Inspect Process Memory</a> <a href="#">Show Loaded Modules</a>
<b>Debug</b>	Click on this option to begin or continue execution of the current program.	<b>F6</b>	
<b>Pause-Resume</b>	Click on this option to pause or resume execution of the current program.		
<b>Step Over</b>	Click on this option to step over the next statement in the program.	<b>Alt+F6</b>	
<b>Step In</b>	Click on this option to step into the next function call in the program.	<b>Shift+F6</b>	

<b>Step Out</b>	Click on this option to step out of the current function call.	<b>Ctrl+F6</b>	
<b>Debug Stop</b>	Click on this option to stop debugging the current program.	<b>Ctrl+Alt+F6</b>	

### Notes

- The toolbar can be docked to any main window edge or floated freely in a convenient location for quick access
- You can move this toolbar to any dockable position and it retains that position in subsequent sessions
- You can hide or show the toolbar by clicking on the **View | Toolbars | Debug** menu option

### Learn more

- [Run the Debugger](#) <sup>[223]</sup>

## 2.4.5.9 Build Toolbar

The **Build** toolbar provides access to tools for building program source code. From this toolbar you can:

- Build the currently active application
- Run the compiled application
- Run your Test and Deployment scripts



**Access** **View | Toolbars | Build**

### Build Toolbar Options

Icon/Option	Action	Shortcut	See also
<b>Active Analyzer Script</b>	Display the name of the currently active analyzer script.		
<b>Execution Analyzer</b>	Click on this icon to display the Execution Analyzer window, which you use to manage the Analyzer scripts for your model.	<b>Shift+F12</b>	<a href="#">Managing Analyzer Scripts</a> <sup>[217]</sup>
<b>Build Current Package</b>	Click on this icon to execute the <b>Build</b> script for the selected package.	<b>Ctrl+Shift+F12</b>	<a href="#">Add Build Commands</a> <sup>[218]</sup>

<b>Cancel Build</b>	Click on this icon to cancel execution of the Build script currently in progress.		
<b>Run Executable</b>	Click on this icon to execute the <b>Run</b> script for the selected package.	<b>Ctrl+Alt+N</b>	<a href="#">Add Run Command</a> <small>[2217]</small>
<b>Run Test Script</b>	Click on this icon to execute the <b>Test</b> script for the selected package.	<b>Ctrl+Alt+T</b>	<a href="#">Add Testing Command</a> <small>[2182]</small>
<b>Run Deployment Script</b>	Click on this icon to execute the <b>Deploy</b> script for the selected package.	<b>Ctrl+Shift+Alt+F12</b>	<a href="#">Add Deploy Command</a> <small>[2218]</small>

### Notes

- The toolbar can be docked to any main window edge or floated freely in a convenient location for quick access
- You can move this toolbar to any dockable position and it retains that position in subsequent sessions
- You can hide or show the toolbar by clicking on the **View | Toolbars | Build** menu option

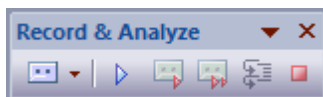
### Learn more

- [Build Application](#) [2221]

## 2.4.5.10 Record & Analyze Toolbar

The **Record & Analyze** toolbar provides access to tools for recording the execution of program source code. From this toolbar you can:



- Begin manual or automation recording of the currently active application
- Step through manual execution
- Generate sequence diagrams from a recording session



**Access** **View | Toolbars | Record**

### Record & Analyze Toolbar Options



Icon/Option	Action	Shortcut	See also
<b>Recorder Command List</b>	Click on the drop-down arrow to show a menu of commands for recording the execution of Analyzer scripts.		<a href="#">Recorder Toolbar</a> <small>[2544]</small>

<b>Recording</b>	Click on this icon to begin or resume recording of the current program.		<a href="#">Start Recording</a> 
<b>Manual Record</b>	Click on this icon to manually step-record the current thread.		
<b>Auto Record Selected Thread</b>	Click on this icon to auto-record an execution trace for the current thread.		
<b>Step Through</b>	Click on this icon to step into a function, record the function call in the History window, and step back out.  The option is enabled for manual recording only.	<b>Alt+;</b>	<a href="#">Step Through Function Calls</a> 
<b>Stop Recording</b>	Click on this icon to end the recording of program execution.		

**Notes**

- The toolbar can be docked to any main window edge or floated freely in a convenient location for quick access
- You can move this toolbar to any dockable position and it retains that position in subsequent sessions
- You can hide or show the toolbar by clicking on the **View | Toolbars | Record** menu option

**Learn more**

- [Recording Sequence Diagrams](#) 
- [Recorder Toolbar](#) 

**2.4.5.11 Workspace Layouts Toolbar**

As you perform work on your model you use various toolbars and windows; you will quite likely use the same toolbars and windows regularly, in an arrangement that suits the way in which you use the system. Rather than having to re-open and arrange these tools at the start of each work session, you can use either a predefined or a customized work environment - or **workspace**. Workspaces:

- Automatically open and organize all the tools appropriate to an area of work such as Requirements Management, Code Engineering or Debugging
- Help a new user by:
  - Opening the tools that are appropriate to a task so that they can immediately get started, and
  - Showing the user what tools they should become familiar with for that area of work
- Help you switch rapidly to work environments for either successive or completely different areas of work
- Re-establish a work environment that you have accidentally or deliberately changed

To select, create and maintain workspaces you can use the **Workspace Layouts toolbar**, which you can leave in the Toolbar ribbon, dock to any main window edge or float freely in a convenient location for quick access.



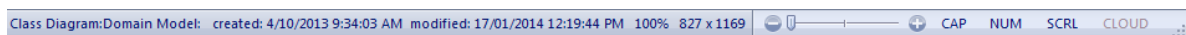
**Access** [View | Toolbars | Workspace Layouts](#)

#### Workspace Toolbar Options

Icon/Option	Action	Short Cut	See also
<b>Save Workspace Layout As</b>	Click on this icon to save the current workspace layout under an existing name or a new name, using the Save Custom Workspace Layout dialog.		<a href="#">Manage Workspace Layout</a> [163]
<b>Manage Workspaces</b>	Click on this icon to open the Workspaces tab on the Workspaces and Commands dialog, to change to a different workspace layout, create or edit a layout, or delete a layout.		
<b>Select Layout</b>	Click on the drop-down arrow and select an existing workspace layout to immediately apply.  User-defined layout names have a preceding asterisk; system-provided layouts are not marked with an asterisk.		

#### 2.4.5.12 Status Bar

The **Status bar** displays at the bottom of the Enterprise Architect workspace. It shows information on the current operation and certain key settings, and provides a 'zoom bar' for adjusting the scale of the diagrams you view.



You can hide the Status bar itself by toggling the **View** menu option (below), and hide sections of the information that the Status bar shows by right-clicking on it and de-selecting either or both of the options:

- Zoom Slider
- Status Indicators

**Access** [View | Toolbars | Status Bar](#)

#### Status Bar Sections

Section	Information	See also
<b>Object Type/ Name</b>	Shows the type and name of the currently-selected diagram, or of the currently selected element in the diagram, or the status of the currently-displayed Model Search.	
<b>Creation data</b>	Shows the date and time at which the current diagram was created and last modified, the percentage zoom of the current display, and the normal page size of the diagram.	
<b>Element/ Feature Name</b>	Shows the name of the currently-selected element, attribute or operation.	
<b>Position and Size</b>	Shows the coordinates of the top left corner of the selected element, and the width and height of the element.	
<b>Zoom Control</b>	Allows a zoom to be applied to all diagrams, regardless of existing zoom levels and without impacting other users. Values range from 100% to 150%.	
<b>Status Indicators</b>	Indicates the status of the <b>Caps Lock</b> , <b>Num Lock</b> and <b>Scrl Lock</b> keyboard keys, and the <b>Connect to Cloud</b> facility (bold indicates 'in use', pale indicates 'off').	<a href="#">Start Page</a> [68]
<b>Display Maximized</b>	A triangle in the bottom right corner indicates that the screen is <b>not</b> maximized; you can drag the screen corner to increase the size of the window.	

### Notes

- The zoom facility has the same function as the **Scale view by** field on the Diagram Appearance page of the Options dialog; changes in the 'zoomed' display scale of a diagram update this field and **are applied to any other diagrams** that you open

### Learn more

- [Diagram Appearance Options](#) [62]

## 2.4.6 Customization

Enterprise Architect has a standard style and appearance, which you can customize to suit your work environment. Several facilities are available for customizing:

- The means of issuing commands to the system - such as menus, toolbars and keyboard keys - and the commands themselves
- The visual style and color of the user interface
- The windows you want to display together on the screen, and their position on the screen (the Workspace Layout)

If you want to quickly select the **sets** of definitions for the user interface to apply, you can select the **Interface Customization** option on the Start Page to customize the interface using the Interface Customization Wizard.

On occasion, you might want to use Enterprise Architect for two distinct types of operation at the same time. You can do this by storing **registry settings** - such as window layouts - to a different path in the registry, by adding the following command line argument when you run Enterprise Architect:

```
/regkey: <regkeyname>
```

You can also configure a wide range of **local** options and specific settings for using Enterprise Architect on your workstation, through the **Options dialog**.

#### Learn more

- [User Interface Customization](#)<sup>[167]</sup>
- [The Customize Dialog](#)<sup>[152]</sup>
- [Workspace Layouts Toolbar](#)<sup>[149]</sup>
- [Manage Workspace Layout](#)<sup>[163]</sup>
- [Customize Command Sets](#)<sup>[165]</sup>
- [Visual Styles](#)<sup>[168]</sup>
- [Local Options](#)<sup>[604]</sup>

### 2.4.6.1 The Customize Dialog

It is possible to customize the Enterprise Architect toolbar and keyboard commands and how they are displayed and made available to you, tailoring the user interface to suit your work methods and environment. You can also define a number of external applications that you can activate from the **Tools** menu. You perform these tasks using the Customize dialog.

#### Access

- Select the **Tools | Customize** option, or
- At the far right of any toolbar, click on the drop-down arrow and on the **Add or Remove buttons | Customize** option

#### Notes

- If a documented toolbar icon, keyboard combination or menu facility does not appear to be available, select the appropriate tab and click on the **Reset** or **Reset All** button to restore the toolbar, menu or key settings to the defaults; however, this also removes any customized icons, options or combinations you might have set, because it is possible that the customization itself has displaced or affected the default setting

#### Learn more

- [Customize Commands](#)<sup>[153]</sup>
- [Customize Toolbars](#)<sup>[156]</sup>
- [Add Custom Tools](#)<sup>[157]</sup>
- [Customize Keyboard Shortcuts](#)<sup>[161]</sup>
- [Customize Menu](#)<sup>[162]</sup>
- [Customize Command Sets](#)<sup>[165]</sup>



- [Customize Options](#)<sup>[163]</sup>

#### 2.4.6.1.1 Customize Commands

The Enterprise Architect toolbars provide a wide range of functions, some of which are useful in several different contexts. You might find it more convenient to have a function command available either from a different toolbar to the one it is currently set in, or from more than one toolbar. You can customize which toolbar a command is available from by generating a button or icon for the command on either another existing toolbar or a new toolbar that you create for this purpose. Conversely, you might remove a command from its current toolbar, to simplify the selection of tools from that toolbar.

**Access** **Tools | Customize > Commands**

#### Manage Toolbar Commands

Task	Action	See also
<b>Add Command To Toolbar</b>	<p><b>Also</b> display the appropriate toolbar, using either:</p> <ul style="list-style-type: none"> <li>• <b>View   Toolbars</b> (for standard toolbars) or</li> <li>• <b>Tools   Customize &gt; Toolbars</b> (for custom toolbars)</li> </ul> <p>The toolbar icons are then highlighted with a dark border as you select them.</p> <p>On the Commands tab:</p> <ol style="list-style-type: none"> <li>1. Click on the appropriate command category in the Categories: panel.</li> <li>2. Click on the command you need from the list in the Commands: panel.</li> <li>3. Drag the command onto the toolbar; it is immediately placed in the toolbar at the cursor position.</li> <li>4. Add any further commands you want.</li> <li>5. Put the icons in the order you prefer, by dragging each icon into position.</li> <li>6. Click on the <b>Close</b> button to close the Customize dialog.</li> <li>7. If necessary, resize the toolbar or drag it back into the top of the screen.</li> </ol> <p>If the command has an associated icon, the icon displays in the toolbar; otherwise, the command name displays.</p>	<p><a href="#">Standard Toolbars</a><sup>[134]</sup></p> <p><a href="#">Customize Toolbars</a><sup>[156]</sup></p>
<b>Delete Command from Toolbar</b>	<p><b>While the Customize dialog is displayed</b>, right-click on the command icon or text in the toolbar, and select the <b>Delete</b> context menu option. The command icon is immediately removed from the toolbar.</p> <p>The context menu does not display if the Customize dialog is closed.</p>	
<b>Change or assign icon for command</b>	<p>All commands listed in the Customize dialog have a text label, and many have associated icons, either system-defined (default) or user-defined. If the command has an icon, you can select to represent the command in the toolbar by:</p> <ul style="list-style-type: none"> <li>• The icon</li> </ul>	

Task	Action	See also
	<ul style="list-style-type: none"> <li>• The text</li> <li>• Both together</li> </ul> <p>You have a range of other options for changing how the command is represented in the toolbar, such as:</p> <ul style="list-style-type: none"> <li>• If the command does not have an associated icon, assign one from an image library</li> <li>• Change the icon currently assigned to the command to a different icon</li> <li>• If the command has a system default icon and you have changed it, restore that default icon to the command</li> <li>• Edit the icons in the library, and create new ones; you can also <b>copy</b> an existing icon and edit the copy</li> </ul> <p>These options are available through a context menu that you display by right-clicking on the icon in the toolbar <i>while the Customize dialog is displayed</i>. The context menu does not display if the Customize dialog is closed.</p> <p>The changes apply only to the selected toolbar. Any instances of the command in other toolbars or menus are not affected.</p>	
<b>Toggle command button between icon, text and both</b>	<ol style="list-style-type: none"> <li>1. Right-click on the command icon or text in the toolbar.</li> <li>2. Select the context menu option you need - <b>Image</b>, <b>Text</b> or <b>Image and Text</b>.</li> </ol> <p>If the option has an icon, the selected change is made. If the command has no icon, the Button Appearance dialog displays (see below). You can also toggle between the image/text options on this dialog.</p>	
<b>Assign icon to command</b>	<ol style="list-style-type: none"> <li>1. Right-click on the command icon or text in the toolbar.</li> <li>2. Select the <b>Button Appearance</b> context menu option; the Button Appearance dialog displays.</li> <li>3. Select either the <b>Image only</b> radio button or the <b>Image and text</b> radio button.</li> <li>4. For a command with a default icon, the <b>Use Default Image</b> radio button is followed by the default icon; if you have assigned a different image, you can restore the default by selecting this radio button.</li> </ol> <p>Otherwise, select the <b>Select User-defined Image</b> radio button, and click on an appropriate image from the selection provided. You can add or edit images in this selection (see below).</p> <ol style="list-style-type: none"> <li>5. Click on the <b>OK</b> button to apply the changes to the toolbar.</li> </ol> <p>(You can also restore the default icon by right-clicking on the current icon in the toolbar and selecting the <b>Reset to default</b> context menu</p>	

Task	Action	See also
	option.)	
<b>Copy a toolbar icon</b>	<ol style="list-style-type: none"> <li>1. Right-click on the icon in the toolbar.</li> <li>2. Select the <b>Copy Button Image</b> context menu option; the image is copied to the clipboard.</li> </ol>	
<b>Create and edit icons</b>	<ol style="list-style-type: none"> <li>1. Right-click on the command graphic or text in the toolbar.</li> <li>2. Select the <b>Button Appearance</b> context menu option; the Button Appearance dialog displays.</li> <li>3. Select either the <b>Image only</b> radio button or the <b>Image and text</b> radio button.</li> <li>4. Select the <b>Select User-defined Image</b> radio button.</li> <li>5. To create a new image, click on the <b>New</b> button; to edit an existing user-defined image, click on the image and then on the <b>Edit</b> button.</li> </ol> <p>The Edit Button Image dialog displays.</p> <ol style="list-style-type: none"> <li>6. If you have copied another icon to edit as a new icon, click on the <b>Paste</b> button in the <b>Tools</b> block.</li> <li>7. Select a color and the appropriate painting tool - the 'pencil' fills individual squares, the 'filler' fills the whole <b>Picture</b> block, and the 'line' creates lines of two or more squares thick.</li> <li>8. Create or edit the icon, working right up to the border of the <b>Picture</b> block if you wish.</li> <li>9. Click on the <b>OK</b> button to apply the changes to the toolbar.</li> </ol>	
<b>Change icon text label</b>	<ol style="list-style-type: none"> <li>1. Right-click on the command graphic or text in the toolbar.</li> <li>2. Select the <b>Button Appearance</b> context menu option; the Button Appearance dialog displays.</li> <li>3. Select either the <b>Text only</b> radio button or the <b>Image and text</b> radio button.</li> <li>4. In the <b>Button text</b> field, highlight and delete the existing text and type in the new text.</li> <li>5. Click on the <b>OK</b> button to apply the changes to the toolbar.</li> </ol>	
<b>Group Icons</b>	<p>If you wish, you can indicate that the command icons are arranged in groups. To do this:</p> <ol style="list-style-type: none"> <li>1. <i>While the Customize dialog is displayed</i>, right-click on the first command icon or text in the group in the toolbar, and select the <b>Start Group</b> context menu option.</li> </ol> <p>A vertical line displays to the left of the icon. All icons to the right of the line form a group, up to any subsequent line.</p> <p>To remove the line and cancel the grouping, repeat the step.</p>	

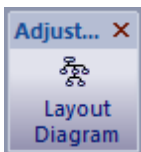
### 2.4.6.1.2 Customize Toolbars


As well as tailoring the commands that are available through the toolbars, you can modify the toolbars themselves; for example, you can:

- Hide or show toolbars by selecting the appropriate checkbox
- Rename toolbars
- Create new toolbars
- Delete toolbars
- Modify toolbar contents by dragging commands onto a visible toolbar from the Commands tab or from another toolbar
- Reset a toolbar (or all toolbars) to the default contents and position, and
- Display text labels under the toolbar icons, to indicate what the icons do

**Access** **Tools | Customize > Toolbars: New**

#### Create a New Toolbar and Populate it with Commands

Step	Action	See also
1	In the <b>Toolbar Name</b> field, type a name for your new toolbar and click on the <b>OK</b> button.  Your new toolbar is created. The toolbar and Customize dialog are independently positioned, so you might have to drag the Customize dialog to one side to expose your new toolbar.	
2	If you want to display the name of each command you put into your toolbar, select the <b>Show text labels</b> checkbox.  Otherwise leave the checkbox unselected.	
3	To add commands to your new toolbar from <b>another toolbar</b> , see step 6.  To add <b>menu</b> commands to your toolbar, click on the Commands tab.  The <b>Categories</b> list on the left of the tab represents the Enterprise Architect menu structure. The <b>Commands</b> list contains the commands in the selected category; the list updates when you click on a different category.	
4	Find and click on the command to add to your toolbar in the <b>Commands</b> list.	
5	Drag the selected command from the list into the new toolbar.  If you selected the <b>Show text labels</b> checkbox, your toolbar should now resemble this:  	

Step	Action	See also
	<p>If you did not select the <b>Show text labels</b> checkbox, your toolbar should resemble this:</p> 	
6	<p>If the command you require is on another toolbar, display that toolbar and drag it to a position near your new toolbar.</p> <p>Press <b>(Ctrl)</b> and drag the required icon for the command from the existing toolbar onto your new toolbar.</p>	
7	<p>Repeat steps 3 to 6 for each command that you want to make available in your new toolbar.</p> <p>When you have finished, click on the <b>Close</b> button on the Customize dialog and drag your toolbar into the toolbar ribbon or to any other convenient position for you to use.</p>	

### Notes

- You can add as many commands to your toolbar as you need; your new toolbar behaves the same way as other toolbars - you can position it next to the other toolbars at the top of the application workspace, dock it to the side of the workspace or close it
- On the Customize dialog, you can customize all the listed toolbars except for the **Format Tool** toolbar (which is actually tied to displayed diagrams)
- You can also modify the display of toolbar options using the **Options** page of the Customize dialog

### Learn more

- [Customize Commands](#) <sup>[153]</sup>
- [Add Custom Tools](#) <sup>[157]</sup>
- [Customize Options](#) <sup>[163]</sup>

### Learning Center topics



- (Alt+F1) | **Enterprise Architect | Getting Started | User Interface | Toolbars**

#### 2.4.6.1.3 Add Custom Tools

You can extend the power of the Enterprise Architect desktop by making external tools accessible from the **Main Menu**. You can create menu options that hyperlink to different applications, compilers, batch scripts, automation scripts, URLs or documentation. The tools can be either commercially available or those you have configured yourself.

**Access** **Tools | Customize > Tools:** 

**Add and configure custom tools**

Field/Option/ Button	Action	See also
<b>Menu contents</b>	In the boxed, blank field in the list, type in the name of the tool as it should appear in the menu.	
<b>Command</b>	Either: <ul style="list-style-type: none"> <li>Type the name of the tool .exe file to use, or</li> <li>Click on the ( ... ) Browse button and browse to the file location</li> </ul> The .exe file must have a valid filename.	
<b>Arguments</b>	Type in any arguments required by the tool	<a href="#">Opening External Tools</a>  <sup>158</sup>  <a href="#">Passing Parameters to External Applications</a>  <sup>160</sup>
<b>Initial directory</b>	(Optional) Type in the location of an initial directory.	
<b>Close</b>	Click on this button to close the Customize dialog.  Your tool is added to the <b>Tools</b> menu.	

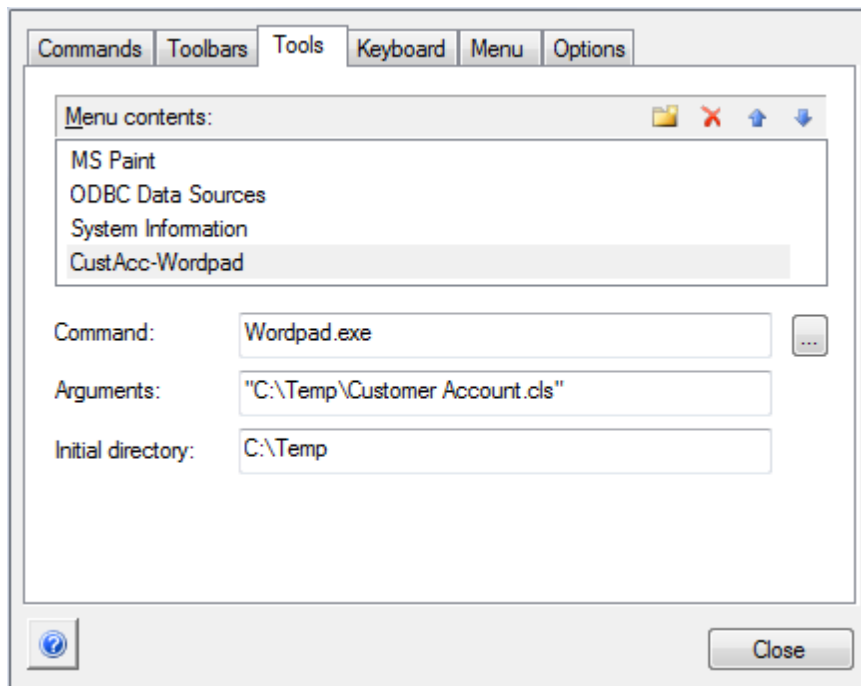
**2.4.6.1.3.1 Open External Tools**

When configuring custom tools in Enterprise Architect, by selecting **Tools | Customize > Tools**, you can:

- Specify the custom tool (application) using the **Command** field
- Define a file to open using the **Arguments** field

**Example 1**

This configuration opens the file `c:\Temp\Customer Account .cls` using Wordpad. If you save from within Wordpad the initial directory is `c:\Temp`.

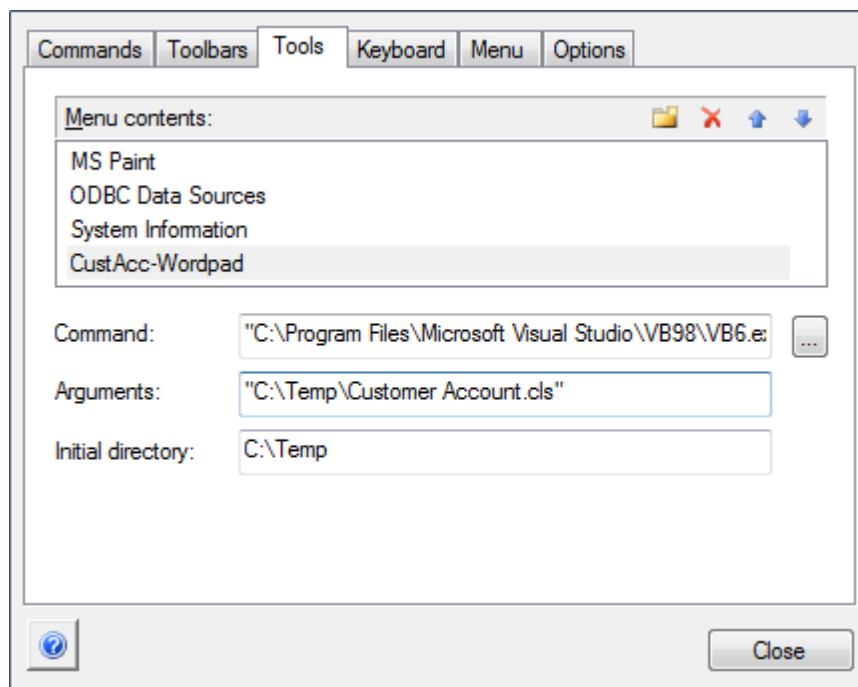


If there are any spaces in the paths in the **Command**, **Arguments** or **Initial directory** fields, you must enclose the whole path in double quotes. For example:

"c:\Temp\Customer Account.cls" must have quotes, but c:\Temp\Customer Account.cls does not have to have quotes.

### Example 2

This example opens the file c:\Temp\Customer Account.cls using **Visual Basic** (VB). As VB is not installed with the operating system, the **whole file path** for VB must be included in the **Command** field; you can locate and select this path using the ( ... ) (Browse) button. Again, if you save from within VB the initial directory is c:\Temp.



#### Learn more

- [Add Custom Tools](#) <sup>[157]</sup>
- [Pass Parameters to Applications](#) <sup>[160]</sup>

#### 2.4.6.1.3.2 Pass Parameters to Applications

When configuring custom tools in Enterprise Architect, by selecting **Tools | Customize > Tools**, you can define parameters to pass to an external application using the **Arguments** field.

#### Available parameters

Parameter	Passes	Notes
<b>\$f</b>	A project name.	For example, <b>C:\projects\EAExample.eap</b> .
<b>\$F</b>	The calling application.	That is, <b>Enterprise Architect</b> .
<b>\$p</b>	The current package ID.	For example, <b>144</b> .
<b>\$P</b>	The package GUID.	A GUID for accessing this package.
<b>\$d</b>	The diagram ID.	An ID for accessing the associated diagram.
<b>\$D</b>	The diagram GUID.	A GUID for accessing the associated diagram.



Parameter	Passes	Notes
<b>\$e</b>	A comma separated list of element IDs.	All elements selected in the current diagram.
<b>\$E</b>	A comma separated list of element GUIDs.	All elements selected in the current diagram.

#### Learn more

- [Custom Tools](#)<sup>[157]</sup>
- [Open External Tools](#)<sup>[158]</sup>

#### 2.4.6.1.4 Customize Keyboard Shortcuts

Various **Main Menu** options can also be invoked using a single keyboard key, or a combination of keys, referred to as **shortcuts**. You can assign your own shortcut keys to either the standard options or your customized options, to suit your work practices or other applications.

**Access** **Tools | Customize > Keyboard**

#### Create or modify a keyboard shortcut

Field/Button/Option	Action	See also
<b>Category</b>	Click on the drop-down arrow and select the name of the <b>Main Menu</b> containing the command to assign a shortcut to.	
<b>Commands</b>	Scroll through the list of commands and click on the one to assign the shortcut key(s) to.  If the command already has a shortcut key it is shown in the <b>Current Keys</b> field.	
<b>Press New Shortcut Key</b>	Click on this field and <b>press</b> the shortcut key(s) to assign to the command.  Press the <b>actual keys</b> to use; for example, to assign <b>F5</b> press the <b>F5 key</b> , do not press <b>F</b> and then <b>5</b> .	
<b>Assign</b>	Click on this button to assign the key to the command.  The button is disabled if the selected keyboard shortcut is already used for another command; if this occurs the command it is assigned to is shown in the <b>Assigned to</b> field, and you must select a different shortcut key.	

Field/Button/Option	Action	See also
<b>Remove</b>	To remove a shortcut, select it in the <b>Current Keys</b> field and click on this button. The shortcut is immediately removed.	
<b>Reset All</b>	Click on this button to reset all commands to the default set of shortcut keys. All keys that <b>you</b> have assigned are removed.	
<b>Close</b>	Click on this button to close the dialog, saving all keys that you have assigned.	

#### Notes

- You can assign more than one shortcut to a command if you wish, although it is simpler to have one shortcut per command
- Modified shortcut keys are stored in the registry, so they are available only to you and not to other users

#### 2.4.6.1.5 *Customize Menu*

It is possible to customize your **Main Menu** submenus to add or remove shadow around the outline of each menu and to apply animation to the way in which the submenus are opened.

Access **Tools | Customize > Menu**

#### Animate Menu Display

Field/Option/ Button	Description	See also
<b>Menu animations</b>	Click on the drop-down arrow and select one of the options: <ul style="list-style-type: none"> <li>• <b>None</b> - the menus are instantaneously displayed on selection</li> <li>• <b>Unfold</b> - the menus unfold from the top left corner diagonally downwards</li> <li>• <b>Slide</b> - the menus slide out from the top downwards</li> <li>• <b>Fade</b> - the menus fade in</li> <li>• <b>[Default]</b> - similar to Fade</li> </ul>	
<b>Menu Shadows</b>	Select the checkbox to display a shadow along the right and lower edges of the menu.  Clear the checkbox to display menus with no shadow on the borders.	
<b>Close</b>	Click on this button to close the Customize dialog.  Your animation options take effect on the menus.	

**Notes**


- When you perform a major upgrade of the system (such as from release 10 to 11) the menus are reset and deleted options are replaced.

**2.4.6.1.6 Customize Options**

It is possible to customize the entire set of Toolbars to display with large icons, and/or show a Screen Tip as you mouse over each toolbar icon. The options take instant effect, so you can see what impact they have before closing the dialog.

**Access** **Tools | Customize > Options**

**Customize Toolbar Display**

Field/Option/ Button	Description	See also
<b>Show Screen Tips on Toolbars</b>	Select this checkbox to display a Screen Tip as you mouse over each toolbar icon, if a Screen Tip is defined for the icon.  The screen tip is predefined in Enterprise Architect and cannot be edited.	
<b>Show shortcut keys in Screen Tips</b>	If you have selected to show Screen Tips, select this checkbox to include any shortcut key combinations in the Screen Tip text.  The shortcut keys are editable, and you can add, change or remove the keys assigned to any of the commands represented by an icon. The current shortcut key combination is added to the Screen Tip text.	<a href="#">Customize Keyboard Shortcuts</a> 
<b>Large Icons</b>	Select this checkbox to enlarge all the Toolbox icons. This takes immediate effect, so that you can see how the display changes.	
<b>Close</b>	Click on this button to save and commit the settings.	

**2.4.6.2 Manage Workspace Layout**


As you perform work on your model you use various toolbars and windows; you will quite likely use the same toolbars and windows regularly, in an arrangement that suits the way in which you use the system. Rather than having to re-open and arrange these tools at the start of each work session, you can use either a predefined or a customized work environment - or **workspace**. Workspaces:

- Automatically open and organize all the tools appropriate to an area of work such as Requirements Management, Code Engineering and Debugging
- Help a new user by:
  - Opening the tools that are appropriate to a task so that they can immediately get started, and
  - Showing the user what tools they should become familiar with for that area of work
- Help you switch rapidly to work environments for either successive or completely different areas of

work

- Re-establish a work environment that you have accidentally or deliberately changed

To select, create and maintain workspaces you use the Workspaces tab on the Workspace and Commands dialog, which lists the currently-available system and user-defined layouts.

**Access** **View | Workspaces and Commands > Workspaces** or  
**View | Toolbars | Workspace Layouts:**  **(Manage Workspaces) > Workspaces**

### Manage Layouts

Task	Description	See also
<b>Change workspace layout</b>	<p>The <b>Current Workspace Layout</b> field at the bottom of the Workspaces and Commands dialog identifies the layout last selected. You might have moved or closed windows after applying that layout. The <b>(Copy of existing Workspace Layout)</b> at the top of the Available Workspaces list is a capture of the workspace layout immediately before you opened the dialog.</p> <p>You can change the layout to:</p> <ul style="list-style-type: none"> <li>• The original layout (as identified by the <b>Current Workspace Layout</b> field), discarding any changes you might have made</li> <li>• The <b>Copy of existing Workspace Layout</b>, preparatory to saving the changes in a new named layout, or</li> <li>• One of the other named layouts</li> </ul> <p>To change the layout in use, either:</p> <ul style="list-style-type: none"> <li>• Double-click on the required layout name</li> <li>• Click on the layout name and click on the <b>Apply</b> button or <b>OK</b> button, or</li> <li>• Right-click on the layout name and select the <b>Apply</b> context menu option</li> </ul> <p>When you change the workspace layout in use, the layout name in the toolbar layout field also changes.</p>	
<b>Copy workspace layout</b>	<p>Either:</p> <ul style="list-style-type: none"> <li>• Change the layout in use to the required layout and click on the <b>Save As New</b> button, or</li> <li>• Right-click on the layout name and select the <b>Save As</b> context menu option</li> </ul> <p>The Save Custom Workspace Layout dialog displays.</p> <p>In the <b>Custom Workspace Layout Name</b> field, type a name for the layout. By selecting an existing name you can change an existing layout to something different. Click on the <b>Save</b> button.</p> <p>If you already have tailored windows or views open that you want to include in your selected layout, select the <b>Include active custom views</b> checkbox.</p>	

Task	Description	See also
<b>Delete workspace layout</b>	Right-click on the layout name and select the <b>Delete</b> context menu option. A prompt displays for you to confirm or cancel the deletion.	

### Notes

- If you apply a working set that invokes floating diagrams or views, it will override the current workspace layout with the layout that was in use when the working set was defined
- If you develop a workspace layout that would be useful for your colleagues, you can share it with them by incorporating it in an MDG Technology that they can download

### Learn more

- [Working Sets](#) <sup>[561]</sup>
- [Create MDG Technologies](#) <sup>[1545]</sup> (MDG Technology Wizard - Contents screen)

### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Getting Started | User Interface | Saving Workspace Layouts**
- (Alt+F1) | **Enterprise Architect | Getting Started | User Interface | Change Workspace Layout**

## 2.4.6.3 Customize Command Sets

The Enterprise Architect User Interface has an extensive range of main menu and context menu options to support the work of many different user roles. Some of these menu options might not be necessary for the work you do; therefore, it is possible to show, hide and customize the menus in a **Command Set**, to tailor them to your specific work practices and responsibilities.

**Access** **View | Workspaces and Commands > Commands** or  
**View | Toolbars | Workspace Layouts:**  **(Manage Workspaces) > Commands**

### Manage Command Sets

Field/Option/ Button	Action	See also
<b>Command Sets</b>	<p>List the available Command Sets (or roles).</p> <p>The following predefined Command Sets are shown:</p> <ul style="list-style-type: none"> <li>• <b>Complete</b> - every menu option is visible</li> <li>• <b>Essential UML</b> - options for modeling with UML; more advanced options, such as for security and version control, are <b>not</b> visible</li> <li>• <b>Business Modeling</b> - menu options appropriate to modeling business structures and processes</li> <li>• <b>Software Engineering</b> - menu options for UML modeling, code generation and reverse engineering</li> <li>• <b>Software Development</b> - menu options for software engineering,</li> </ul>	

Field/Option/ Button	Action	See also
	<p>plus commands for building and debugging code</p> <ul style="list-style-type: none"> <li>• <b>System Engineering</b> - menu options for real-time development, embedded development and behavioral model execution</li> <li>• <b>Testing</b> - menu options for writing and running tests, including integration with external unit testing frameworks</li> <li>• <b>Project Management</b> - menu options for managing a project, including resource and task tracking</li> </ul> <p>These predefined Command Sets can be used as a template for creating new Command Sets.</p> <p>To create or work on a command set, right-click on it in the <b>Command Sets</b> field and select from these context menu options:</p> <ul style="list-style-type: none"> <li>• <b>Save</b> - Save the modifications to the currently selected user-defined Command Set</li> <li>• <b>Save As</b> - Duplicate the selected Command Set</li> <li>• <b>Delete</b> - Delete the selected user-defined Command Set</li> <li>• <b>Export</b> - Export the selected user-defined Command Set to file</li> <li>• <b>Import</b> - Import a Command Set from an external file</li> </ul> <p>The <b>predefined</b> Command Sets are <b>protected</b> from change, deletion and export.</p>	
<b>Clear check boxes to hide menu commands</b>	<p>A full list of Enterprise Architect menus is shown. You can expand these menus to show the menu options, and the options of any submenus.</p> <p>For each command set, the checkboxes against those menus, submenus and menu options that are to be <b>hidden</b> in the User Interface are <b>not selected</b>.</p> <ul style="list-style-type: none"> <li>• If you want to hide further options in your customized command sets, deselect the corresponding check box</li> <li>• To make menus and options available again, select the checkbox</li> </ul> <p>Click on the <b>Save As</b> button to create a new user-defined Command Set based on your changes; provide a name for the Command Set.</p> <p>Click on the <b>Save</b> button to save changes to an existing user-defined Command Set.</p>	
<b>Apply</b>	Click on this button to <b>apply</b> the currently-selected command set to the menus of the user interface, <b>without</b> closing the dialog.	
<b>OK</b>	Click on this button to <b>apply</b> the currently-selected command set to the menus of the user interface, <b>and</b> close the dialog.	

Field/Option/ Button	Action	See also
<b>Cancel</b>	Click on this button to <b>discard</b> changes to the command sets and close the dialog.	

#### 2.4.6.4 User Interface Customization

If you want to set up your work environment to make specific commands, windows and technologies more easily available, you can customize the various ways in which those tools and commands are selected so that your preferred options are presented together. You might set up a number of groupings, in which case you can quickly select which groups of options to apply, using the **Interface Customization Wizard**.

**Access**   **Start Page:** Interface Customization

##### Select custom options and tools

Step	Action	See also
<b>1</b>	Read the introduction to the Interface Customization Wizard, then click on the <b>Next</b> button.  The Select a Command Set screen displays.	
<b>2</b>	In the list, click on the Command Set that defines the group of menus and options you want to have available (or accept the default of <b>Complete</b> - all menus and options).  Click on the <b>Next</b> button.  The Select a Workspace Layout window displays.	<a href="#">Customize Command Sets</a> [165]
<b>3</b>	Click on the Workspace Layout you want to apply. This displays the windows and toolbars appropriate to the tasks you intend to perform.  Click on the <b>Next</b> button.  The Select Active Technologies window displays.	<a href="#">Manage Workspace Layout</a> [163]
<b>4</b>	Select the checkbox against each technology you want to have available for use. You can also: <ul style="list-style-type: none"> <li>• Enable all technologies, by clicking on the <b>Select All</b> button</li> <li>• Clear all current technology selections, by clicking on the <b>Select None</b> button</li> </ul> To apply your selections, click on the <b>Finish</b> button.	<a href="#">Using MDG Technologies</a> [147]

##### Notes

- On any of the windows, if you decide not to process your selections click on the **Cancel** button

#### Learn more

- [Start Page](#)<sup>[68]</sup>

### 2.4.6.5 Visual Styles

It is possible to change the overall style and color of the Enterprise Architect **user interface** to match the standard **styles** (or **themes**) of a number of Microsoft Office and Visual Studio product releases.

**Access** [View | Visual Style | Select Visual Style](#)

#### Reset the style of the user interface

Step	Action	See also
1	On the Application Look dialog, select the radio button for the style you want to set. Note that if you select the: <ul style="list-style-type: none"> <li><b>Microsoft Office 2007</b>, <b>Microsoft Office 2010</b> or <b>Microsoft Office 2013</b> radio buttons, you can also select from a drop-down list of base-color options</li> <li><b>Microsoft Visual Studio 2012</b> radio button, you can also select either a dark or a light base color</li> <li><b>Microsoft Office 2013</b> and <b>Microsoft Visual Studio 2012</b> radio buttons, the <b>Accent Color</b> field, is enabled, and you can select from a drop-down list of colors for the active window borders and highlighting</li> </ul>	
2	To just try out a style, select it and click on the <b>Apply</b> button. The current user interface setup is redisplayed in the selected style.	
3	When you have selected the style you want to use, click on the <b>OK</b> button.	

#### Notes

- You can also apply a specific color and format **theme** to all **diagrams** that you display

#### Learn more

- [Diagram Theme Options](#)<sup>[61]</sup>

### 2.4.7 Other Windows

Most of the **standard** windows you use in Enterprise Architect have a specific, task-oriented purpose. However, two windows have **broader** functions.

#### Broad-function windows



Window	Detail	See also
<b>System Output Window</b>	This window displays the data that Enterprise Architect generates during each of a range of processes. The data from each type of operation generally has its own window tab.	<a href="#">The System Output Window</a> <sup>[169]</sup>
<b>Web Browser</b>	This view, or window, provides access to internet facilities within your Enterprise Architect work area.	<a href="#">The Web Browser</a> <sup>[170]</sup>

#### Learn more

- [Standard Windows](#)<sup>[125]</sup>

### 2.4.7.1 The System Output Window

The **System Output** window is used to display information generated by internal Enterprise Architect processes, or by Add-Ins and other third-party tools. The output from each type of operation is generally displayed on its own tab within the window. The System Output window can provide useful information during long-running processes and batch operations; through this window you can, for example:

- **Validate** information - many validation processes within Enterprise Architect write out their results to this window
- Review **progress** information during the launch of external processes
- Review command line output from **Build** and **Test** scripts
- Check **parse errors** generated during import of various types of files
- Check **Audit** information
- Re-dock the **Model Search** results into an Output window tab for more convenient reference during various operations

#### Access

- Select **View | System Output**
- Press **Ctrl+Shift+8**
- **Edit | Find in Project: Options | Dock in Output Bar**

#### Options available on output data

Right-click on an item and select the appropriate context menu option to:

- Copy the selected item to the clipboard
- Copy all items to the clipboard
- Save the output to an external file
- Clear the output from the window
- Rollback selected **Get All Latest** updates to a version controlled package

You can also:

- **Drag** the selected items or text out of the System Output window and **onto a diagram** to quickly create a named element
- Double-click on model validation errors or parsing errors to display the **source** of the error
- **Switch** between tabs to view different categories of information from different tools

### Notes

- (Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, with **Auditing turned on**) The **Audit History** tab of the System Output window shows a history of changes to any element or connector selected from the:
  - Audit View
  - Package Browser
  - Diagram List
  - Project Browser, or
  - Current diagram

### Learn more

- [Model Search](#)<sup>[700]</sup>
- [Get All Latest](#)<sup>[427]</sup>
- [Add-Ins](#)<sup>[3010]</sup>
- [Audit History](#)<sup>[455]</sup>
- [Auditing](#)<sup>[446]</sup>




## 2.4.7.2 The Web Browser

The Web Browser provides access within Enterprise Architect to internet facilities such as email, websites and search engines; you define the default home website, search engine and email exchange address on the General page of the Options dialog. The Web Browser displays as a tabbed view in the application work area, where the Start Page and Diagram View display.

**Access**    **View | Web Browser ( Ctrl+Alt+W )**

### Options

To access the:

- Email exchange server, click on the  icon in the toolbar; the appropriate email login window displays
- Web search engine (such as *Google*), click on the  icon in the toolbar; the search engine screen displays
- Home web site, after displaying other web pages, click on the  icon in the toolbar

To go directly to another website or email server (your internet security permitting), in the **Address** field type or select the website http address and click on the **Go** button.

Learn more

- [General Options](#)<sup>[60]</sup>

## 2.4.8 Keyboard Shortcuts

You can display the Enterprise Architect dialogs, windows and views, or initiate processes, using menu options and Toolbar icons. In many cases, you can also access these facilities by pressing individual **keyboard keys** or **combinations of keys**, as **shortcuts**. This table lists the default keyboard shortcut for each of the functions. You can also display the key combinations on the **Help Keyboard** dialog.

Access    **Start Page: Keyboard Accelerators** or  
**Help | Keyboard Accelerator Map**

Notes

- There are additional shortcuts using the keyboard and **mouse** in combination
- If necessary, you can change the keyboard shortcuts using the Keyboard tab of the **Customize** dialog

Learn more

- [Keyboard-Mouse Shortcuts](#)<sup>[180]</sup>
- [Customize Keyboard Shortcuts](#)<sup>[16]</sup>

Operations and keyboard shortcuts

Operation	Shortcut	Menu/Category	See also
Create a new Enterprise Architect project	<b>Ctrl+N</b>	<b>File</b>	<a href="#">Create a Project</a> <sup>[47]</sup> <a href="#">File Based Repositories</a> <sup>[210]</sup>
Open an Enterprise Architect project	<b>Ctrl+O</b>		<a href="#">Open a Project</a> <sup>[202]</sup>
Reload the current project	<b>Ctrl+Shift+F11</b>		<a href="#">Refresh View of Shared Project</a> <sup>[308]</sup>
Print the active diagram	<b>Ctrl+P</b>		<a href="#">Scale Image To Page Size</a> <sup>[87]</sup> <a href="#">Configure Diagram Display</a> <sup>[825]</sup>
Undo Change	<b>Ctrl+Z</b>		<a href="#">Undo Last Action</a> <sup>[873]</sup>
Redo Change	<b>Ctrl+Y</b>		<a href="#">Redo Last Action</a> <sup>[873]</sup>

Operation	Shortcut	Menu/Category	See also
		<b>Edit</b>	
Add a single element to the clipboard list	<b>Ctrl+Space</b>		<a href="#">Edit Menu</a> <sup>[80]</sup>
Paste element as metafile from clipboard	<b>Ctrl+Shift+Insert</b>		
Paste element as new	<b>Ctrl+Shift+V</b>		<a href="#">Paste Elements</a> <sup>[82]</sup> <a href="#">Copy Elements Between Diagrams</a> <sup>[912]</sup>
Paste element(s) from the clipboard	<b>Shift+Insert</b>		<a href="#">Move Elements Between Diagrams</a> <sup>[59]</sup> <a href="#">Paste Elements</a> <sup>[82]</sup>
Bookmark current element with red marker	<b>Shift+Space</b>		<a href="#">Manage Bookmarks</a> <sup>[59]</sup>
Delete selected element(s) in diagram	<b>Delete or Ctrl+D</b>		<a href="#">Package Browser</a> <sup>[673]</sup> <a href="#">Operations on Elements</a> <sup>[939]</sup>
Delete selected element(s) from model (through diagram OR Project Browser)	<b>Ctrl+Delete</b>		<a href="#">Delete Elements From Your Model</a> <sup>[65]</sup> <a href="#">Delete Elements from Diagram and Model</a> <sup>[92]</sup>
Search for items in the project (Model Search)	<b>Ctrl+F</b> <b>Ctrl+Alt+A</b>		<a href="#">Model Search</a> <sup>[700]</sup>
Display the Project Browser	<b>Alt+0</b>	<b>View</b>	<a href="#">The Project Browser</a> <sup>[646]</sup>
View Notes window	<b>Ctrl+Shift+1</b>		<a href="#">Notes</a> <sup>[1142]</sup>
View Traceability window	<b>Ctrl+Shift+4</b>		<a href="#">The Traceability Window</a> <sup>[725]</sup>
Display Model Views	<b>Ctrl+Shift+5</b>		<a href="#">Model Views</a> <sup>[686]</sup>
View Tagged Values window	<b>Ctrl+Shift+6</b>		<a href="#">Tagged Values</a> <sup>[1134]</sup>
View System Output window	<b>Ctrl+Shift+8</b>		<a href="#">The System Output Window</a> <sup>[169]</sup>

Operation	Shortcut	Menu/Category	See also
View Learning Center	<b>Alt+F1</b>		<a href="#">Learning Center</a> <sup>[74]</sup>
View Package Browser	<b>Ctrl+A</b>		<a href="#">Package Browser</a> <sup>[673]</sup>
Open Team Review	<b>Ctrl+Alt+U</b>		<a href="#">Team Review Tools</a> <sup>[343]</sup>
Display Web Browser	<b>Ctrl+Alt+W</b>		<a href="#">The Web Browser</a> <sup>[170]</sup>
View Project Glossary	<b>Alt+2</b>	<b>Project</b>	<a href="#">Project Glossary</a> <sup>[533]</sup>
View Resources window	<b>Alt+6</b>		<a href="#">Resources</a> <sup>[1173]</sup>
Add a new Model to the project, using the Model Wizard	<b>Ctrl+Shift+M</b>		<a href="#">Model Wizard</a> <sup>[753]</sup>
Add new package to project	<b>Ctrl+W</b>		<a href="#">Add a Package</a> <sup>[772]</sup>
Add new diagram to package	<b>Ctrl+Insert</b>		<a href="#">Add new diagrams</a> <sup>[822]</sup>
Manage Baselines	<b>Ctrl+Alt+B</b>		<a href="#">Manage Baselines</a> <sup>[461]</sup>
Configure Package Control	<b>Ctrl+Alt+P</b>	<b>Project   Version Control</b>	<a href="#">Configure Controlled Package</a> <sup>[424]</sup>
Manage locks you have applied	<b>Ctrl+Shift+L</b>	<b>Project   Security</b>	<a href="#">Manage Your Own Locks</a> <sup>[342]</sup>
Validate selected modeling objects	<b>Ctrl+Alt+V</b>	<b>Project   Model Validation</b>	<a href="#">Model Validation</a> <sup>[2594]</sup>
Create Document reports	<b>F8</b>		<a href="#">Document Reports</a> <sup>[2640]</sup>
Generate Web Report	<b>Shift+F8</b>	<b>Project   Documentation</b>	<a href="#">Web Reports</a> <sup>[2744]</sup>

Operation	Shortcut	Menu/Category	See also
Generate Diagrams-only Report	<b>Ctrl+Shift+F8</b>		<a href="#">Diagrams Only Report</a> <sup>[274]</sup>
Import a Package from XML	<b>Ctrl+Alt+I</b>	<b>Project   Model Import/ Export</b>	<a href="#">Import from XML</a> <sup>[478]</sup>
Export a Package to XML	<b>Ctrl+Alt+E</b>		<a href="#">Export to XML</a> <sup>[475]</sup>
Import and export to data CSV files	<b>Ctrl+Alt+C</b>		<a href="#">CSV Import and Export</a> <sup>[49]</sup>
Display Diagram Toolbox	<b>Alt+5</b>		<a href="#">Diagram Toolbox</a> <sup>[792]</sup>
View Pan & Zoom Window	<b>Ctrl+Shift+N</b>	<b>Diagram</b>	<a href="#">The Pan &amp; Zoom Window</a> <sup>[698]</sup>
Edit diagram properties	<b>F5</b>		<a href="#">Set Diagram Properties</a> <sup>[823]</sup>
Save changes to a diagram	<b>Ctrl+S</b>		<a href="#">Save Changes</a> <sup>[51]</sup>
Save image to file	<b>Ctrl+T</b>		<a href="#">Copy Image to Disk</a> <sup>[841]</sup>
Save image to clipboard	<b>Ctrl+B</b>		<a href="#">Copy Image to Clipboard</a> <sup>[842]</sup>
Locate in Project Browser	<b>Shift+Alt+G</b>		<a href="#">Diagram Menu</a> <sup>[93]</sup>
Create a new element of the same type as the last	<b>Shift+F3</b> or <b>Ctrl + click</b>		<a href="#">Create Elements</a> <sup>[902]</sup>
Create a new connector of the same type as the last	<b>F3</b>		<a href="#">Connect Elements</a> <sup>[1109]</sup>
Set visibility of connectors	<b>Ctrl+Shift+I</b>	<b>Diagram   Advanced</b>	<a href="#">Relationship Visibility</a> <sup>[1119]</sup>
View Properties window	<b>Alt+1</b>		<a href="#">Properties Window</a> <sup>[992]</sup>
View Testing window	<b>Alt+3</b>		<a href="#">Working on Test Records</a> <sup>[2605]</sup>
View Maintenance window	<b>Alt+4</b>		<a href="#">Working on Maintenance</a>

Operation	Shortcut	Menu/Category	See also
		<b>Element</b>	<a href="#">Items</a> <sup>[2623]</sup>
View Source Code window	<b>Alt+7</b>		<a href="#">Editing Source Code</a> <sup>[2146]</sup>
View Element Relationships window	<b>Ctrl+Shift+2</b>		<a href="#">The Relationships Window</a> <sup>[742]</sup>
View Rules and Scenarios (Requirements and Constraints) window	<b>Ctrl+Shift+3</b>		<a href="#">Element Scenarios &amp; Requirements</a> <sup>[992]</sup>
View Project Management window	<b>Ctrl+Shift+7</b>		<a href="#">The Project Management Window</a> <sup>[510]</sup>
View Element Browser	<b>Alt+9</b>		<a href="#">The Element Browser</a> <sup>[989]</sup>
Display element properties	<b>Alt+Enter</b>		<a href="#">Properties Dialog</a> <sup>[956]</sup>
Create a Linked Document	<b>Ctrl+Alt+D</b>		<a href="#">Create Linked Document on an Element</a> <sup>[1047]</sup>
Locate diagrams where element is used	<b>Ctrl+U</b>		<a href="#">Show Element Use</a> <sup>[910]</sup>
Display Attribute Properties dialog	<b>F9</b>		<a href="#">Attributes</a> <sup>[999]</sup>
Display Operation Properties dialog	<b>F10</b>		<a href="#">General Properties of Operations</a> <sup>[1015]</sup>
View source code in default external editor	<b>Ctrl+E</b> <b>F12</b>		<a href="#">Editing Source Code</a> <sup>[2146]</sup>
Locate element in Project Browser	<b>Alt+G</b>		<a href="#">Element Menu</a> <sup>[97]</sup> <a href="#">Finding Elements</a> <sup>[944]</sup>
Specify which element features are visible on a diagram	<b>Ctrl+Shift+Y</b>		<a href="#">Feature Visibility</a> <sup>[845]</sup>
Add a Tagged Value to an element	<b>Ctrl+Shift+T</b>		<a href="#">Quick Start - Add Tagged Value To Elements</a> <sup>[1136]</sup>
Manage embedded elements	<b>Ctrl+Shift+B</b>		<a href="#">Advanced Submenu</a> <sup>[100]</sup>

Operation	Shortcut	Menu/Category	See also
		<b>Element   Advanced</b>	<a href="#">Manage Structural Elements</a> <sup>[935]</sup>
Set element parent or implement interface (s)	<b>Ctrl+I</b>		<a href="#">Set Element Parent</a> <sup>[908]</sup>
Set references to other elements and diagrams	<b>Ctrl+J</b>		<a href="#">Set Up Cross References</a> <sup>[916]</sup>
Override inherited features	<b>Ctrl+Shift+O</b>		<a href="#">Override Parent Operations</a> <sup>[1023]</sup>
Select alternative image	<b>Ctrl+Shift+W</b>	<b>Element   Appearance</b>	<a href="#">Using the Image Manager</a> <sup>[860]</sup>
Auto-size selected elements	<b>Alt+Z</b>		<a href="#">Autosize Elements</a> <sup>[851]</sup>
Configure element default appearance	<b>Ctrl+Shift+E F4</b>		<a href="#">Set an Element's Default Appearance</a> <sup>[927]</sup>
Add an attribute to an element	<b>Ctrl+Shift+F9</b>	<b>Element   Inline Features</b>	<a href="#">Attributes</a> <sup>[999]</sup>
Add an operation to an element	<b>Ctrl+Shift+F10</b>		<a href="#">General Properties of Operations</a> <sup>[1015]</sup>
Add other types of feature to an element	<b>Ctrl+F11</b>		<a href="#">Insert Maintenance Feature</a> <sup>[1042]</sup> <a href="#">Insert Testing Features</a> <sup>[1043]</sup>
Edit selected	<b>F2</b>		<a href="#">In-place Editing Tasks</a> <sup>[1032]</sup>
Insert new feature after current selection	<b>Insert</b>		<a href="#">Insert New Feature</a> <sup>[1040]</sup>
Generate code from an element	<b>F11</b>	<b>Element   Source Code</b>	<a href="#">Generate a Single Class</a> <sup>[2113]</sup>
Synchronize Current Element	<b>F7</b>		<a href="#">Importing Source Code</a> <sup>[2136]</sup> <a href="#">Element Options in the Project Browser</a> <sup>[660]</sup> <a href="#">Operation Menu - Project Browser</a> <sup>[667]</sup> <a href="#">Update Package Contents</a>



Operation	Shortcut	Menu/Category	See also
		<b>Engineering</b>	<a href="#">[2117]</a>
Batch generate selected elements	<b>Shift+F11</b>		<a href="#">Generate a Group of Classes</a> <a href="#">[2114]</a>
Batch synchronize selected elements	<b>Ctrl+R</b>		
Open source directory	<b>Ctrl+Alt+Y</b>		<a href="#">Editing Source Code</a> <a href="#">[2146]</a> <a href="#">Compare Editors</a> <a href="#">[2148]</a>
Space elements evenly horizontally	<b>Alt+-</b>	<b>Element (Context)   Space Evenly</b>	<a href="#">Align Elements</a> <a href="#">[918]</a>
Space elements evenly vertically	<b>Alt+=</b>		<a href="#">Operations on Multiple Elements</a> <a href="#">[951]</a>
Create Workbench Instance	<b>Ctrl+Shift+J</b>	<b>Element (Context menu)   Execution Analyzer</b>	<a href="#">Create &amp; Delete Workbench Instances</a> <a href="#">[2568]</a>
View element Properties dialog	<b>Enter</b>		<a href="#">Properties Window</a> <a href="#">[992]</a>
Move element by increments	<b>Shift+↑ , ↓ , → , ←</b>		<a href="#">Move Elements Within Diagrams</a> <a href="#">[911]</a>
Resize selected element	<b>Ctrl+↑ , ↓ , ← , →</b>		<a href="#">Resize Elements</a> <a href="#">[919]</a>
Align bottom edges of selected elements	<b>Ctrl+Alt+Down</b>		<a href="#">Operations on Multiple Elements</a> <a href="#">[951]</a>
Align top edges of selected elements	<b>Ctrl+Alt+Up</b>		
Align selected elements on left boundaries	<b>Ctrl+Alt+Left</b>		
Align selected elements on right boundaries	<b>Ctrl+Alt+Right</b>		
Open Debugger Window	<b>Alt+8</b>		<a href="#">Debugging</a> <a href="#">[2222]</a>

Operation	Shortcut	Menu/Category	See also
Manage Analyzer Scripts on Execution Analyzer window	<b>Shift+F12</b>	<b>Analyzer</b>	<a href="#">Managing Analyzer Scripts</a> <sup>[2175]</sup>
Create Build Script	<b>Ctrl+Shift+F12</b>		<a href="#">Add Build Commands</a> <sup>[2180]</sup>
Create Test Script	<b>Ctrl+Alt+T</b>		<a href="#">Add Testing Command</a> <sup>[2182]</sup>
Create Run Script	<b>Ctrl+Alt+N</b>		<a href="#">Add Run Command</a> <sup>[2217]</sup>
Create Deploy Script	<b>Ctrl+Shift+Alt+F12</b>		<a href="#">Add Deploy Command</a> <sup>[2218]</sup>
Start a Debug session	<b>F6</b>	<b>Analyzer   Debug</b>	<a href="#">Run the Debugger</a> <sup>[2231]</sup>
Step Into a function call	<b>Shift+F6</b>		
Step Over a function call	<b>Alt+F6</b>		
Step Out of a function call	<b>Ctrl+F6</b>		
Stop the debug session	<b>Ctrl+Alt+F6</b>		
Check project data integrity	<b>Shift+F9</b>	<b>Tools</b>	<a href="#">Check Project Data Integrity</a> <sup>[597]</sup>
Configure system options	<b>Ctrl+F9</b>		<a href="#">Local Options</a> <sup>[604]</sup>
Spell check current package	<b>Ctrl+Shift+F7</b>		<a href="#">Using the Spell Checker</a> <sup>[553]</sup>
Spell check model	<b>Ctrl+F7</b>		
Open a Source File	<b>Ctrl+Alt+O</b>		<a href="#">Compare Editors</a> <sup>[2148]</sup>
Transform selected elements	<b>Ctrl+H</b> or <b>Ctrl+Alt+F</b>	<b>Tools   Model Transformation</b>	<a href="#">Transform Elements</a> <sup>[2017]</sup>
Transform the current package	<b>Ctrl+Shift+H</b>		

Operation	Shortcut	Menu/Category	See also
Generate package source code	<b>Ctrl+Alt+K</b>	<b>Tools   Source Code Engineering</b>	<a href="#">Generate a Package</a> <sup>[2115]</sup>
Synchronize package contents	<b>Ctrl+Alt+M</b>		<a href="#">Update Package Contents</a> <sup>[2117]</sup>
Import source directory	<b>Ctrl+Shift+U</b>		<a href="#">Import a Directory Structure</a> <sup>[2142]</sup>
Edit code generation templates	<b>Ctrl+Shift+P</b>	<b>Settings</b>	<a href="#">Code and Transform Templates</a> <sup>[1632]</sup> <a href="#">The Code Template Editor</a> <sup>[1641]</sup> <a href="#">The Code Template Editor in MDG Development</a> <sup>[1701]</sup>
Edit transformation templates	<b>Ctrl+Alt+H</b>		<a href="#">Edit Transformation Templates</a> <sup>[2048]</sup> <a href="#">Write Transformations</a> <sup>[2051]</sup>
Make current view occupy the full screen space	<b>Shift+Alt+Enter</b>	<b>Window</b>	<a href="#">Window Menu</a> <sup>[121]</sup>
Set focus to current window	<b>Ctrl+Shift+0</b>		
Close the current window	<b>Ctrl+F4</b>		
Autohide the current window	<b>Ctrl+Shift+F4</b>		<a href="#">Autohide Windows</a> <sup>[132]</sup>
Add new element to package	<b>Ctrl+M</b>	<b>Project Browser</b>	<a href="#">Add Elements Directly To Packages</a> <sup>[903]</sup>
Spell check Notes text	<b>F7</b>	<b>Object notes</b>	<a href="#">Notes</a> <sup>[1142]</sup>
Make text bullet list item	<b>Ctrl+.</b> (full stop)		
Make text numbered list item	<b>Ctrl+1</b>		
Make text bold	<b>Ctrl+B</b>		

Operation	Shortcut	Menu/Category	See also
Make text italic	<b>Ctrl+I</b>		
Make text underlined	<b>Ctrl+U</b>		
Insert date and time in Notes text	<b>F5</b>		
Select line of text in Notes text	<b>F8</b>		
Copy text	<b>Ctrl+C</b>	<b>Everywhere</b>	
Paste text	<b>Ctrl+V</b>		
Cut text, or element in diagram	<b>Ctrl+X</b>		

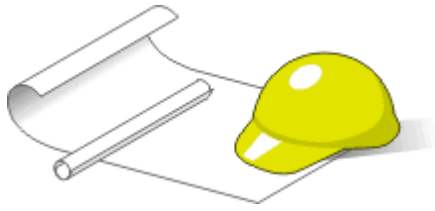
### 2.4.8.1 Keyboard-Mouse Shortcuts

When you work on an aspect of your model **within a diagram**, there are a number of operations you can perform quickly using a combination of **keyboard keys** and the **computer mouse**.

Action	Shortcut
Create an element of the same type as was last created on the diagram.	<b>Ctrl</b> + click
Display the element Properties dialog for the element shown in the scenario Context References tab.	<b>Ctrl</b> + click
Select a number of individual objects for an operation.	<b>Ctrl</b> + click each object
Select a range of objects for an operation.	<b>Shift</b> + click the first and last object in the range
Move the selected elements along the horizontal axis or vertical axis.	<b>Shift</b> + hold left mouse button and drag
Create a bend in and change the direction of the connector line.	<b>Shift</b> + hold left mouse button and drag

Action	Shortcut
Move the selected elements in any direction, including diagonally.	<b>Alt</b> + hold left mouse button and drag
Zoom in to or zoom out of the diagram.	<b>Ctrl</b> + mouse scroll wheel
Pan the diagram horizontally or vertically.	<b>Ctrl +Shift</b> + hold left mouse button and drag

## 2.5 Typical Project Roles



Enterprise Architect supports common **working practices** and **tasks** that are performed by specific **roles** and **professions**. A number of the roles and their responsibilities that the system supports are outlined here.

Role & Responsibilities	See also
<b>Business Analyst</b> , to create high-level models of business processes	<a href="#">Business Analyst</a> <sup>[185]</sup>
<b>Software Architect</b> , to map functional requirements, perform real time modeling of objects, design the Deployment model and detail the deliverable components	<a href="#">Software Architect</a> <sup>[186]</sup>
<b>Software Engineer</b> , to map Use Cases onto Class diagrams, detail the interactions between Classes, define the system deployment and define software packages	<a href="#">Software Engineer</a> <sup>[187]</sup>
<b>Developer</b> , to perform round trip code engineering, including reverse engineering of existing code and generation of code from Class elements	<a href="#">Developer</a> <sup>[188]</sup>
<b>Project Manager</b> , to assign resources to elements, measure risk and effort, estimate project sizes, and manage element status, change control and maintenance	<a href="#">Project Manager</a> <sup>[190]</sup>
<b>Tester</b> , to create test scripts against elements in the modeling environment	<a href="#">Tester</a> <sup>[191]</sup>
<b>Implementation Manager</b> , to: <ul style="list-style-type: none"> <li>Track and assign maintenance-related items to elements within Enterprise Architect</li> <li>Rapidly capture and keep records of maintenance tasks such as issues, changes, defects and tasks</li> <li>Trace the maintenance of the items and processes involved in system deployment</li> </ul>	<a href="#">Implementation Manager</a> <sup>[192]</sup>
<b>Technology Developer</b> , to create customized additions to the functionality already present within Enterprise Architect	<a href="#">Technology Developer</a> <sup>[193]</sup>

Role & Responsibilities	See also
<b>Database Developer</b> , to develop databases, including modeling database structures, importing database structures from an existing database and generating DDL for rapidly creating databases from a model	<a href="#">Database Developer</a> [194]

You can review a summary of the **typical tasks** supported for each role, or click on the appropriate role title to explore how Enterprise Architect can assist you in carrying out that role within a model-driven project.

#### Learn more

- [Summary of Typical Tasks](#) [183]

### 2.5.1 Summary of Typical Tasks

Throughout a design and development project there are many different tasks to be performed, which could be carried out either by one person or - more probably - by members of a team with different responsibilities. In either case, Enterprise Architect supports most - if not all - of the **responsibilities** you might have on your project. The descriptions below cover a number of job roles that the system supports. For those that most resemble your role on a project, follow the job title hyperlink to display a description of how that role might make use of Enterprise Architect, then follow links within those topics to explore some of the features of importance to the role.

#### Summary of Typical Job Roles

Role	Responsibilities	See also
<b>Business Analyst</b>	For modeling: <ul style="list-style-type: none"> <li>• Requirements</li> <li>• High-level business processes</li> <li>• Business activities</li> <li>• Work flows</li> <li>• System behavior</li> </ul>	<a href="#">Business Analyst</a> [185]
<b>Database Developer</b>	<ul style="list-style-type: none"> <li>• Developing databases</li> <li>• Modeling database structures</li> <li>• Creating logical data models</li> <li>• Generating schema</li> <li>• Reverse engineering databases</li> </ul>	<a href="#">Database Developer</a> [194]
<b>Software Architect</b>	<ul style="list-style-type: none"> <li>• Mapping functional requirements of the system</li> <li>• Mapping objects in real time</li> <li>• Mapping the deployment of objects</li> <li>• Defining deliverable components</li> </ul>	<a href="#">Software Architect</a> [186]
<b>Tester</b>	<ul style="list-style-type: none"> <li>• Developing test cases</li> <li>• Importing requirements, constraints and</li> </ul>	<a href="#">Tester</a> [197]

Role	Responsibilities	See also
	<ul style="list-style-type: none"> <li>scenarios</li> <li>Creating Quality Test documentation</li> <li>Tracking element defects and changes</li> </ul>	
<b>Software Engineer</b>	<ul style="list-style-type: none"> <li>Mapping Use Cases into detailed Classes</li> <li>Defining the interaction between Classes</li> <li>Defining system deployment</li> <li>Defining software packages and the software architecture</li> </ul>	<a href="#">Software Engineer</a> <sup>[187]</sup>
<b>Project Manager</b>	<ul style="list-style-type: none"> <li>Providing project estimates</li> <li>Resource Management</li> <li>Risk Management</li> <li>Maintenance Management</li> </ul>	<a href="#">Project Manager</a> <sup>[190]</sup>
<b>Developer</b>	<ul style="list-style-type: none"> <li>Forward, reverse and round-trip engineering</li> <li>Visualizing the system states</li> <li>Visualizing package arrangements</li> <li>Mapping the flow of code</li> </ul>	<a href="#">Developer</a> <sup>[188]</sup>
<b>Implementation Manager</b>	<ul style="list-style-type: none"> <li>Modeling the tasks in rolling-out a project, including network and hardware deployment</li> <li>Assigning and tracking maintenance items on elements (issues, changes, defects and tasks)</li> </ul>	<a href="#">Implementation Manager</a> <sup>[192]</sup>
<b>Technology Developer</b>	For creating or customizing: <ul style="list-style-type: none"> <li>UML Profiles</li> <li>UML Patterns</li> <li>Code Templates</li> <li>Tagged Value Types</li> <li>MDG Technologies</li> <li>Add-Ins</li> </ul>	<a href="#">Technology Developer</a> <sup>[193]</sup>

Most of these roles work with specific types of diagram, so you might want to learn more about diagram types in general and specific types of diagram in particular.

Several types of project team member might want to **generate documentation** on their work and report on how the project is developing and changing. Using Enterprise Architect you can generate project reports in either document or web format.

#### Notes

- The Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect have a user security feature that can be applied or turned off; if security is turned



on, you need to have the appropriate access permissions to use many of the facilities listed above

#### Learn more

- [List of Available Permissions](#) <sup>[329]</sup>
- [UML Diagrams](#) <sup>[1181]</sup>
- [Report Generation](#) <sup>[2638]</sup>


## 2.5.2 Business Analysts

A **Business Analyst** can use Enterprise Architect to create high-level models of business processes, including business requirements, activities, work flow, and the display of system behavior.

Using Enterprise Architect, a Business Analyst can describe the procedures that govern what a particular business does. Such a model is intended to deliver a high-level overview of a proposed system.

#### Business Analyst Tasks

Task	Detail	See also
<b>Model High Level Business Processes</b>	<p>Using <b>Analysis</b> diagrams, you can model the high-level processes of the business.</p> <p>Analysis diagrams are a subset of UML 2.4.1 Activity diagrams and are less formal than other diagram types, but they provide a useful means for expressing essential business characteristics and requirements.</p>	<a href="#">Analysis Diagram</a> <sup>[1801]</sup> <a href="#">Example Analysis Diagram</a> <sup>[1802]</sup>
<b>Model Requirements</b>	<p>Gathering <b>requirements</b> is typically the first step in developing a solution, be it for developing a software application or for detailing a business process; it is an important step in the implementation of a project.</p> <p>Using Enterprise Architect, you can define the Requirement elements, connect Requirements to the model elements for implementation, connect Requirements together into a hierarchy, report on Requirements, and move Requirements out of model element responsibilities.</p>	<a href="#">Requirements Management</a> <sup>[1726]</sup> <a href="#">Specification Manager</a> <sup>[1728]</sup>
<b>Model Business Activities</b>	<p>You can use <b>Activity</b> diagrams to model the behavior of a system and the way in which these behaviors are related to the overall flow of the system.</p> <p>Activity diagrams do not model the exact internal behavior of the system but show instead the general processes and pathways at a high level.</p>	<a href="#">Activity Diagram</a> <sup>[1199]</sup>
<b>Model Work Flow</b>	<p>To visualize the cooperation between elements involved in the work flow, you can use an <b>Interaction Overview</b> diagram, which provides an overview of sub activities that are involved in a system.</p>	<a href="#">Interaction Overview Diagram</a> <sup>[1262]</sup>

Task	Detail	See also
<b>Display System Behavior</b>	In displaying the behavior of a system as a <b>Use Case</b> diagram, Enterprise Architect provides an easily understood tool for mapping the functional requirements and behavior of a system.	<a href="#">Use Case Diagram</a> 







#### Learn more

- [Business Modeling](#) 

## 2.5.3 Software Architects

**Software Architects** can use Enterprise Architect to map functional requirements with Use Cases, perform real time modeling of objects using **Interaction diagrams** (Sequence, Timing, Communication or Interaction Overview), design the **Deployment** model and detail the deliverable components using **Component** diagrams.

#### Software Architect Tasks

Task	Detail	See also
<b>Map Functional Requirements of the System</b>	With Enterprise Architect you can take the high level business processes that have been modeled by the Business Analyst and create detailed <b>Use Cases</b> .  Use Cases describe the proposed functionality of a system and are only used to detail a single unit of discrete work.	<a href="#">Use Cases</a> 
<b>Map Objects in Real Time</b>	You can use <b>Interaction</b> diagrams (Sequence and Communication diagrams) to model the dynamic design of the system.  <b>Sequence</b> diagrams detail the messages that are passed between objects, and the lifetimes of the objects.  <b>Communication</b> diagrams are similar to Sequence diagrams, but instead display the way in which the object interacts with other objects.	<a href="#">Sequence Diagram</a>  <a href="#">Example Sequence Diagram</a>  <a href="#">Communication Diagram</a> 
<b>Map Deployment of Objects</b>	You can use <b>Deployment</b> diagrams to provide a static view of the run-time configuration of processing nodes and the components that run on the nodes.  Deployment diagrams show the connections between hardware, software and any middleware that is used on a system.	<a href="#">Deployment Diagram</a> 
<b>Detail Deliverable</b>	Using <b>Component</b> diagrams, you can model the	<a href="#">Component Diagram</a> 

Task	Detail	See also
<b>Components</b>	<p>physical aspects of a system.</p> <p>Components can be executables, libraries, data files or another physical resource that is part of a system.</p> <p>The component model can be developed from scratch from the Class model or can be brought in from existing projects and from third-party vendors.</p>	

#### Learn more

- [Analysis Diagrams](#) <sup>[1801]</sup>
- [Timing Diagram](#) <sup>[1225]</sup>
- [Interaction Overview Diagram](#) <sup>[1262]</sup>
- [Modeling Fundamentals](#) <sup>[750]</sup>
- [XML Import and Export](#) <sup>[473]</sup>
- [XML Technologies](#) <sup>[2387]</sup>

## 2.5.4 Software Engineers

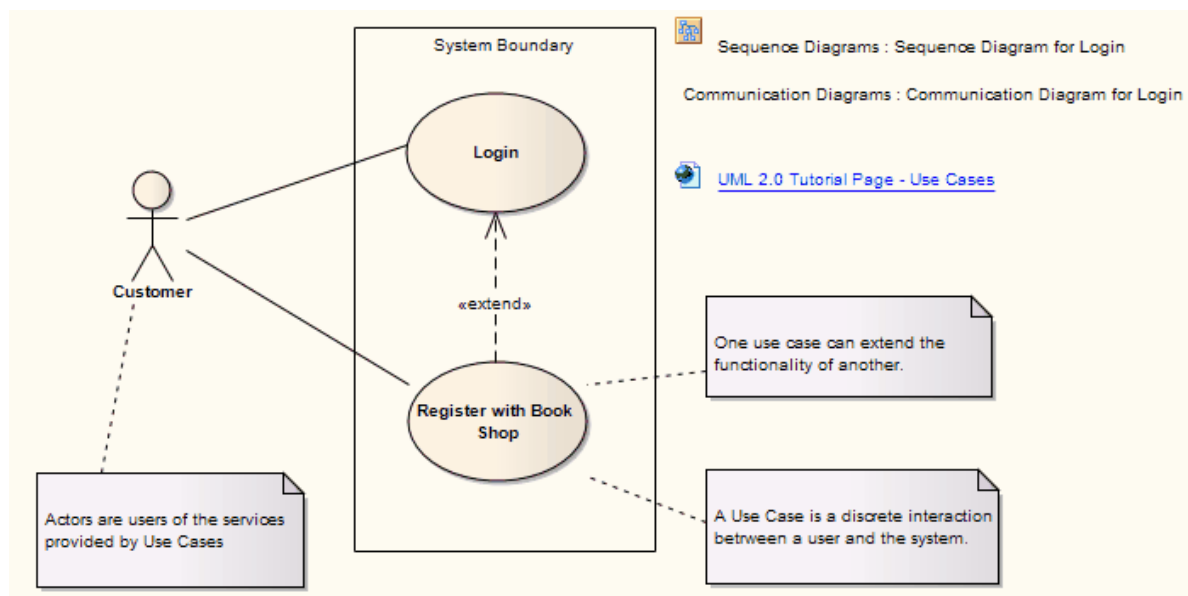
**Software Engineers** using Enterprise Architect can manually map Use Cases onto Class diagrams, detail the interactions between Classes, define the system deployment with Deployment diagrams and define software packages with Package diagrams.

#### Software Engineering Tasks

Task	Detail	See also
<b>Map Use Cases into Detailed Classes</b>	<p>Within Enterprise Architect you can study the <b>Use Cases</b> developed by the Software Architect, and with that information create <b>Classes</b> that fulfill the objectives defined in the Use Cases.</p> <p>A Class is one of the standard UML constructs that is used to detail the pattern from which objects are produced at run time; to record the relationships between Use Cases and Classes, you can create diagrams linking the elements with <b>Realization</b> connectors, and/or map the Realization connectors in the <b>Relationship Matrix</b>.</p>	<a href="#">Use Cases</a> <sup>[1201]</sup> <a href="#">Example Use Case Diagram</a> <sup>[1203]</sup> <a href="#">Class Diagram</a> <sup>[1184]</sup> <a href="#">Realization</a> <sup>[1440]</sup> <a href="#">Relationship Matrix</a> <sup>[727]</sup>
<b>Detail Interaction Between Classes</b>	<p>You can use <b>Interaction</b> diagrams (<b>Sequence</b> and <b>Communication</b> diagrams) to model the dynamic design of the system.</p> <p>Sequence diagrams are used to detail the messages passed between objects, and the lifetimes of the objects.</p> <p>Communication diagrams are similar to Sequence diagrams, but instead display the way in which objects interact with other</p>	<a href="#">Sequence</a> <sup>[1249]</sup> <a href="#">Communication</a> <sup>[1259]</sup>

Task	Detail	See also
	objects.	
<b>Define System Deployment</b>	<p>Deployment diagrams provide a static view of the run-time configuration of processing nodes and the components that run on the nodes.</p> <p>Deployment diagrams can be used to show the connections between hardware, software and any middleware that is used on a system, to explain the connections and relationships of the components.</p>	<a href="#">Deployment</a> <sup>[119]</sup>
<b>Define Software Packages</b>	<p>You can use Package diagrams to detail the software architecture.</p> <p>Package diagrams are used to organize diagrams and elements into manageable groups, declaring the dependencies.</p>	<a href="#">Package</a> <sup>[118]</sup>

### Simple Use Case diagram













### Learn more

- [Modeling Fundamentals](#)<sup>[750]</sup>

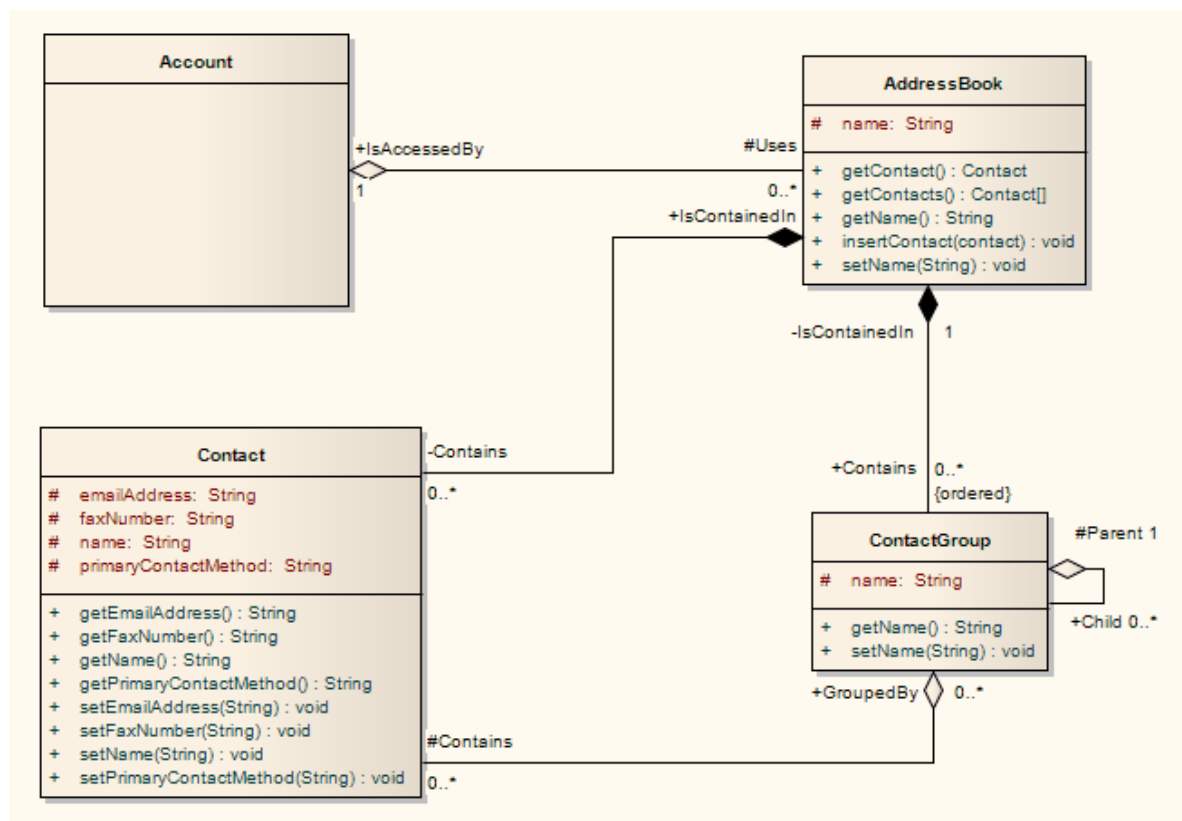
## 2.5.5 Developers

**Developers** can use Enterprise Architect to perform round trip code engineering, which includes reverse engineering of existing code and generation of code from Class elements.

### Developer Tasks

Task	Detail	See also
<b>Round Trip Engineering</b>	Enterprise Architect gives you unparalleled flexibility in <b>'round tripping'</b> software from existing source code to UML 2.4.1 models and back again.  Round trip engineering involves both <b>forward</b> and <b>reverse engineering</b> of code, keeping the model and code synchronized.	<a href="#">Code Engineering</a>  <a href="#">Update Package Contents</a>  <a href="#">Estimation</a> 
<b>Reverse Engineering</b>	In Enterprise Architect, you can reverse engineer code from a number of supported languages and view the existing code as <b>Class</b> diagrams, which illustrate the static design view of the system.  Class diagrams show Classes and interfaces, and the relationships between them; the Classes defined in UML Class diagrams can have direct counterparts in the implementation of a programming language.	<a href="#">Reverse Engineering</a>  <a href="#">Class Diagram</a>  <a href="#">Class Element</a> 
<b>Forward Engineering</b>	As well reverse engineering your code into your model, you can forward engineer elements of your model into code ( <b>code generation</b> ).  This way you can make changes to your model with Enterprise Architect and quickly implement the changes in the source code.	<a href="#">Code Engineering</a> 
<b>Determine the System State</b>	To visualize the state of the system you can use <b>State Machine</b> diagrams to describe how elements move between states, classifying their behavior according to transition triggers and constraining guards.  State Machine diagrams capture system changes over time, typically being associated with particular Classes; often a Class can have one or more State Machine diagrams to fully describe its potential states.	<a href="#">State Machine</a> 
<b>Visualize Package Arrangement</b>	<b>Package</b> diagrams help you design the architecture of the system; they are used to organize diagrams and elements into manageable groups, and to declare their dependencies.	<a href="#">Package</a> 
<b>Follow the Flow of Code</b>	<b>Activity</b> diagrams help you develop a better understanding of the flow of code.  Activity diagrams illustrate the dynamic nature of the system; you can model the flow of control between Activities and represent the changes in state of the system.	<a href="#">Activity</a> 

### Simple Class Diagram



### Notes

- You can use State Machine, Package and Activity diagrams to better understand the interaction between code elements and the arrangement of the code

### Learn more

- [Model Transformations - MDA](#)<sup>[2013]</sup>
- [Visual Execution Analysis](#)<sup>[2527]</sup>

## 2.5.6 Project Managers

Enterprise Architect provides support for the management of projects. **Project Managers** can use the system to assign resources to elements, measure risk and effort, estimate project sizes, and manage element status, change control and maintenance.

### Project Manager Tasks

Task	Detail	See also
<b>Provide Project Estimates</b>	In Enterprise Architect you have access to a comprehensive <b>project estimation</b> tool that calculates effort from Use Case and Actor objects, coupled with project configurations defining the technical and environmental complexity of the work environment.	<a href="#">Estimation</a> <sup>[584]</sup>

Task	Detail	See also
<b>Resource Management</b>	Managing the <b>allocation of resources</b> in the design and development of system components is an important and sometimes difficult task; Enterprise Architect provides you with an effective tool for assigning resources directly to model elements and tracking progress over time.	<a href="#">Resources</a> <sup>[512]</sup>
<b>Risk Management</b>	You can use the <b>Project Management window</b> to assign <b>risk</b> to an element within a project; using risk <b>types</b> you can name the risk, define the type of risk and give it a weighting.	<a href="#">Risk</a> <sup>[519]</sup> <a href="#">Risk Types</a> <sup>[524]</sup>
<b>Maintenance</b>	Within Enterprise Architect you can assign maintenance-related items to elements and track them, providing rapid capture and record keeping for items such as issues, changes, defects, and tasks.  You can also create and maintain a project glossary of processes, procedures, terms and descriptions.	<a href="#">Issues</a> <sup>[528]</sup> <a href="#">Changes and Issues</a> <sup>[2631]</sup> <a href="#">Tasks</a> <sup>[526]</sup> <a href="#">Project Glossary</a> <sup>[533]</sup>

#### Learn more

- [Project Management](#) <sup>[509]</sup>
- [Element Status](#) <sup>[590]</sup>
- [Maintenance](#) <sup>[2619]</sup>

## 2.5.7 Testers

Enterprise Architect provides a design testing facility for **Testers** and **Quality Assurance** personnel to create a range of test scripts against elements in the modeling environment.

#### Testing Tasks

Task	Detail	See also
<b>Test Cases</b>	With Enterprise Architect, you can set up a series of tests for each model element.  The test types include <b>Unit</b> , <b>Acceptance</b> , <b>System</b> , <b>Integration</b> and <b>Scenario</b> tests.	<a href="#">Testing</a> <sup>[2604]</sup>
<b>Import requirements, constraints and scenarios</b>	To use testing to maintain the integrity of the entire business process, you can import <b>requirements</b> , <b>constraints</b> and <b>scenarios</b> defined in earlier iterations of the development life cycle.  Requirements indicate contractual obligations that elements must perform within the model.	<a href="#">Requirements</a> <sup>[1726]</sup>

Task	Detail	See also
	<p>Constraints are conditions that must be met in order to pass the testing process; constraints can be:</p> <ul style="list-style-type: none"> <li>• Pre-conditions (states which must be true before an event is processed)</li> <li>• Post-conditions (events that must occur after the event is processed) or</li> <li>• Invariant constraints (which must remain true through the duration of the event)</li> </ul> <p>Scenarios are textual descriptions of an object's action over time and can be used to describe the way a test works.</p>	<p><a href="#">Constraints</a> <sup>[963]</sup></p> <p><a href="#">Scenarios</a> <sup>[965]</sup></p>
<b>Create quality test documentation</b>	Enterprise Architect provides the facility to generate high quality test <b>documentation</b> in the industry-standard .RTF file format.	<a href="#">Test Documentation</a> <sup>[2617]</sup>
<b>Element defect changes</b>	In defect tracking you can allocate defect reports to any element within the model, so that all who are involved in the project can quickly view the status of defects and see which defects have to be addressed and which have been dealt with.	<a href="#">Create Maintenance Items</a> <sup>[2625]</sup>

### 2.5.8 Implementation Managers

Enterprise Architect provides support for the management of **project implementation**. You can track and assign maintenance-related items to elements within Enterprise Architect, and rapidly capture and update records of maintenance tasks such as issues, changes, defects and tasks. By providing a centralized facility for each element involved in the deployment process Enterprise Architect offers a powerful solution for tracing the maintenance of the items and processes involved in system deployment.

#### Implementation Tasks and Tools

Task	Detail	See also
<b>Develop Deployment Diagrams</b>	<p>Using Deployment diagrams you can model the roll out of a project, including network deployment and workstation deployment.</p> <p>Users involved in project deployment can add maintenance tasks to the diagram elements.</p> <p>Deployment diagrams provide a static view of the run-time configuration of nodes on the network or of workstations, and the components that run on the nodes or are used in the workstations.</p>	<p><a href="#">Deployment Diagram</a> <sup>[1191]</sup></p> <p><a href="#">Example Deployment Diagram</a> <sup>[1193]</sup></p>

[Learn more](#)








- [Working on Maintenance Items](#) 

## 2.5.9 Technology Developers

**Technology Developers** are Enterprise Architect users who create customized additions to the functionality already present within Enterprise Architect.

Additions include UML Profiles, UML Patterns, Code Templates, Tagged Value Types, Scripts, Custom Queries, Transformations, MDG Technologies and Enterprise Architect Add-Ins. By creating these extensions the Technology Developer can customize the Enterprise Architect modeling process to specific tasks and speed up development.

### Developing Technologies

Extension	Detail	See also
<b>UML Profiles</b>	By creating <b>UML Profiles</b> you can create a customized extension for building UML models that are specific to a particular domain.  Profiles are stored as XML files and can be imported into any model as required.	<a href="#">Using UML Profiles</a> 
<b>UML Patterns</b>	<b>Patterns</b> are sets of collaborating Objects and Classes that provide a generic template for repeatable solutions to modeling problems.  As patterns are discovered in any new project, you can create the basic pattern template.  Patterns can be re-used with the appropriate variable names modified for any future project.	<a href="#">Design Patterns</a> 
<b>Code Templates</b>	<b>Code templates</b> are used to customize the output of source code generated by Enterprise Architect; in this way you can generate code languages not specifically supported by Enterprise Architect and define how the system generates source code to comply with your own company style guidelines.	<a href="#">Code Template Framework</a> 
<b>Tagged Value Types</b>	<b>Tagged Values</b> are used in Enterprise Architect to extend the information relating to an element in addition to the information directly supported by the UML language.  A Tagged Value, strictly, is the value of a property of a modeling item, the property being called a tag; for example: a Class element called Person might have a tag called Age with the Tagged Value of <b>42</b> .  More loosely, the combination of tag and value can be referred to as a Tagged Value.  A <b>Tagged Value Type</b> is a group of parameters that define and/or limit the possible values of a tag and, in many instances, how a specific value is assigned to the tag; for example, the tag <b>Age</b> might have a Tagged Value Type of Integer, so the user simply types in a numeric value.	<a href="#">Tagged Values</a>   <a href="#">Tagged Value Type</a> 

	<p>Alternatively, the type could be <b>Spin</b>, with lower and upper limits of, say, <b>20</b> and <b>120</b>, so the user sets a value by clicking on arrows in the field to increment or decrement the value within the limits of <b>20</b> and <b>120</b>.</p> <p>Typically, Tagged Values are used during the code generation process, or by other tools to pass on information that is used to operate on elements in particular ways.</p>	
<b>MDG Technologies</b>	<b>MDG Technologies</b> can be used to create a logical collection of resources that can contain UML Profiles, Patterns, Code Templates, Image files and Tagged Value types that are accessed through a technology file.	<a href="#">MDG Technologies</a> <sup>[1475]</sup>
<b>Enterprise Architect Add-Ins</b>	Using <b>Add-Ins</b> you can build your own functionality into Enterprise Architect, creating your own mini programs that can extend the capabilities of the system, defining your own menus, and creating your own Custom Views.	<a href="#">Enterprise Architect Add-Ins</a> <sup>[9010]</sup>

#### Learn more

- [Extending UML](#) <sup>[1483]</sup>
- [MDG Technology SDK](#) <sup>[1483]</sup>

## 2.5.10 Database Developers

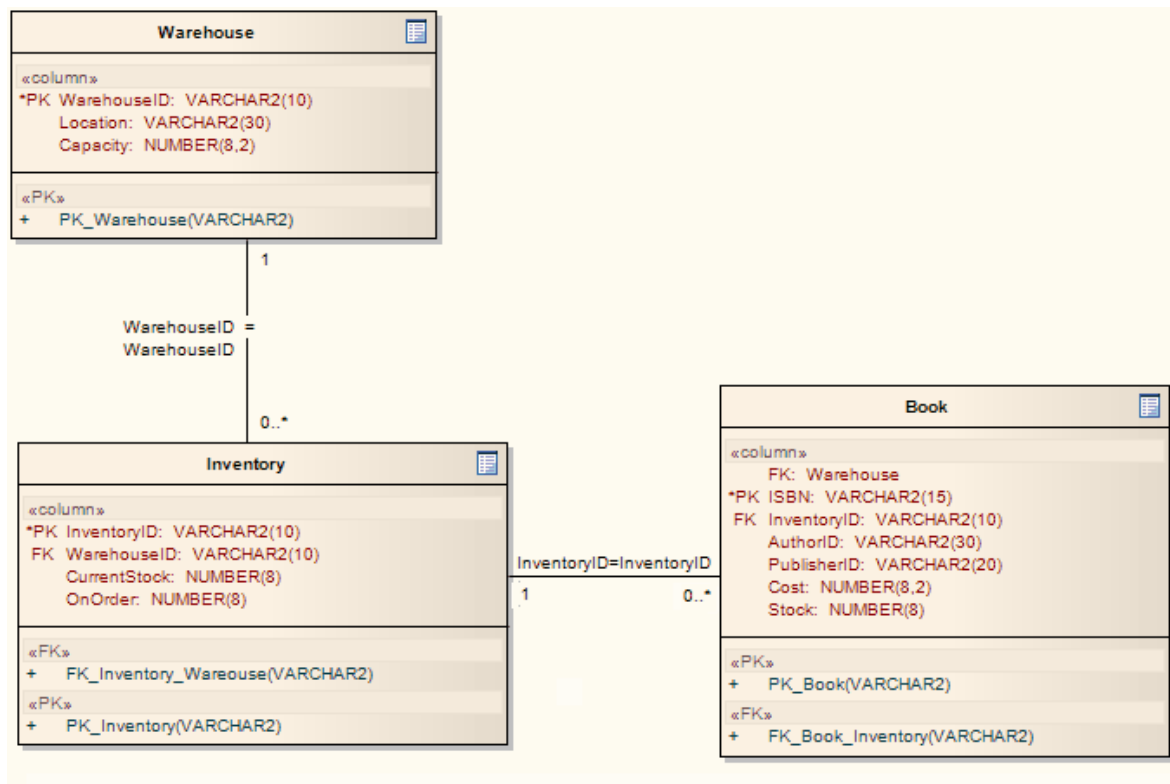
Enterprise Architect supports a range of features for the **development of databases**, including modeling database structures, importing database structures from an existing database and generating DDL for rapidly creating databases from a model.

#### Database Development Tasks

Task	Detail	See also
<b>Create Logical Data Models</b>	<p>With Enterprise Architect you can build database diagrams using the built-in UML Data Modeling Profile.</p> <p>This supports the definition of Primary and Foreign keys, cardinality, validation, triggers, constraints and indexes.</p>	<a href="#">Logical Data Model</a> <sup>[1939]</sup>
<b>Generate Schema</b>	<p>By using Enterprise Architect's DDL generation function you can create a DDL script to create the database table structure from the model.</p> <p>Enterprise Architect currently supports:</p> <ul style="list-style-type: none"> <li>• JET-based databases</li> <li>• DB2</li> <li>• InterBase</li> <li>• Informix</li> </ul>	<a href="#">Database Schema</a> <sup>[1943]</sup>

	<ul style="list-style-type: none"> <li>• Ingres</li> <li>• MySQL</li> <li>• SQL Server</li> <li>• PostgreSQL</li> <li>• Sybase Adaptive Server Anywhere and Adaptive Server Enterprise, and</li> <li>• Oracle 9i, 10g and 11g</li> </ul>	
<b>Reverse Engineer Database</b>	<p>Using an ODBC data connection you can import a database structure from an existing database to create a model of the database.</p> <p>By generating the model directly from the database you can quickly document your work and create a diagrammatic account of a complex database through the graphical benefits of UML.</p>	<a href="#">Reverse Engineering</a> [2136]

### Example Data Model Diagram

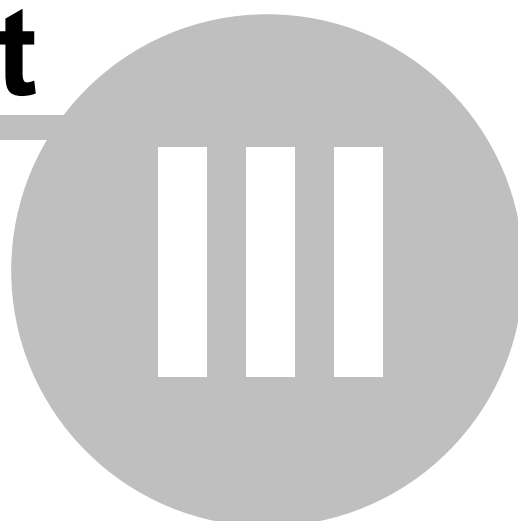


### Learn more

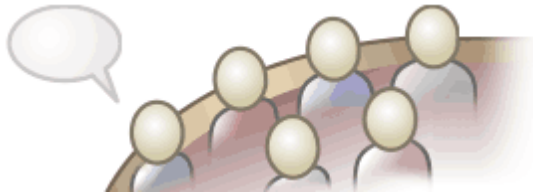
- [Data Models](#) [1937]
- [Physical Data Model](#) [2335]

# Part

---







### 3 Projects and Teams



Enterprise Architect helps you to create projects for development under a [range of work conditions](#)<sup>[199]</sup>, from single user/local access through to multiple-role teams working in a distributed environment. You both protect and manage the model data itself, and communicate information on the data in the form of documentation and reports, using facilities such as those listed below.

#### Topics

Topic	Link
Creating a project in a simple, file-based repository (.EAP file )	<a href="#">File-Based Repositories</a> <sup>[210]</sup>
Creating the project in one of a range of DBMS repositories - these provide for larger models with more concurrently connected users (Corporate and extended editions)	<a href="#">Server-Based Repositories</a> <sup>[214]</sup>
Tools for enabling team or multiple-user development in the project	<a href="#">Team Development</a> <sup>[306]</sup>
Tools for managing change within the model	<a href="#">Change Management</a> <sup>[382]</sup>
A variety of tools for managing modeling and project activities	<a href="#">Project Management</a> <sup>[509]</sup>
Maintaining the integrity of the project data	<a href="#">Project Maintenance</a> <sup>[596]</sup>
Sharing the reference data used across the project, between models and between projects	<a href="#">Sharing Reference Data</a> <sup>[374]</sup> <a href="#">Reference Data</a> <sup>[1146]</sup>
Enabling each user to configure their personal preferences for how project tools display and behave on their workstation	<a href="#">Local Options</a> <sup>[604]</sup>
You can have recorded discussion and communication of decisions using the Team Review	<a href="#">Team Review Tools</a> <sup>[843]</sup>
Managing work and resources allocated to tasks	<a href="#">Project Task Allocation</a> <sup>[538]</sup>

Topic	Link
Communicating and collaborating on model development through the Model Mail facilities	<a href="#">Model Mail</a> 
Record and manage personal work within the project.	<a href="#">Personal Tasks</a> 
Tracking important project events, resource issues and allocations through the Project Calendar	<a href="#">Project Calendar</a> 
Documenting your model through document or web reports, which you can tailor to your purposes	<a href="#">Reporting</a> 

### 3.1 Introduction



An Enterprise Architect project is stored in a data repository.

#### Topics

Topic	Detail	See also
<b>Project Files</b>	<p><b>Files</b></p> <p>In Enterprise Architect Desktop and Professional editions, you store a project in a single file with a .EAP or .FEAP extension. A .EAP file is a Microsoft JET database, so you can also open it using MS Access 97, 2000 or 2003, or any other reporting tool that can work with JET databases.</p> <p>A .FEAP file is a Firebird project file</p> <p><b>DBMS Repositories</b></p> <p>In Enterprise Architect Corporate Business and Software Engineering, Systems Engineering and Ultimate editions, you can also use a suitable DBMS database for project files.</p> <p>DBMS project files have the same logical structure as .EAP files, but must be connected to using ADO/ODBC (see <i>Connect to a Data Repository</i>, below).</p>	<p><a href="#">File Based Repositories</a> <sup>[210]</sup></p> <p><a href="#">Server Based Repositories</a> <sup>[214]</sup></p>
<b>Creating Project Files</b>	<p>On creating a new project, the Model Wizard enables you to create a model containing various Model Packages.</p> <p>You can also add Model Packages to a project from the Project Browser by:</p> <ul style="list-style-type: none"> <li>• Right-clicking on an existing model and selecting the <b>New Model</b> or <b>Add a New Model using Wizard</b> context menu options</li> <li>• Right-clicking on a Package and selecting the <b>Add   Add a New Model using Wizard</b> context menu option</li> <li>• Clicking on an existing model, pressing <b>Insert</b> and selecting the <b>New Model</b> or <b>Add a New Model using Wizard</b> context menu options</li> <li>• Clicking on a Package, pressing <b>Insert</b> and selecting the <b>Add a New Model using Wizard</b> context menu option</li> </ul>	<p><a href="#">File Based Repositories</a> <sup>[210]</sup></p> <p><a href="#">Model Wizard</a> <sup>[753]</sup></p>

Topic	Detail	See also
<b>Opening Existing Projects</b>	There are various ways to open a project in Enterprise Architect; new users are advised to explore the <i>EAExample</i> file supplied with Enterprise Architect.	<a href="#">Open a project</a> <sup>[202]</sup>
<b>Connect to a Data Repository</b>	<p>You can connect to any of the following data repositories:</p> <ul style="list-style-type: none"> <li>• MS Access 97, 2000 and 2003 (in all editions - .EAP files are stored in Microsoft JET databases)</li> <li>• Access 2007</li> <li>• SQL Server 2000, 2005, 2008 and 2012</li> <li>• SQL Server Express 2005 and 2008</li> <li>• MySQL</li> <li>• Oracle 9i, 10g and 11g</li> <li>• PostgreSQL</li> <li>• Adaptive Server Anywhere</li> <li>• Progress OpenEdge</li> </ul> <p>To create a new data repository, you first create a new database with the DBMS management software, then run supplied scripts to create the logical structure.</p> <p>You then use Enterprise Architect data transfer functions to move a project from a .EAP, .FEAP or DBMS model into the new project.</p>	<a href="#">Server Based Repositories</a> <sup>[214]</sup>

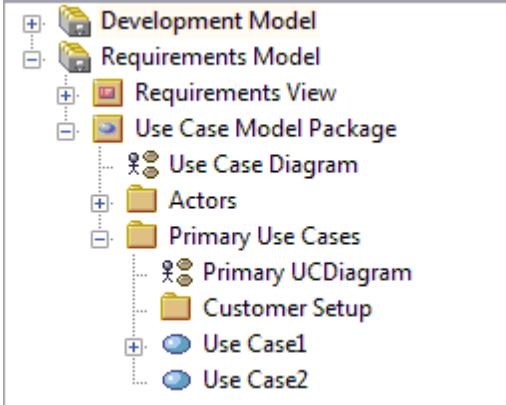
### 3.1.1 Projects Defined

An Enterprise Architect Project is a repository for storing, manipulating and managing one or more Models. A single repository can contain many models, and a repository can be either file based or hosted in a larger DBMS system.

#### Concepts

Concept	Detail	See also
<b>Project</b>	<p>A project can contain a single model, or a number of models, each of which defines a particular system or process. A model contains the diagrams, elements, relationships and associated metadata that define the structure and function of the system or process.</p> <p>These components are organized into a hierarchy of Packages, which help to group and manage related components.</p>	<a href="#">Open a Project</a> <sup>[202]</sup> <a href="#">Model Shortcuts</a> <sup>[204]</sup> <a href="#">Project Management</a> <sup>[509]</sup> <a href="#">Team Development</a> <sup>[306]</sup>
<b>Model Package</b>	Different aspects of the process or system - or their development - are defined by Model Packages, which you generate from templates	<a href="#">Model Wizard</a> <sup>[753]</sup>



Concept	Detail	See also
	<p>specifically structured to support the aspects that the Model Packages represent, such as requirements or deployment.</p> <p>You can generate these templated Packages at any level of the hierarchy, but as they are created with their own content they are more useful at the top levels.</p>	<a href="#">Model Templates</a> <sup>759</sup>
<b>View</b>	<p>The top-level Packages in a model can also be Views, which represent partitions of the model that you define yourself.</p> <p>You can start with standard Views such as Class or Component, or create whatever partitions are appropriate to your model.</p>	<a href="#">Views</a> <sup>769</sup>
<b>Example Project Structure</b>	 <p>Each View or Model Package contains Packages; <i>Use Case Model Package</i> contains:</p> <ul style="list-style-type: none"> <li>• Actors and</li> <li>• Primary Use Cases</li> </ul> <p>It also contains the diagram Use Case Diagram, which could be an overview of the Package structure or function.</p> <p>Each Package itself can contain one or more diagrams, one or more Packages, and several elements; the <i>Primary Use Cases</i> Package contains the:</p> <ul style="list-style-type: none"> <li>• Primary UCDiagram</li> <li>• Customer Setup Package</li> <li>• Use Case1 element</li> <li>• Use Case2 element</li> </ul> <p>Each subordinate Package also contains diagrams, elements and (if necessary) further Packages; the elements are related by connectors created in the diagrams, and each element and connector has properties, attributes, operations and extensions defined in the respective Properties dialogs.</p>	
<b>Storage</b>	A project can be a .EAP file in an MS Access database. a .FEAP file or	<a href="#">File Based</a>

Concept	Detail	See also
	(in the Enterprise Architect Corporate, Business and Software Engineering, System Engineering and Ultimate editions) a structure of files in a database management system such as MySQL or Oracle.	<a href="#">Repositories</a> <sup>[210]</sup> <a href="#">Server Based Repositories</a> <sup>[214]</sup>

### 3.1.2 Open a Project

An Enterprise Architect project is used for storing and managing the components of one or more UML models. The Desktop and Professional versions of Enterprise Architect work on file-based projects (.eap files). If you are using the Corporate edition (or above), you can also use one of a number of DBMSs such as Oracle or MySQL to host the project repository.

When you select to open a project, you can:

- Select a shortcut to a recent project, to open it
- Browse for another existing project to open
- Remove a shortcut from the list of recently-opened projects
- Specify connection details for opening a project hosted on a DBMS
- Create a new Enterprise Architect project file

**Access** **File | Open Project (Ctrl+O)**

#### Options

Field/Button/Option	Action	See also
<b>New Project...</b>	Click on this button to specify the location and name of a new .EAP or .FEAP project file.  Alternatively, click on the drop-down arrow and select the <b>Copy Base Project</b> option to create the new project from an existing project file. The system provides a standard base file as a template, but you can select any other .eap project file as a base.	<a href="#">File Based Repositories</a> <sup>[210]</sup>  <a href="#">Copy a Base Project</a> <sup>[211]</sup>
<b>Browse for Project</b>	Click on this button to open a file browser dialog, to select an existing .EAP file; the .EAP file can be a project file or a shortcut to a project hosted on a DBMS.	<a href="#">Model Shortcuts</a> <sup>[204]</sup>
<b>Connect to Server</b>	Click on this button to specify connection details for opening an Enterprise Architect project that is hosted on a DBMS. The <b>Windows</b> Data Link Properties dialog displays; start to define the connection to the project's server-based repository.  You can also click on the drop-down arrow and select the: <ul style="list-style-type: none"> <li>• <b>Connection Wizard</b> option - the Data Link Properties dialog again displays</li> </ul>	<a href="#">Server-Based</a>

	<ul style="list-style-type: none"> <li>• <b>Connection String</b> option - the Connection String dialog displays; type in or paste the connection string and click on the <b>OK</b> button to connect directly to the project</li> <li>• <b>Connect to Cloud</b> option - the Cloud Connection dialog displays, on which you specify the server, URL and model name to access via the Cloud</li> </ul>	<a href="#">Repositories</a> <sup>[214]</sup>  <a href="#">Connect to a Project Via the Cloud</a> <sup>[266]</sup>
<b>Recent Projects</b>	<p>This panel lists the ten most recently opened projects. To open one of these projects either double-click on the name, or click on it and then click on the <b>Open</b> button.</p> <p>Each project name is shown with its access path. You can edit the access path by right-clicking on it and selecting the <b>Edit Connection String</b> context menu option; make the changes on the Connection String dialog, and click on the <b>OK</b> button.</p> <p>This list is also displayed:</p> <ul style="list-style-type: none"> <li>• On the Start Page, under the <b>Recent</b> heading</li> <li>• As a drop-down menu under the <b>Open Project</b> button on the Default Tools toolbar</li> <li>• As part of the <b>File</b> menu</li> </ul>	
<b>Remove Selection from List</b>	Click on a project name and click on this button to remove that project name from the <b>Recent Projects</b> list.	
<b>Show this Dialog at Startup</b>	<p>Select the checkbox to show this Open Enterprise Architect Project dialog automatically upon starting Enterprise Architect.</p> <p>Deselect the checkbox to hide the dialog.</p>	
<b>Open</b>	Click on this button to open the currently selected project in the Recent Project list.	
<b>Cancel</b>	Click on this button to close this dialog without any further action.	
<b>Help</b>	Click on this button to display this Help topic.	

### Notes

- Use of a DBMS for hosting the model repository is available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect
- Enterprise Architect supports the following DBMS products for hosting model repositories; [SQL Server and SQL Server Express](#),<sup>[217]</sup> [MySQL](#),<sup>[222]</sup> [Oracle 9i, 10g or 11g](#),<sup>[253]</sup> [Postgre SQL](#),<sup>[233]</sup> [ASA](#),<sup>[239]</sup> and [Progress OpenEdge](#).<sup>[248]</sup>

Learn more

- [Projects Defined](#) <sup>[200]</sup>
- [File Based Repositories](#) <sup>[210]</sup>
- [DBMS Repositories](#) <sup>[214]</sup>
- [Team Development](#) <sup>[306]</sup>

### 3.1.3 Project Shortcuts

Enterprise Architect enables you to create a desktop shortcut (or *proxy* file) to an Enterprise Architect project (DBMS or file-based). Each shortcut is a file containing the connection string for the model.

However, the shortcut also defines *views* that Enterprise Architect should open when it opens the model, as outlined below.

Topics

View	Detail	See also
<b>Diagrams</b>	Define one or more specific diagram(s) to be loaded on opening the project.	
<b>Model Search</b>	<p>Open the Model Search with a specific text string and search type.</p> <p>For searches operating on the current tree selection, a diagram in the target package must be opened first.</p> <p>If you use a custom SQL search, the SQL must include <i>ea_guid</i> AS <i>CLASSGUID</i> and the object type.</p> <p>You specify a single Model Search to open.</p>	<a href="#">Model Search</a> <sup>[700]</sup>
<b>The Relationship Matrix with a saved profile</b>	<p>Open the Relationship Matrix with a saved profile.</p> <p>You specify a single Relationship Matrix profile to open.</p>	<a href="#">Relationship Matrix</a> <sup>[727]</sup>
<b>The default Team Review</b>	<p>Open the default Team Review document.</p> <p>You specify the Team Review once.</p>	<a href="#">Team Review Tools</a> <sup>[343]</sup>
<b>Working Set</b>	<p>Open a working set.</p> <p>You specify a single Working Set to open.</p> <p>This is very similar to the shortcut itself, opening a defined set of diagrams and views. However, the working set can <b>also</b> open source code editors, therefore widening the capabilities of the shortcut alone.</p> <p>Working sets make it easy to customize the main views you want to open without having to resave your project short cuts. The working set is easy to tailor to your changing requirements, adding greater flexibility to any short cut that calls the working</p>	<a href="#">Working Sets</a> <sup>[567]</sup>

View	Detail	See also
	<p>set</p> <p>Also, working sets provide the capability of storing the currently-engaged files when closing a model, and reopening them in the context in which you were last using them. The basic project shortcut can also do this, but once the work environment is captured the shortcut returns to the <b>same</b> set up each time you use it. The working set always captures the <b>current</b> work environment each time the model closes.</p> <p>This option can be very useful in, for example, a coding environment when you want to return to the last files you were editing.</p>	
<b>Workspace Layout</b>	<p>Apply a selected Workspace Layout, which opens and sets out the appropriate screens and windows for a specific area of work, such as model simulation.</p> <p>You specify a single Workspace Layout to open. However, by creating and using <b>separate shortcuts</b> you can open Enterprise Architect in exactly the configuration you need for the work you want to do - modeling, coding, or debugging, for example.</p>	<a href="#">Manage Workspace Layout</a> <small>163</small>
<b>An example shortcut</b>	<p>You might create a shortcut to open, in sequence:</p> <ul style="list-style-type: none"> <li>• A Development module</li> <li>• The Model Search for a simple search on the term <i>Issue</i></li> <li>• The module <i>Issues</i> diagram</li> <li>• The module <i>Changes</i> diagram</li> </ul> <p>Enterprise Architect opens the appropriate windows in the sequence in which you list the options, displaying the last view in the list as the active view.</p> <p>In this example, the project opens with the Enterprise Architect work area showing the two diagram tabs and the Model Search tab, and with the <i>Changes</i> diagram displayed in the Diagram View.</p>	

### Notes

- If specified, the shortcut views override any default diagram defined for the model or current user
- A shortcut does not affect the original Enterprise Architect .exe file or icon, or any other shortcut you might have defined; you can use all of these independently
- If you are using a database repository other than MS Access 97, 2000 or 2003, you can configure the shortcut to **encrypt the password** used to set up the connection between Enterprise Architect and the repository; the Enterprise Architect user does not have the real password, thereby preventing them from accessing the repository using other tools such as Query Analyzer or SQLPlus

Learn more

- [Create Project Shortcut](#)<sup>[206]</sup>
- [Capture Current Work Environment](#)<sup>[207]</sup>
- [Encrypt Repository Password](#)<sup>[209]</sup>
- [Diagram Menu](#)<sup>[93]</sup>
- [Creating Search Filters](#)<sup>[71]</sup>

**3.1.3.1 Create Project Shortcut**

You can create a shortcut to an Enterprise Architect project (either a DBMS project or file based project). The shortcut can specify additional windows and diagrams to open up automatically every time the shortcut is run, to create a working environment in advance for other users.

Shortcuts are stored with a .EAP extension, but are actually small text files that tell Enterprise Architect what project to open and what initial views and windows to display.

**Access**   **File | Save Shortcut**

Create a project shortcut

Step	Action	See also
1	Open Enterprise Architect.	
2	Open the required project.	
3	Select the <b>File   Save Shortcut</b> menu option. The Save Project Shortcut dialog displays.	
4	Click on the ( ... ) (Browse) button at the end of the <b>Target File</b> field. The Save Project As dialog displays.	
5	Browse for the appropriate file location and, in the <b>File name</b> field, type an appropriate filename.  All shortcuts are .EAP files, regardless of whether the model itself is a .EAP file or a DBMS model.	
6	Click on the <b>Save</b> button to return to the Save Project Shortcut dialog.	
7	Click on the <b>Add Other</b> button and select the required option to define: <ul style="list-style-type: none"> <li>• A diagram to open</li> <li>• A Relationship Matrix profile to open</li> </ul>	

Step	Action	See also
	<ul style="list-style-type: none"> <li>• The Team Review</li> <li>• A Model Search to perform</li> <li>• A working set to apply</li> <li>• A workspace layout to apply</li> </ul>	
8	<p>The appropriate browser or dialog displays to define the view to display; enter the details and click on the <b>OK</b> button.</p> <p>The view is added to the <b>Actions when model is opened</b> field; the entry is automatically selected, with a tick in the checkbox.</p>	
9	Repeat steps 7 and 8 for as many additional views as you require.	
10	<p>Review the items in the <b>Actions when model is opened</b> field.</p> <ul style="list-style-type: none"> <li>• If you decide not to have an item in the shortcut, deselect its checkbox</li> <li>• If you want to clear all selected items, click on the <b>Include None</b> button</li> </ul> <p>Unselected entries are deleted when you save the shortcut.</p>	
11	<p>If you decide to change the sequence and/or make a different view display first in the Diagram View:</p> <ul style="list-style-type: none"> <li>• Click on the appropriate entry</li> <li>• Click on the <b>'Up Hand'</b> or <b>'Down Hand'</b> buttons</li> </ul>	
12	Click on the <b>OK</b> button to save the shortcut.	

### Notes

- When you subsequently open the Save Project Shortcut dialog, it lists the currently-opened views in the order in which they were opened; you can add further views or remove them from the shortcut

### Learn more

- [Capture Current Work Environment](#)<sup>[207]</sup>
- [Encrypt Repository Password](#)<sup>[209]</sup>

### 3.1.3.2 Capture Current Work Environment

You can capture the current Enterprise Architect work environment in your shortcut. This maintains the work environment from that point, and is useful if you intend to close and re-open the model and return to **the same** configuration many times.

If you simply want to return to **whatever** work environment you happen to be in each time you close the

model, you should use a **work set** that captures the current environment, either by itself or as part of a broader project shortcut.

**Access** **File | Save Shortcut**

#### Capture your current work environment

Step	Action	See also
1	Open Enterprise Architect.	
2	Open the required project and work in it.	
3	At the point at which you decide to capture your work environment in a shortcut, ensure that: <ul style="list-style-type: none"> <li>You have opened all diagrams you require</li> <li>If necessary, you have opened the Team Review, Model Search (with appropriate search term and type) and/or Relationship Matrix (at the appropriate profile)</li> <li>The view you want to resume work on is the last one opened</li> </ul>	
4	Select the <b>Save Project Shortcut</b> menu option.  The Save Project Shortcut dialog displays, showing a list of actions derived from the views you currently have open.	
5	If you accessed Enterprise Architect via a shortcut, the <b>Target File</b> field displays the file location of that shortcut.  Otherwise, click on the ( ... ) (Browse) button at the end of the <b>Target File</b> field.  The Save Project As dialog displays.	
6	Browse for the appropriate file location and, in the <b>File name</b> field, type an appropriate filename.  All shortcuts are .EAP files, regardless of whether the model itself is a .EAP file, .FEAP file or DBMS model.	
7	Click on the <b>Save</b> button to return to the Save Project Shortcut dialog.	
8	In the <b>Actions when model is opened</b> field, click on the <b>Include All</b> button.	
9	If you also want to save current window positions click <b>Add Other</b> followed by <b>Add Workspace Layout...</b> The dialog shown allows you to select an existing layout or save the current layout.	<a href="#">Manage Workspace Layout</a> <sup>[163]</sup>



Step	Action	See also
10	Click on the <b>OK</b> button to save the shortcut.	

#### Learn more

- [Working Sets](#) <sup>[564]</sup>
- [Encrypt Repository Password](#) <sup>[209]</sup>

### 3.1.3.3 *Encrypt Repository Password*

If your model is developed on a DBMS repository, the Save Project Shortcut dialog has an **Encrypt Connection String** check box.

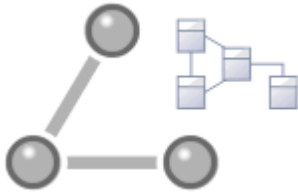
You can create the shortcut actions and, if necessary, select the checkbox to encrypt the database connection string.

You distribute the shortcut file to the database users who are to access the model. The users then have an encrypted string that prevents them from directly accessing the database using other tools.

#### Learn more

- [Create Project Shortcut](#) <sup>[206]</sup>

## 3.2 File Based Repositories



Having a light weight, portable, file-based repository for Enterprise Architect is ideal for those who want to quickly develop modeling suggestions and solutions and share the model information easily with others. All editions of Enterprise Architect support this type of repository in the form of .eap files in an Access database, or .feap files on a Firebird repository.

A default project file (*EABase.eap*) is supplied in the Enterprise Architect installer and is automatically stored in your Enterprise Architect installation directory. You can:

- Copy the EABase.eap file as the base for your own projects
- Create your own base projects as .eap files or .feap files, and
- Copy existing operational projects

Task	Detail	See also
<b>Create .EAP Project Files</b>	<p>Select one of:</p> <ul style="list-style-type: none"> <li>• <b>File   New Project</b> menu option</li> <li>• <b>New Project   New Project</b> option on the Open Enterprise Architect Project dialog, or</li> <li>• <b>Create a New Project</b> option on the Start Page</li> </ul> <p>All of these options display the New Project dialog; select a directory and enter a file name for your project.</p> <p>Once the project has been saved, the Model Wizard displays, which makes a selection of Model Patterns available; select the Model Patterns to use.</p> <p>Enterprise Architect adds a model containing the selected Model Packages to the Project Browser.</p>	<a href="#">Model Wizard</a> <sup>[753]</sup>
<b>Create .FEAP Project Files</b>	<p>As for a .EAP file (above), except that in the New Project dialog, in the <b>Save as Type</b> field, click on the drop-down arrow and select the <b>.feap</b> file extension.</p> <p>Enterprise Architect will check your input and display an error message if:</p> <ul style="list-style-type: none"> <li>• Your Firebird repository is not located on the local drive; a Firebird repository is not appropriate for a network shared project</li> <li>• The file path contains characters that are not in the system codepage</li> </ul>	

Task	Detail	See also
<b>Default EABase. EAP Installation directories</b>	<p>The default installation directories, depending on which version you have installed, are:</p> <ul style="list-style-type: none"> <li>Registered version: C:\Program Files\Sparx Systems\EA</li> <li>Trial version: C:\Program Files\Sparx Systems\EA Trial</li> <li>Lite version: C:\Program Files\Sparx Systems\EA Lite</li> </ul> <p>Having copied the base project as a template for your own project, you can rename it.</p>	<a href="#">Copy a Base Project</a> <sup>[211]</sup> <a href="#">Rename a Project</a> <sup>[601]</sup>
<b>Configure Project</b>	Having created your project, you can set a range of project parameters to define defaults, tailor the project to particular coding languages, and ensure consistent development and use of the project.	<a href="#">Settings Menu</a> <sup>[119]</sup> <a href="#">Defaults and User Settings</a> <sup>[151]</sup>
<b>Create Custom Templates</b>	You can customize any Enterprise Architect project as a template project with company standards, tutorials, frameworks and any other common piece of modeling already in-built; with careful planning you can save yourself many hours of work at project start-up.	<a href="#">Incorporate Model Templates</a> <sup>[1576]</sup>
<b>Copy Existing Project</b>	<p>A Base project contains templates and reference data to enable you to develop your own project quickly.</p> <p>You can also copy an operational .EAP project file to enable separate development by team members, or to create an evaluation or distribution version of the project.</p>	<a href="#">Copy Existing Projects</a> <sup>[212]</sup>

**Notes**

- You can also add Model Packages to a project using the **New Model From Pattern** icon in the Project Browser toolbar

**3.2.1 Copy a Base Project**

When you create a new project, you can use the Model Wizard to define the structure and contents. Alternatively, you can copy an existing **template** or **base project** which is already largely set up, containing company standards, tutorials, frameworks and any other common modeling structures.

If you intend to use an existing project file as the template for a new project, it is **important** to use this method rather than simply copying the .EAP file using Windows Explorer. This process **resets all the unique identifiers** for packages and elements, so that your new project is truly **unique** - otherwise it is simply an exact copy of the original project.

**Access** **File | Open Project (Ctrl+O) : New Project (arrow) | Copy Base project**

Create a new project from a base project

Step	Action	See also
1	On the Create New Enterprise Architect Project dialog, in the <b>New Project</b> field, click on the <b>Browse</b> button and select the file path for saving your project.  If this is to be a shared project, store the file on a shared network resource such as a Network Server or Workgroup Server.	
2	In the <b>Model Project</b> field, select the project that is the base template.  The field defaults to <b>EABase.eap</b> , the default base project provided in the Enterprise Architect installer.  If you want to use a different base, such your own custom project, click on the <b>Browse</b> button and locate the file to use.	<a href="#">Incorporate Model Templates</a> <sup>[1576]</sup>
3	The <b>Reset New Project GUIDs</b> checkbox defaults to selected, to replace all GUIDs from the base model with fresh GUIDs in the new model.  However, if the new project is based on one that is under version control, we recommend that you <b>deselect</b> this checkbox to prevent duplication of Packages when the <i>Get Latest</i> facility is used.	<a href="#">Package Version Control Menu</a> <sup>[427]</sup>
4	Click on the <b>Create Project</b> button to create your project.	

### 3.2.2 Copy Existing Project

You can copy an operational .EAP or .FEAP project file to a new file location under a new name to, for example:

- Provide separate copies for individual team members
- Create an evaluation or distribution version of the project

**Access**   **File | Save Project As**

#### Create a copy of a project

Step	Action	See also
1	Select the <b>Save Project As</b> menu option.  The Save Enterprise Architect Project Copy dialog displays.	
2	In the <b>Target Project</b> field, type the new project filename.  You can either type in the file path as well, or click on the <b>Browse</b> button and browse for the required file path.	

Step	Action	See also
3	<p>If this copy is to be used for a different project, select the <b>Reset New Project GUIDs</b> checkbox.</p> <p>This ensures that all components of the new project have different IDs to their counterparts in the source project.</p>	
4	<p>Click on the <b>Save As</b> button to save the new project.</p>	

### 3.3 Server Based Repositories



You can create a model in any edition of Enterprise Architect as a .eap or .feap file. However, if you need to develop **large** models with **many concurrently-connected users**, you might prefer to use the greater capacity and flexibility of a Data Base Management System (DBMS) and create your project on a DBMS (server-based) repository. This facility is available through the Corporate, Business and Software Engineering, Systems Engineering or Ultimate editions.

#### Set up a project on a DBMS repository

To set up your project on a DBMS repository, you work through these stages:

1. Set up your DBMS software and create a repository.
2. Create the required tables in your repository, by running a script supplied on the Sparx Systems website.
3. (For certain DBMS products) Set up an ODBC driver to enable connection to the repository.
4. Transfer the project from the source file to the DBMS repository; the source file can be:
  - a .eap or .feap template file, to begin a new project from scratch, or
  - a previously-developed project file, to move an existing project into the DBMS repository
5. Connect to your repository.

#### DBMS Products you can use

You can set up your project on a repository in:

- SQL Server 2000, 2005, 2008 or 2012
- SQL Server Express 2005 or 2008
- MySQL 4 or 5
- PostgreSQL 7, 8 or 9
- Adaptive Server Anywhere 8 or 9, or SQL Anywhere 10, 11 or 12
- Access 2007
- Progress OpenEdge, or
- Oracle 9i, 10g or 11g

For information on creating a project on a specific DBMS from this list, see *Learn more*, below.

#### Notes

- You cannot move a model from a source .eap file of an Enterprise Architect version earlier than 3.5.0 without updating it first
- Before proceeding, you must have MDAC 2.6 or higher installed on your system

- (Optional, but recommended) before actually transferring the project structure from the file to the repository, perform a Project Data Integrity Check on the file

#### Learn more

- [Create a project in a SQL Server database](#)<sup>[215]</sup>
- [Create a project in a MySQL database](#)<sup>[220]</sup>
- [Create a project in a PostgreSQL database](#)<sup>[227]</sup>
- [Create a project in a Sybase ASA database](#)<sup>[234]</sup>
- [Create a project in an Access 2007 database](#)<sup>[241]</sup>
- [Create a project in a Progress OpenEdge database](#)<sup>[243]</sup>
- [Create a project in an Oracle database](#)<sup>[250]</sup>

### **3.3.1 Create a Project in a SQL Server Database**

To create a project on a repository in SQL Server 2000, 2005, 2008 or 2012, or SQL Server Express 2005 or 2008, work through the following stages:

- Stage 1: Create an empty database repository and set up the data tables
- Stage 2: Perform a project integrity check on the project file you are using as a base (optional, but recommended)
- Stage 3: Transfer the data
- Stage 4: Connect to the repository to open the project


#### Prerequisites

- Install SQL Server and MDAC 2.6 or higher, and obtain access permissions to create a new database


#### Create a SQL Server Repository

Step	Action	See also
1	Create an empty SQL Server database.	<a href="#">Create a SQL Server Repository</a> <sup>[217]</sup> <a href="#">Registered User's Scripts page</a> <a href="#">Trial User's Scripts Page</a>
2	Using a tool such as the SQL Enterprise Manager, load the <b>SQLServer_EASchema.sql</b> file from the Sparx Systems website.	
3	Run the script to create all required data tables in the repository.	

#### Perform a Project Integrity Check

Step	Action	See also
1	In Enterprise Architect, open the file-based project or template from which you are creating the project on the repository.	
2	Select and run <b>Tools   Data Management   Project Integrity Check</b> . This ensures your project data is 'clean' before being copied to the repository.	<a href="#">Project Integrity Check</a> 

#### Transfer the project data to the repository

Step	Action	See also
1	Open Enterprise Architect. (If the Open Project screen displays, click on the <b>Cancel</b> button to open with no project loaded.)	
2	Select the <b>Tools   Data Management   Project Transfer</b> menu option. The Project Transfer dialog displays.	
3	In the Transfer Type panel, select <b>.EAP to DBMS</b> .	
4	In the <b>Source Project</b> field, type the name of the project file to copy to the repository. If the .EAP file has Replication enabled, this must be removed before performing the transfer.	<a href="#">Remove Replication</a> 
5	At the right of the <b>Target Project</b> field, click on the ( ... ) (Browse) button. The Datalink Properties dialog displays.	
6	Select <b>Microsoft OLE DB Provider for SQL Server</b> from the list. Click on the <b>Next</b> button.	
7	On the Data Source Details page of the Datalink Properties dialog, type in the server name, database name and any security details required. Click on the <b>OK</b> button. The Project Transfer dialog redisplay.	
8	If required, select the <b>Logfile</b> checkbox and type a path and filename for the data transfer log file.	



Step	Action	See also
9	Click on the <b>Transfer</b> button to begin the data transfer process.	
10	When the process is complete, you have created a project on a SQL Server database and can now connect to it and open it from Enterprise Architect.	<a href="#">Connect to a SQL Server Data Repository</a> <sup>[218]</sup>

#### Learn more

- [Perform a Project Data Transfer](#)<sup>[504]</sup>

### 3.3.1.1 Create a SQL Server Repository

Creating a SQL Server database is fully documented in the product information provided with your SQL Server installer.

Having created your SQL Server database, use the SQL script (**SQLServer\_EASchema.sql**) provided by Sparx Systems to create the required table definitions for Enterprise Architect. You can obtain the script from the Sparx Systems website, on the:

- Registered Corporate edition Resources page (Registered users)
- Corporate edition Resources page (Trial users)

If you are unfamiliar with SQL Server and DBMS systems in general, you might want to consider a suitable front end tool, such as SQL Enterprise Manager.

#### SQL Enterprise Manager example

Step	Action
1	In SQL Enterprise Manager, locate the server on which to create your new project; for example: DBSERVER02\SQLEXPRESS.
2	Right-click and choose the <b>New Database</b> context menu option.
3	Enter a suitable name for the database. Set any file options as required. Ensure that the database collation is set to the alphabet you use, such as Latin or Cyrillic, and case-insensitive.
4	Click on the database to select it, then select the <b>New Query</b> menu option.
5	In the Query window, use the Open File dialog to locate the <i>SQLServer_EASchema.sql</i> script file.

Step	Action
6	Click on the <b>Open</b> button. In the drop-down menu, check that you have selected the correct database to run the script in.
7	Click on the <b>Execute</b> button; SQL Server executes the script, which creates the base tables for an Enterprise Architect project.

#### Notes

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions
- When creating a project in a SQL Server database you must have **db\_ddladmin** permission in order to execute the **SET IDENTITY\_INSERT ( table ) {ON | OFF}** command

#### Learn more

- [Create a project in a SQL Server database](#)<sup>[215]</sup>
- [Registered User's Scripts page](#)
- [Trial User's Scripts Page](#)

### 3.3.1.2 Connect to a SQL Server Data Repository

To access a project in your SQL Server data repository, you need to connect to the data repository from Enterprise Architect.

#### Prerequisites

- The SQL Server repository and the project already exist
- You have **SELECT**, **UPDATE**, **INSERT** and **DELETE** access permissions

Access   **File | Open Project ( Ctrl+O )**

#### Connect to the repository

Step	Action
1	In the Open Project dialog, select the <b>Connect to Server</b> checkbox.
2	Click on the ( ... ) (Browse) button. The Data Link Properties dialog displays.

Step	Action
3	Select <b>Microsoft OLE DB Provider for SQL Server</b> from the list.
4	Click on the <b>Next&gt;&gt;</b> button. The Connection tab displays.
5	Type in the server details, including Server Name, User Name and Password.
6	Click on the <b>Select the database on the server</b> option and on the drop-down arrow. From the list, select the project to connect to.
7	Click on the <b>Test Connection</b> button to confirm that the details are correct.
8	If the test does not succeed, revise your settings. If the test succeeds, click on the <b>OK</b> button; the Connection Name & Type dialog displays.
9	Give the connection a suitable name so that you can recognize it in the Recent Projects panel on the Open Project dialog.
10	If required, select the <b>Encrypt Connection String</b> checkbox. This encrypts and hides the connection details of the database from the users that the connection string is given to.
11	If required, select the <b>Lazy Load</b> checkbox to not load the full project view when the model is loaded; instead, only the parts that are necessary to display the visible portion of the tree are loaded. With this setting, the model loads faster and users can begin work sooner, but at the expense of later small delays as Enterprise Architect loads specific portions of the model.
12	If required, select the <b>Use WAN Optimization</b> checkbox. To improve performance over a Wide Area Network, remote database calls can be routed through a WAN Optimizer that compresses the data returned from the repository, reducing transfer time. If you select this checkbox, complete the next two fields (see your administrator for the correct values); otherwise go to step 15.
13	In the <b>Server</b> field, type the network name or address of the optimizer server.
14	In the <b>Port</b> field, type the port on which the server is running on the remote machine.

Step	Action
15	Click on the <b>OK</b> button to complete the configuration and open the project.  This also adds the project name to the <b>Recent Projects</b> list on the Start Page; from now on, you can open the project again just by clicking on this name.

#### Notes

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions

#### Learn more

- [Create a project in a SQL Server database](#)<sup>[215]</sup>
- [Create a SQL Server Repository](#)<sup>[217]</sup>
- [The WAN Optimizer](#)<sup>[259]</sup>
- [Open a Project](#)<sup>[202]</sup>

### 3.3.2 Create a Project in a MySQL Database

To create a project in a repository on MySQL 4 or 5, you work through the following stages:

- Stage 1: Create an empty database repository and set up the data tables
- Stage 2: Set up the ODBC Driver
- Stage 3: Perform a project integrity check on the .project file you are using as a base (optional, but recommended)
- Stage 4: Transfer the data
- Stage 5: Connect to the repository and open the project

#### Prerequisites

- Install MySQL version 4.0.3 or higher, and MySQL ODBC 5.1 or higher

#### Create a MySQL Repository

If MySQL is set up to use **MyISAM** table types (the default), transactional support is disabled. To enable transactions you must set up MySQL to use **InnoDB** tables and create the database tables as InnoDB type.

Sparx Systems provide scripts to create both InnoDB- and MyISAM-based repository tables.

Step	Action	See also
1	Create an empty MySQL database.	<a href="#">Create a MySQL Repository</a>
2	Set up the <i>MySQL .ini</i> file.	

Step	Action	See also
3	<p>To use <i>MyISAM</i> tables, run the <b>MySQL_MyISAM_EASchema.sql</b> script.</p> <p>To use <i>InnoDB</i> tables, run the <b>MySQL_InnoDB_EASchema.sql</b> script.</p>	<p><a href="#">Registered User's Scripts page</a></p> <p><a href="#">Trial User's Scripts Page</a></p>

### Set Up the ODBC Driver

Step	Action	See also
1	<p>Create a suitable ODBC Data Source to point to your new database.</p> <p>Select the following checkboxes:</p> <ul style="list-style-type: none"> <li>• <b>Return matched rows instead of affected rows</b></li> <li>• <b>Allow big result sets</b></li> </ul>	<p><a href="#">Set up a MySQL ODBC Driver</a></p>

### Perform a Project Integrity Check

Step	Action	See also
1	In Enterprise Architect, open the file-based project or template from which you are creating the project on the repository.	
2	<p>Select and run <b>Tools   Data Management   Project Integrity Check</b>.</p> <p>This ensures your project data is 'clean' before being copied to the repository.</p>	<p><a href="#">Project Integrity Check</a></p>

### Transfer the Project Data

Step	Action	See also
1	<p>Open Enterprise Architect.</p> <p>(If the Open Project screen displays, click on the <b>Cancel</b> button to open with no project loaded.)</p>	
2	<p>Select the <b>Tools   Data Management   Project Transfer</b> menu option.</p> <p>The Project Transfer dialog displays.</p>	

Step	Action	See also
3	In the Transfer Type panel, select <b>.EAP to DBMS</b> .	
4	In the <b>Source Project</b> field, type the name of the project file to copy to the repository.  If the .EAP file has Replication enabled, this must be removed before performing the transfer.	<a href="#">Remove Replication</a> <sup>[313]</sup>
5	At the right of the <b>Target Project</b> field, click on the ( ... ) (Browse) button.  The Datalink Properties dialog displays.	
6	Select <b>Microsoft OLE DB Provider for ODBC Drivers</b> from the list.  Click on the <b>Next</b> button.	
7	In the <b>Use Data source name</b> field, click on the drop-down arrow and select the ODBC Data Source you configured to point to your new database.  Click on the <b>OK</b> button. The Project Transfer dialog redisplay.	<a href="#">Set up a MySQL ODBC Driver</a> <sup>[223]</sup>
8	If required, select the <b>Logfile</b> checkbox and type a path and filename for the data transfer log file.	
9	Click on the <b>Transfer</b> button to begin the data transfer process.	
10	When the process is complete, you have created a project on a MySQL database and can now connect to it and open it from Enterprise Architect.	<a href="#">Connect to a MySQL Data Repository</a> <sup>[225]</sup>

#### Learn more

- [Perform a Project Data Transfer](#)<sup>[504]</sup>

### 3.3.2.1 Create a MySQL Repository

Creating a MySQL database is fully documented in the product information provided with your MySQL installer.

Having created your MySQL database, use the SQL script provided by Sparx Systems to create the required table definitions for Enterprise Architect.

You can obtain the scripts from the Sparx Systems website, on the:

- Registered Corporate edition Resources page (Registered users)
- Corporate edition Resources page (Trial users)

### **Third Party Tools**

If you are unfamiliar with MySQL and DBMS systems in general, you might want to consider a suitable front end tool. MySQL Administrator is one such tool, providing a convenient graphical user interface to enable the creation of databases, execution of scripts, backups and restores.

You might, therefore:

1. Run MySQL Administrator and create a new database, then
2. Run MySQL Query Browser, and open and execute the MySQL repository script

After creating a MySQL data repository in Enterprise Architect, you must set up the MySQL ODBC driver.

### **Notes**

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions
- Ensure that the collation is set to the alphabet you use, such as Latin or Cyrillic

### **Learn more**

- [Create a project in a MySQL database](#) <sup>220</sup>
- [Registered User's Scripts page](#)
- [Trial User's Scripts Page](#)
- [MySQL Workbench \(including MySQL Administrator\) download site](#)
- [Set up a MySQL ODBC Driver](#) <sup>223</sup>

### **3.3.2.2 Set up a MySQL ODBC Driver**

After you have created a repository on your MySQL database, you set up the MySQL ODBC driver so that you can connect to a project on that repository from Enterprise Architect.

### **Prerequisites**

Install:

- MySQL DBMS and repository
- MySQL ODBC driver software version 5.1.5

### **Set up the ODBC Driver**

Step	Action
1	<p>If you are using a 32-bit operating system:</p> <ul style="list-style-type: none"> <li>• Select the Windows™ <b>Control Panel   Administrative Tools   Data Sources (ODBC)</b> option</li> </ul> <p>If you are using a 64-bit operating system:</p>

Step	Action
	<ul style="list-style-type: none"> <li>Enterprise Architect requires 32-bit ODBC drivers to connect to a repository through ODBC; to set up the ODBC configuration on <b>64-bit clients</b>, run the 32-bit ODBC Data Source Administrator from <i>C:\Windows\SysWOW64\odbcad32.exe</i></li> <li>You can quickly load the correct 32-bit ODBC Data Source Administrator by selecting <b>Tools   ODBC Data Sources</b> in the Enterprise Architect menu bar</li> </ul> <p>The ODBC Data Source Administrator window displays.</p>
2	<p>Click on the <b>Add</b> button.</p> <p>The Create New Data Source dialog displays, enabling you to add a new DSN.</p>
3	<p>Select <b>MySQL ODBC 5.1 Driver</b> from the list.</p>
4	<p>Click on the <b>Finish</b> button.</p> <p>The MySQL Connector/ODBC dialog displays.</p>
5	<p>Enter the following configuration details:</p> <ul style="list-style-type: none"> <li>A data source name for the connection</li> <li>A description (optional)</li> <li>The host address of the DBMS server</li> <li>User name and password</li> <li>The database name on the selected server</li> </ul>
6	<p>To set the advanced options, click on the <b>Details&gt;&gt;</b> button.</p>
7	<p>Select the following checkboxes (where provided):</p> <ul style="list-style-type: none"> <li><b>Return matched rows instead of affected rows</b> (Connection or Cursors/Results tab)</li> <li><b>Allow big result sets</b> (Connection tab)</li> </ul>
8	<p>Click on the <b>Test Connection</b> button to confirm that the details are correct.</p>
9	<p>If the test succeeds, click on the <b>OK</b> button to complete the configuration.</p> <p>If the test does not succeed, review your settings.</p>

Your MySQL driver is now available to connect to the repository from Enterprise Architect.



Learn more

- [Create a project in a MySQL database](#)<sup>[220]</sup>
- [Create a MySQL Repository](#)<sup>[222]</sup>
- [Connect To a MySQL Data Repository](#)<sup>[225]</sup>

**3.3.2.3 Connect to a MySQL Data Repository**

To access a project in your MySQL data repository, you need to connect to the data repository from Enterprise Architect.

Prerequisites

- The MySQL repository and the project already exist
- You have **SELECT**, **UPDATE**, **INSERT** and **DELETE** access permissions
- The MySQL ODBC driver has been set up

Access   **File | Open Project (Ctrl+O)**

Connect to the repository

Step	Action
1	In the Open Project dialog, select the <b>Connect to Server</b> checkbox.
2	Click on the ( ... ) (Browse) button. The Data Link Properties dialog displays.
3	Select <b>Microsoft OLE DB Provider for ODBC Drivers</b> from the list.
4	Click on the <b>Next&gt;&gt;</b> button. The Connection tab displays.
5	Click on the <b>Use data source name</b> radio button and on the drop-down arrow in its field. From the list, select the ODBC driver you have set up to connect to your MySQL repository.
6	If required, type in a <b>User</b> name and <b>Password</b> .
7	If required, type in an initial catalog.
8	Click on the <b>Test Connection</b> button to confirm that the details are correct.

Step	Action
9	<p>If the test does not succeed, revise your settings.</p> <p>If the test succeeds, click on the <b>OK</b> button; the Connection Name &amp; Type dialog displays.</p>
10	<p>Give the connection a suitable name so that you can recognize it in the Recent Projects panel on the Open Project dialog.</p>
11	<p>If required, select the <b>Encrypt Connection String</b> checkbox.</p> <p>This encrypts and hides the connection details of the database from the users that the connection string is given to.</p>
12	<p>If required, select the <b>Lazy Load</b> checkbox to not load the full project view when the model is loaded; instead, only the parts that are necessary to display the visible portion of the tree are loaded.</p> <p>With this setting, the model loads faster and users can begin work sooner, but at the expense of later small delays as Enterprise Architect loads specific portions of the model.</p>
13	<p>If required, select the <b>Use WAN Optimization</b> checkbox.</p> <p>To improve performance over a Wide Area Network, remote database calls can be routed through a WAN Optimizer that compresses the data returned from the repository, reducing transfer time.</p> <p>If you select this checkbox, complete the next three fields (see your administrator for the correct values); otherwise go to step 17.</p>
14	<p>In the <b>Server</b> field, type the network name or address of the optimizer server.</p>
15	<p>In the <b>Port</b> field, type the port on which the server is running on the remote machine.</p>
16	<p>In the <b>DSN</b> field, type the data source name of the database as it appears on the remote machine.</p>
17	<p>Click on the <b>OK</b> button to complete the configuration and open the project.</p> <p>This also adds the project name to the <b>Recent Projects</b> list on the Start Page; from now on, you can open the project again just by clicking on this name.</p>

#### Notes

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions

#### Learn more

- [Create a project in a MySQL database](#)<sup>[220]</sup>
- [Create a MySQL Repository](#)<sup>[222]</sup>
- [Set up a MySQL ODBC Driver](#)<sup>[223]</sup>
- [The WAN Optimizer](#)<sup>[259]</sup>
- [Open a Project](#)<sup>[202]</sup>

### 3.3.3 Create a Project in a PostgreSQL Database

To create a project in a repository on PostgreSQL 7, 8 or 9, you work through the following stages:

- Stage 1: Create an empty database repository and set up the data tables
- Stage 2: Set up the PostgreSQL ODBC Driver
- Stage 3: Perform a project integrity check on the project file you are using as a base (optional, but recommended)
- Stage 4: Transfer the data
- Stage 5: Connect to the repository to open the project

#### Prerequisites

- PostgreSQL 7.3.2 or higher has been installed
- psqLODBC, version 7.03.01.00 or higher has been installed (do not use version 8.3.4 or 8.4.1)

#### Create Database Repository

Step	Action	See also
1	Create the empty database.	<a href="#">Create a PostgreSQL Repository</a> <sup>[229]</sup> <a href="#">Registered User's Scripts page</a> <a href="#">Trial User's Scripts Page</a>
2	Load the <b>PostgreSQL_EASchema.sql</b> file from the Sparx Systems website. For this you can work from the PostgreSQL (PSQL) command line, or with a tool such as <b>pgAdminIII</b> or EMS PostgreSQL Manager.	
3	Run the script to create all required data tables.	

#### Set Up the ODBC Driver

Step	Action	See also
1	Create a suitable ODBC Data Source to point to your new database.	<a href="#">Set up a PostgreSQL ODBC Driver</a> <sup>[230]</sup>

**Perform a Project Integrity Check**

Step	Action	See also
1	In Enterprise Architect, open the file-based project or template from which you are creating the project on the repository.	
2	Select and run <b>Tools   Data Management   Project Integrity Check</b> . This ensures your project data is 'clean' before being copied to the repository.	<a href="#">Project Integrity Check</a> [59]

**Transfer the project data to the repository**

Step	Action	See also
1	Open Enterprise Architect. (If the Open Project screen displays, click on the <b>Cancel</b> button to open with no project loaded.)	
2	Select the <b>Tools   Data Management   Project Transfer</b> menu option. The Project Transfer dialog displays.	
3	In the Transfer Type panel, select <b>.EAP to DBMS</b> .	
4	In the <b>Source Project</b> field, type the name of the project file to copy to the repository. If the .EAP file has Replication enabled, this must be removed before performing the transfer.	<a href="#">Remove Replication</a> [313]
5	At the right of the <b>Target Project</b> field, click on the ( ... ) (Browse) button. The Datalink Properties dialog displays.	
6	Select <b>Microsoft OLE DB Provider for ODBC Drivers</b> from the list. Click on the <b>Next</b> button.	
7	On the <b>Use Data Source Name</b> field, click on the drop-down arrow and select the ODBC Data Source you configured to point to your new database. Click on the <b>OK</b> button. The Project Transfer dialog redisplay.	<a href="#">Set up a PostgreSQL ODBC Driver</a> [230]

Step	Action	See also
8	If required, select the <b>Logfile</b> checkbox and type a path and filename for the data transfer log file.	
9	Click on the <b>Transfer</b> button to begin the data transfer process.	
10	When the process is complete, you have created a project on a PostgreSQL database and can now connect to it and open it from Enterprise Architect.	<a href="#">Connect to a PostgreSQL Data Repository</a> <sup>[233]</sup>

**Notes**

- During the transfer, if an error message displays reporting '*...nonstandard use of \ in a string literal...*', set the server variable in the *postgresql.conf* file to: *escape\_string\_warning = off*

**Learn more**

- [Perform a Project Data Transfer](#)<sup>[504]</sup>

**3.3.3.1 Create a PostgreSQL Repository**

Creating a PostgreSQL database is fully documented in the product information provided with your PostgreSQL installer.

Having created your PostgreSQL database, use the SQL script provided by Sparx Systems to create the required table definitions for Enterprise Architect. You can obtain the scripts from the Sparx Systems website, on the:

- Registered Corporate edition Resources page (Registered users)
- Corporate edition Resources page (Trial users)

**Third Party Tools**

If you are unfamiliar with PostgreSQL and DBMS systems in general, you might want to consider a suitable front end tool. One such tool is **pgAdminIII**. It provides a convenient graphical user interface to enable creation of databases, execution of scripts and restores.

After creating a PostgreSQL data repository in Enterprise Architect, you must set up the PostgreSQL ODBC driver.

**Notes**

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions
- Ensure that the collation is set to the alphabet you use, such as Latin or Cyrillic

**Learn more**

- [Registered User's Scripts page](#)

- [Trial User's Scripts Page](#)
- [pgAdminIII Download Site](#)
- [Set up a PostgreSQL ODBC Driver](#)<sup>[230]</sup>

### 3.3.3.2 Set up a PostgreSQL ODBC Driver

After you have created a repository on your PostgreSQL database, you set up the PostgreSQL ODBC driver so that you can connect to a project on that repository from Enterprise Architect.

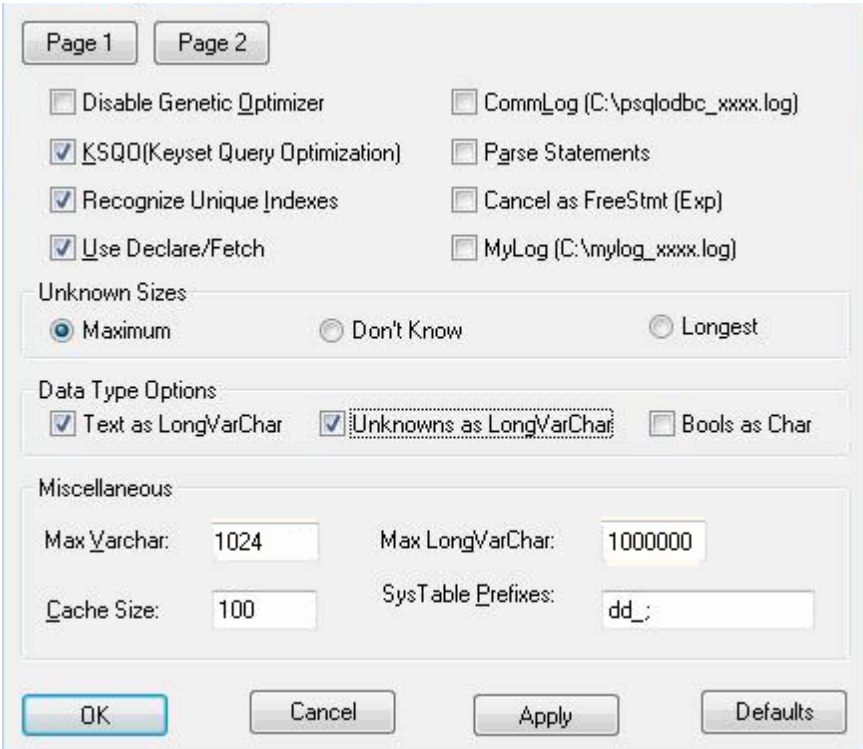
#### Prerequisites

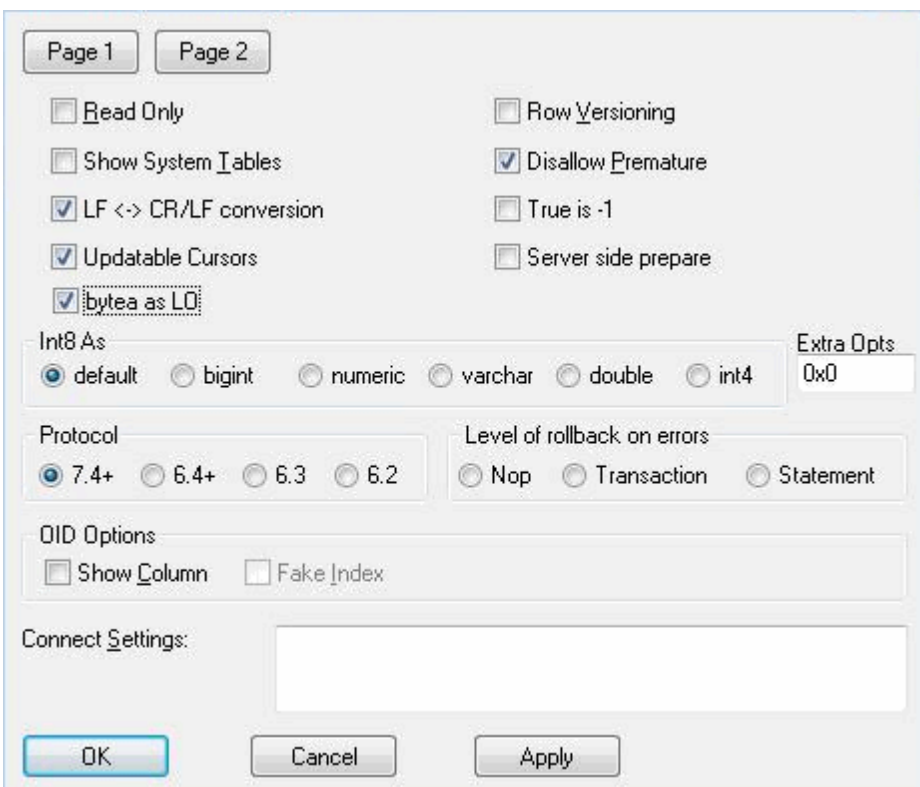
Install:

- PostgreSQL DBMS and repository
- PostgreSQL ODBC driver software version 7.03.01.00 or above (note that versions 8.3.4 and 8.4.1 of the PostgreSQL ODBC Driver are not supported)

#### Set up the ODBC driver

Step	Action
1	<p>If you are using a 32-bit operating system:</p> <ul style="list-style-type: none"> <li>• Select the Windows™ <b>Control Panel   Administrative Tools   Data Sources (ODBC)</b> option</li> </ul> <p>If you are using a 64-bit operating system:</p> <ul style="list-style-type: none"> <li>• Enterprise Architect requires 32-bit ODBC drivers to connect to a repository through ODBC; to set up the ODBC configuration on <b>64-bit clients</b>, run the 32-bit ODBC Data Source Administrator from <code>C:\Windows\SysWOW64\odbcad32.exe</code></li> <li>• You can quickly load the correct 32-bit ODBC Data Source Administrator by selecting <b>Tools   ODBC Data Sources</b> in the Enterprise Architect menu bar</li> </ul> <p>The ODBC Data Source Administrator window displays.</p>
2	<p>Click on the <b>Add</b> button.</p> <p>The Create New Data Source dialog displays, enabling you to add a new DSN.</p>
3	<p>Select <b>PostgreSQL UNICODE</b> from the list.</p>
4	<p>Click on the <b>Finish</b> button.</p> <p>The Postgre SQL Connector/ODBC dialog displays.</p>
5	<p>Enter the following configuration details:</p> <ul style="list-style-type: none"> <li>• A data source name for the connection</li> <li>• The actual name of the database</li> </ul>

Step	Action
	<ul style="list-style-type: none"> <li>• The host address of the DBMS server</li> <li>• User name</li> <li>• A description (optional)</li> <li>• The password</li> </ul>
6	<p>To set the advanced options, click on the <b>Datasource</b> button and set the options on Page 1 and Page 2 as shown below:</p> 

Step	Action
	
7	<p>If you are using PostgreSQL version 8+, on Page 2 select:</p> <ul style="list-style-type: none"> <li>• The <b>Disallow Premature</b> checkbox</li> <li>• In the Protocol panel, the <b>7.4+</b> radio button</li> </ul>
8	<p>Click on the <b>OK</b> button to complete the configuration.</p>

Your PostgreSQL driver is now available to connect to the repository from Enterprise Architect.

#### Learn more

- [Create a project in a PostgreSQL database](#)<sup>[227]</sup>
- [Create a PostgreSQL Repository](#)<sup>[229]</sup>
- [Connect To a PostgreSQL Data Repository](#)<sup>[233]</sup>



### 3.3.3.3 Connect to a PostgreSQL Data Repository

To access a project in your MySQL data repository, you need to connect to the data repository from Enterprise Architect.

#### Prerequisites

- The PostgreSQL repository and the project already exist
- You have **SELECT**, **UPDATE**, **INSERT** and **DELETE** access permissions
- The PostgreSQL ODBC driver has been set up

Access   **File | Open Project ( Ctrl+O )**

#### Connect to the repository

Step	Action
1	In the Open Project dialog, select the <b>Connect to Server</b> checkbox.
2	Click on the ( ... ) (Browse) button. The Data Link Properties dialog displays.
3	Select <b>Microsoft OLE DB Provider for ODBC Drivers</b> from the list.
4	Click on the <b>Next&gt;&gt;</b> button. The Connection tab displays.
5	Click on the <b>Use data source name</b> radio button and on the drop-down arrow in its field. From the list, select the ODBC driver you have set up to connect to your PostgreSQL repository.
6	Click on the <b>Test Connection</b> button to confirm that the details are correct.
7	If the test does not succeed, revise your settings. If the test succeeds, click on the <b>OK</b> button; the Connection Name & Type dialog displays.
8	Give the connection a suitable name so that you can recognize it in the Recent Projects panel on the Open Project dialog.
9	If required, select the <b>Encrypt Connection String</b> checkbox. This encrypts and hides the connection details of the database from the users that the connection

Step	Action
	string is given to.
10	<p>If required, select the <b>Lazy Load</b> checkbox to not load the full project view when the model is loaded; instead, only the parts that are necessary to display the visible portion of the tree are loaded.</p> <p>With this setting, the model loads faster and users can begin work sooner, but at the expense of later small delays as Enterprise Architect loads specific portions of the model.</p>
11	<p>If required, select the <b>Use WAN Optimization</b> checkbox.</p> <p>To improve performance over a Wide Area Network, remote database calls can be routed through a WAN Optimizer that compresses the data returned from the repository, reducing transfer time.</p> <p>If you select this checkbox, complete the next three fields (see your administrator for the correct values); otherwise go to step 15.</p>
12	In the <b>Server</b> field, type the network name or address of the optimizer server.
13	In the <b>Port</b> field, type the port on which the server is running on the remote machine.
14	In the <b>DSN</b> field, type the data source name of the database as it appears on the remote machine.
15	<p>Click on the <b>OK</b> button to complete the configuration and open the project.</p> <p>This also adds the project name to the <b>Recent Projects</b> list on the Start Page; from now on, you can open the project again just by clicking on this name.</p>

### Notes

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions

### Learn more

- [Create a project in a PostgreSQL database](#)<sup>[227]</sup>
- [Create a PostgreSQL Repository](#)<sup>[229]</sup>
- [Set up a PostgreSQL ODBC Driver](#)<sup>[230]</sup>
- [The WAN Optimizer](#)<sup>[259]</sup>
- [Open a Project](#)<sup>[202]</sup>

### 3.3.4 Create a Project in a Sybase ASA Database

To create a project in a repository on Sybase Adaptive Server Anywhere 8 or 9, or SQL Anywhere 10, 11 or 12, you work through the following stages:

- Stage 1: Create an empty database repository

- Stage 2: Set up the ASA ODBC Driver
- Stage 3: Perform a project integrity check on the .project file you are using as a base (optional, but recommended)
- Stage 4: Transfer the data
- Stage 5: Connect to the repository and open the project

### Prerequisites

- Install Adaptive Server Anywhere - SQL Anywhere Studio 8 or higher; this also installs the ASA ODBC driver

### Create Database Repository

Step	Action	See also
1	Create an empty ASA database.	<a href="#">Create an Adaptive Server Anywhere Repository</a> <sup>[237]</sup> <a href="#">Registered User's Scripts page</a> <a href="#">Trial User's Scripts Page</a>
2	Using (for example) Sybase Central, load the <b>ASA_EASchema.sql</b> file from the Sparx Systems website.	
3	Run the script to create all required data tables.	

### Set up the ASA ODBC Driver



Step	Action	See also
1	Create a suitable ODBC Data Source to point to your new database.	<a href="#">Set up an ASA ODBC Driver</a> <sup>[237]</sup>

### Perform Project Integrity Check

Step	Action	See also
1	In Enterprise Architect, open the file-based project or template from which you are creating the project on the repository.	<a href="#">Project Integrity Check</a> <sup>[597]</sup>
2	Select and run <b>Tools   Data Management   Project Integrity Check</b> . This ensures your project data is 'clean' before being copied to the repository.	

Step	Action	See also

### Transfer the Project Data

Step	Action	See also
1	Open Enterprise Architect. (If the Open Project screen displays, click on the <b>Cancel</b> button to open with no project loaded.)	
2	Select the <b>Tools   Data Management   Project Transfer</b> menu option. The Project Transfer dialog displays.	
3	In the Transfer Type panel, select <b>.EAP to DBMS</b> .	
4	In the <b>Source Project</b> field, type the name of the project file to copy to the repository.  If the .EAP file has Replication enabled, this must be removed before performing the transfer.	<a href="#">Remove Replication</a> 
5	At the right of the <b>Target Project</b> field, click on the ( ... ) (Browse) button.  The Datalink Properties dialog displays.	
6	Select <b>Microsoft OLE DB Provider for ODBC Drivers</b> from the list.  Click on the <b>Next</b> button.	
7	In the <b>Use Data source name</b> field, click on the drop-down arrow and select the ODBC Data Source you configured to point to your new database.  Click on the <b>OK</b> button. The Project Transfer dialog redisplay.	<a href="#">Set up an ASA ODBC Driver</a> 
8	If required, select the <b>Logfile</b> checkbox and type a path and filename for the data transfer log file.	
9	Click on the <b>Transfer</b> button to begin the data transfer process.	
10	When the process is complete, you have created a project on an ASA	<a href="#">Connect to an ASA Data</a>

Step	Action	See also
	database and can now connect to it and open it from Enterprise Architect.	<a href="#">Repository</a> <sup>[239]</sup>

**Learn more**

- [Perform a Project Data Transfer](#)<sup>[504]</sup>

**3.3.4.1 Create an Adaptive Server Anywhere Repository**

Creating an ASA database is fully documented in the product information provided with your ASA installer.

Having created your ASA database, use the SQL script (**ASA\_EASchema.sql**) provided by Sparx Systems to create the required table definitions for Enterprise Architect. You can obtain the scripts from the Sparx Systems website, on the:

- Registered Corporate edition Resources page (Registered users)
- Corporate edition Resources page (Trial users)

**Third Party Tools**

If you are unfamiliar with ASA and DBMS systems in general, you might want to consider a suitable front end tool. Sybase Central is one such tool, that can be installed along with the DBMS. It provides a convenient graphical user interface to enable creation of databases, execution of scripts and restores.

After creating an ASA data repository in Enterprise Architect, you must set up the ASA ODBC driver.

**Notes**

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions
- Ensure that the collation is set to the alphabet you use, such as Latin or Cyrillic

**Learn more**

- [Create a project in a Sybase ASA database](#)<sup>[234]</sup>
- [Registered User's Scripts page](#)
- [Trial User's Scripts Page](#)
- [Setup an Adaptive Server Anywhere ODBC Driver](#)<sup>[237]</sup>

**3.3.4.2 Set up an ASA ODBC Driver**

After you have created a repository on your ASA database, you set up the ASA ODBC driver so that you can connect to a project on that repository from Enterprise Architect.

**Prerequisites**

Install:

- Adaptive Server Anywhere - SQL Anywhere Studio 8 or higher, and create a repository

- ASA ODBC driver software (installed with the ASA DBMS)

### Set up the ODBC Driver

Step	Action
1	<p>If you are using a 32-bit operating system:</p> <ul style="list-style-type: none"> <li>• Select the Windows™ <b>Control Panel   Administrative Tools   Data Sources (ODBC)</b> option</li> </ul> <p>If you are using a 64-bit operating system:</p> <ul style="list-style-type: none"> <li>• Enterprise Architect requires 32-bit ODBC drivers to connect to a repository through ODBC; to set up the ODBC configuration on <b>64-bit clients</b>, run the 32-bit ODBC Data Source Administrator from <code>C:\Windows\SysWOW64\odbcad32.exe</code></li> <li>• You can quickly load the correct 32-bit ODBC Data Source Administrator by selecting <b>Tools   ODBC Data Sources</b> in the Enterprise Architect menu bar</li> </ul> <p>The ODBC Data Source Administrator window displays.</p>
2	<p>Click on the <b>Add</b> button.</p> <p>The Create New Data Source dialog displays, enabling you to add a new DSN.</p>
3	<p>Select <b>Adaptive Server Anywhere</b> or <b>SQL Anywhere</b> from the list.</p>
4	<p>Click on the <b>Finish</b> button.</p> <p>The ASA Connector/ODBC dialog displays.</p>
5	<p>Enter the following configuration details:</p> <ul style="list-style-type: none"> <li>• A data source name for the connection, on the ODBC tab</li> <li>• User name and password on the Login tab (dba, sql are the defaults when ASA is installed)</li> <li>• The server name and the path to the database, on the Database tab</li> <li>• The network protocol on the Network tab (if the database is on a network location)</li> </ul>
6	<p>Return to the ODBC tab and click on the <b>Test Connection</b> button to confirm that the details are correct.</p>
7	<p>If the test succeeds, click on the <b>OK</b> button to complete the configuration.</p> <p>If the test does not succeed, review your settings.</p>

Your ASA driver is now available to connect to the repository from Enterprise Architect.

Learn more

- [Create a project in a Sybase ASA database](#)<sup>[234]</sup>
- [Create an Adaptive Server Anywhere Repository](#)<sup>[237]</sup>
- [Connect to an ASA Data Repository](#)<sup>[239]</sup>

**3.3.4.3 Connect to an ASA Data Repository**

To access a project in your Adaptive Server Anywhere (ASA) data repository, you need to connect to the data repository from Enterprise Architect.

Prerequisites

- The ASA repository and the project already exist
- You have **SELECT**, **UPDATE**, **INSERT** and **DELETE** access permissions
- The ASA ODBC driver has been set up

Access   **File | Open Project (Ctrl+O)**

Connect to the repository

Step	Action
1	In the Open Project dialog, select the <b>Connect to Server</b> checkbox.
2	Click on the ( ... ) (Browse) button. The Data Link Properties dialog displays.
3	Select <b>Microsoft OLE DB Provider for ODBC Drivers</b> from the list.
4	Click on the <b>Next&gt;&gt;</b> button. The Connection tab displays.
5	Click on the <b>Use data source name</b> radio button and on the drop-down arrow in its field. From the list, select the ODBC driver you have set up to connect to your ASA repository.
6	Click on the <b>Test Connection</b> button to confirm that the details are correct.
7	If the test does not succeed, revise your settings. If the test succeeds, click on the <b>OK</b> button; the Connection Name & Type dialog displays.

Step	Action
8	Give the connection a suitable name so that you can recognize it in the Recent Projects panel on the Open Project dialog.
9	If required, select the <b>Encrypt Connection String</b> checkbox.  This encrypts and hides the connection details of the database from the users that the connection string is given to.
10	If required, select the <b>Lazy Load</b> checkbox to not load the full project view when the model is loaded; instead, only the parts that are necessary to display the visible portion of the tree are loaded.  With this setting, the model loads faster and users can begin work sooner, but at the expense of later small delays as Enterprise Architect loads specific portions of the model.
11	If required, select the <b>Use WAN Optimization</b> checkbox.  To improve performance over a Wide Area Network, remote database calls can be routed through a WAN Optimizer that compresses the data returned from the repository, reducing transfer time.  If you select this checkbox, complete the next three fields (see your administrator for the correct values); otherwise go to step 15.
12	In the <b>Server</b> field, type the network name or address of the optimizer server.
13	In the <b>Port</b> field, type the port on which the server is running on the remote machine.
14	In the <b>DSN</b> field, type the data source name of the database as it appears on the remote machine.
15	Click on the <b>OK</b> button to complete the configuration and open the project.  This also adds the project name to the <b>Recent Projects</b> list on the Start Page; from now on, you can open the project again just by clicking on this name.

### Notes

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions

### Learn more

- [Create a project in a Sybase ASA database](#)<sup>[234]</sup>
- [Create an Adaptive Server Anywhere Repository](#)<sup>[237]</sup>
- [Set up an ASA ODBC Driver](#)<sup>[237]</sup>
- [The WAN Optimizer](#)<sup>[259]</sup>
- [Open a Project](#)<sup>[202]</sup>



### 3.3.5 Create a Project in an Access 2007 Database

In setting up a project in an Access 2007 repository, you could be working in one of two situations:

- You are able to work in Access 2007 itself, in which case you can **convert** your .eap project directly to an Access 2007 .accdb database file, which you can connect to and open in Enterprise Architect
- You have access to an empty Access 2007 .accdb file, and need to **transfer** a .eap project into that .accdb file

In either case, before you convert or transfer the project data, you could perform a project integrity check on the .eap file you are using as a base (optional, but recommended).

#### Perform a Project Integrity Check

Step	Action	See also
1	Open the base project or template .eap file.	
2	Select and run <b>Tools   Data Management   Project Integrity Check</b> . This ensures your project data is 'clean' before uploading.	<a href="#">Project Integrity Check</a> <sup>[59]</sup>

#### Convert a .eap file in Access 2007

Step	Action	See also
1	Open Access 2007 and open the source .EAP file within this product. Allow Access to convert the .eap file to a .accdb file. This forms the Access 2007 repository.	
2	Open Enterprise Architect and open the .accdb file containing the project, browsing for it using the Open project dialog.	<a href="#">Open a Project</a> <sup>[20]</sup>

#### Transfer the project data into an empty .accdb file

Step	Action	See also
1	Open Enterprise Architect. (If the Open Project screen displays, click on the <b>Cancel</b> button to open with no project loaded.)	

Step	Action	See also
2	Select the <b>Tools   Data Management   Project Transfer</b> menu option. The Project Transfer dialog displays.	
3	In the Transfer Type panel, select <b>.EAP to DBMS</b> .	
4	In the <b>Source Project</b> field, type the name of the .EAP file to copy to the repository. If the .EAP file has Replication enabled, this must be removed before performing the transfer.	<a href="#">Remove Replication</a> <sup>[313]</sup>
5	At the right of the <b>Target Project</b> field, click on the ( ... ) (Browse) button. The Datalink Properties dialog displays.	
6	Select <b>Microsoft Office 12.0 Access Database Engine OLE DB Provider</b> from the list. Click on the <b>Next</b> button.	
7	On the Data Source Details page of the Datalink Properties dialog, type in the full path to the Access 2007 .accdb file. Click on the <b>OK</b> button to return to the Project Transfer dialog.	
8	If required, select the <b>Logfile</b> checkbox and type a path and filename for the data transfer log file.	
9	Click on the <b>Transfer</b> button to begin the data transfer process.  When the process is complete, you have created your project in an Access 2007 database and can now open it directly from Enterprise Architect, browsing for the .accdb file location in the Open Project dialog.	<a href="#">Open a Project</a> <sup>[202]</sup>

### Notes

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions
- If you do not have Access 2007, you can download the Access Database Engine from the [Microsoft downloads site](#)
- Ensure that the collation is set to the alphabet you use, such as Latin or Cyrillic

### Learn more

- [Perform a Project Data Transfer](#)<sup>[504]</sup>

### 3.3.6 Create a Project in a Progress OpenEdge Database

To create a project in a repository on Progress OpenEdge, you work through the following stages:

- Stage 1: Create an empty database repository and set up the data tables
- Stage 2: Set up the Progress OpenEdge ODBC Driver
- Stage 3: Perform a project integrity check on the project file you are using as a base (optional, but recommended)
- Stage 4: Transfer the data
- Stage 5: Connect to the repository to open the project

#### Prerequisites

- Install OpenEdge, version 10.1C
- Install DataDirect ODBC driver for OpenEdge (version 4.20 or later)
- Obtain access permissions to create a new database

#### Create Database Repository

Step	Action	See also
1	Create an empty OpenEdge database for the new repository.	<a href="#">Create a Progress OpenEdge Repository</a> <sup>[245]</sup> <a href="#">Registered User's Scripts page</a> <a href="#">Trial User's Scripts Page</a>
2	Download the <b>OpenEdge_EASchema.sql</b> file from the Sparx Systems website.	
3	Run the script to create all required data tables.	

#### Set up the ODBC Driver

Step	Action	See also
1	Create a suitable ODBC Data Source to point to your new database.	<a href="#">Set Up a Progress OpenEdge ODBC Driver</a> <sup>[247]</sup>

#### Perform Project Integrity Check

Step	Action	See also
1	In Enterprise Architect, open the file-based project or template from	

Step	Action	See also
	which you are creating the project on the repository.	
2	Select and run <b>Tools   Data Management   Project Integrity Check</b> . This ensures your project data is 'clean' before uploading.	<a href="#">Project Integrity Check</a> <sup>[597]</sup>

### Transfer the Project Data

Step	Action	See also
1	Open Enterprise Architect. (If the Open Project screen displays, click on the <b>Cancel</b> button to open with no project loaded.)	
2	Select the <b>Tools   Data Management   Project Transfer</b> menu option. The Project Transfer dialog displays.	
3	In the Transfer Type panel, select <b>.EAP to DBMS</b> .	
4	In the <b>Source Project</b> field, type the name of the project file to copy to Progress OpenEdge. If the .EAP file has Replication enabled, this must be removed before performing the transfer.	<a href="#">Remove Replication</a> <sup>[313]</sup>
5	At the right of the <b>Target Project</b> field, click on the ( ... ) (Browse) button. The Datalink Properties dialog displays.	
6	Select <b>Microsoft OLE DB Provider for ODBC Drivers</b> from the list. Click on the <b>Next</b> button.	
7	In the <b>Use Data source name</b> field, click on the drop-down arrow and select the ODBC Data Source you configured to point to your new database. Click on the <b>OK</b> button. The Project Transfer dialog redisplay.	<a href="#">Set up a Progress OpenEdge ODBC Driver</a> <sup>[247]</sup>
8	If required, select the <b>Logfile</b> checkbox and type a path and filename for the data transfer log file.	

Step	Action	See also
9	Click on the <b>Transfer</b> button to begin the data transfer process.	
10	When the process is complete, you have created a project on a Progress OpenEdge database and can now connect to it and open it from Enterprise Architect.	<a href="#">Connect to a Progress OpenEdge Repository</a> <sup>[248]</sup>

#### Learn more

- [Perform a Project Data Transfer](#)<sup>[504]</sup>

### 3.3.6.1 Create a Progress OpenEdge Repository

Creating a Progress OpenEdge database is fully documented in the product information provided with your OpenEdge installer.

Having created your OpenEdge database, use the SQL script (**OpenEdge\_EASchema.sql**) provided by Sparx Systems to create the required table definitions for Enterprise Architect. You can obtain the script from the Sparx Systems website, on the:

- Registered Corporate edition Resources page (Registered users)
- Corporate edition Resources page (Trial users)

#### Create the repository

Step	Action
1	From the Windows Start menu, run proenv in OpenEdge; that is: <b>Start   Programs   OpenEdge   proenv</b> .
2	Create the database: <code>prodb &lt;database_name&gt; empty</code>
3	Start the database server: <code>proserve &lt;database_name&gt; -S &lt;port_number&gt;</code>
4	Open Data Administration to add users:  <code>prowin32 -db &lt;database_name&gt; -S &lt;port_number&gt; -p _admin -rx</code>  The Database Administration dialog displays.
5	Select <b>Admin   Security   Edit User List</b> and set up one or more user IDs from the Edit User List dialog, as follows:

Step	Action
	<ol style="list-style-type: none"> <li>1. Click on the <b>Add</b> button; the Add User dialog displays.</li> <li>2. Complete the <b>User ID</b>, <b>User Name</b> and <b>Password</b> fields, and click on the <b>OK</b> button; the Password Verification dialog displays.</li> <li>3. Verify the password and click on the <b>OK</b> button; the Edit User List dialog redisplay.</li> <li>4. Repeat the process for the next user or, if finished, click on the <b>OK</b> button; the Database Administration dialog redisplay.</li> </ol>
6	Close Database Administration ( <b>Database   Exit</b> ).
7	<p>On the proenv command prompt, type:</p> <pre>sql exp -user &lt;user id&gt; -password &lt;password&gt; -db &lt;database_name&gt; -S &lt;port_number&gt;</pre> <p>The SQL Explorer &gt; prompt displays.</p>
8	<p>Grant privileges to the administrator user:</p> <pre>SQL Explorer&gt;grant dba, resource to oe_user; SQL Explorer&gt;commit; SQL Explorer&gt;exit</pre>
9	<p>Execute the <b>OpenEdge_EASchema.sql</b> script, with the following command:</p> <pre>sql exp -user oe_user -password sparx -db &lt;path\to\database&gt; -S &lt;port_number&gt; -infile &lt;path\to\OpenEdge_EASchema.sql&gt;</pre>

After creating a Progress OpenEdge data repository in Enterprise Architect, you must set up the Progress OpenEdge ODBC driver.

### Notes

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions
- Ensure that the collation is set to the alphabet you use, such as Latin or Cyrillic

### Learn more

- [Create a project in a Progress OpenEdge database](#)<sup>[243]</sup>
- [Registered User's Scripts page](#)
- [Trial User's Scripts Page](#)
- [Set up a Progress OpenEdge ODBC Driver](#)<sup>[247]</sup>

- [Connect to a Progress OpenEdge Repository](#)<sup>[248]</sup>

### 3.3.6.2 Set up a Progress OpenEdge ODBC Driver

After you have created a repository on your Progress OpenEdge database, you set up the Progress OpenEdge ODBC driver so that you can connect to a project on that repository from Enterprise Architect.

#### Prerequisites

Install:

- OpenEdge DBMS (10.1C) and repository
- DataDirect ODBC driver for OpenEdge software (version 4.20 or later)

#### Set up the ODBC driver

Step	Action
1	<p>If you are using a 32-bit operating system:</p> <ul style="list-style-type: none"> <li>• Select the Windows™ <b>Control Panel   Administrative Tools   Data Sources (ODBC)</b> option</li> </ul> <p>If you are using a 64-bit operating system:</p> <ul style="list-style-type: none"> <li>• Enterprise Architect requires 32-bit ODBC drivers to connect to a repository through ODBC; to set up the ODBC configuration on <b>64-bit clients</b>, run the 32-bit ODBC Data Source Administrator from <code>C:\Windows\SysWOW64\odbcad32.exe</code></li> <li>• You can quickly load the correct 32-bit ODBC Data Source Administrator by selecting <b>Tools   ODBC Data Sources</b> in the Enterprise Architect menu bar</li> </ul> <p>The ODBC Data Source Administrator window displays.</p>
2	<p>Click on the <b>Add</b> button.</p> <p>The Create New Data Source dialog displays, enabling you to add a new DSN.</p>
3	<p>Select <b>DataDirect/OpenEdge SQL Driver</b> from the list.</p>
4	<p>Click on the <b>Finish</b> button.</p> <p>The DSN Configuration dialog displays.</p>
5	<p>Enter the following configuration details:</p> <ul style="list-style-type: none"> <li>• A data source name for the connection</li> <li>• A description (optional)</li> <li>• The host name and port number of the DBMS server</li> <li>• The database name on the selected server</li> <li>• The User ID</li> </ul>

Step	Action
6	Click on the <b>Test Connect</b> button to confirm that the details are correct.
7	If the test succeeds, click on the <b>OK</b> button to complete the configuration. If the test does not succeed, review your settings.

Your OpenEdge connection is now available to connect to the repository from Enterprise Architect.

#### Learn more

- [Create a project in a Progress OpenEdge database](#)<sup>[243]</sup>
- [Create a Progress OpenEdge Repository](#)<sup>[245]</sup>
- [Connect to a Progress OpenEdge Repository](#)<sup>[248]</sup>

### 3.3.6.3 Connect to a Progress OpenEdge Data Repository

To access a project in your Progress OpenEdge data repository, you need to connect to the data repository from Enterprise Architect.

#### Prerequisites

- The Progress OpenEdge repository and the project already exist
- You have **SELECT**, **UPDATE**, **INSERT** and **DELETE** access permissions
- The Progress OpenEdge ODBC driver has been set up

Access   **File | Open Project (Ctrl+O)**

#### Connect to the repository

Step	Action
1	In the Open Project dialog, select the <b>Connect to Server</b> checkbox.
2	Click on the ( ... ) (Browse) button. The Data Link Properties dialog displays.
3	Select <b>Microsoft OLE DB Provider for ODBC Drivers</b> from the list.



Step	Action
4	Click on the <b>Next&gt;&gt;</b> button. The Connection tab displays.
5	Click on the <b>Use data source name</b> radio button and on the drop-down arrow in its field. From the list, select the ODBC driver you have set up to connect to your Progress OpenEdge repository.
6	Enter the <b>User Name</b> and <b>Password</b> .
7	Enter the <b>Initial catalog</b> .
8	Click on the All tab and double-click on <b>Extended Properties</b> .
9	Click on the Connection tab again.
10	In the <b>Property Value</b> field, edit the value to: <b>DefaultSchema=PUB</b> . Click on the <b>Test Connection</b> button to confirm that the details are correct.
11	If the test does not succeed, revise your settings. If the test succeeds, click on the <b>OK</b> button; the Logon to Progress dialog displays.
12	Check the details to ensure that they are correct. Click on the <b>OK</b> button; the Connection Name & Type dialog displays.
13	Give the connection a suitable name so that you can recognize it in the Recent Projects panel on the Open Project dialog.
14	If required, select the <b>Encrypt Connection String</b> checkbox. This encrypts and hides the connection details of the database from the users that the connection string is given to.
15	If required, select the <b>Lazy Load</b> checkbox to not load the full project view when the model is loaded; instead, only the parts that are necessary to display the visible portion of the tree are loaded. With this setting, the model loads faster and users can begin work sooner, but at the expense of later small delays as Enterprise Architect loads specific portions of the model.

Step	Action
16	<p>If required, select the <b>Use WAN Optimization</b> checkbox.</p> <p>To improve performance over a Wide Area Network, remote database calls can be routed through a WAN Optimizer that compresses the data returned from the repository, reducing transfer time.</p> <p>If you select this checkbox, complete the next three fields (see your administrator for the correct values); otherwise go to step 20.</p>
17	In the <b>Server</b> field, type the network name or address of the optimizer server.
18	In the <b>Port</b> field, type the port on which the server is running on the remote machine.
19	In the <b>DSN</b> field, type the data source name of the database as it appears on the remote machine.
20	<p>Click on the <b>OK</b> button to complete the configuration and open the project.</p> <p>This also adds the project name to the <b>Recent Projects</b> list on the Start Page; from now on, you can open the project again just by clicking on this name.</p>

#### Notes

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions

#### Learn more

- [Create a project in a Progress OpenEdge database](#)<sup>[243]</sup>
- [Create a Progress OpenEdge Repository](#)<sup>[245]</sup>
- [Set up a Progress OpenEdge ODBC Driver](#)<sup>[247]</sup>
- [The WAN Optimizer](#)<sup>[259]</sup>
- [Open a Project](#)<sup>[202]</sup>

### 3.3.7 Create a Project in an Oracle Database

To create a project in a repository on Oracle 9i, 10g or 11g, you work through the following stages:

- Stage 1: Create an empty database repository and set up the data tables
- Stage 2: Set up the ODBC Driver (if required; you can connect to an Oracle repository using either OLE DB or an ODBC driver; if you intend to connect through the ODBC driver, you must first set it up)
- Stage 3: Perform a project integrity check on the project file you are using as a base (optional, but recommended)
- Stage 4: Transfer the data
- Stage 5: Connect to the repository to open the project

**Prerequisites**

- Oracle 9i, 10g or 11 g has been installed

**Create Database Repository**

Step	Action	See also
1	Create the empty database.	<a href="#">Create an Oracle Data Repository</a> <sup>[253]</sup>
2	Using a tool such as Oracle SQL *Plus, load the <b>Oracle_EASchema.sql</b> file.	<a href="#">Registered User's Scripts page</a> <a href="#">Trial User's Scripts Page</a>
3	Run the script to create all required data structures.	




**Set Up the ODBC Driver**

Step	Action	See also
1	Create a suitable ODBC Data Source to point to your new database.	<a href="#">Set up an Oracle ODBC Driver</a> <sup>[254]</sup>

**Perform Project Integrity Check**

Step	Action	See also
1	Open the file-based project or template.	
2	Select and run <b>Tools   Data Management   Project Integrity Check</b> . This ensures your project data is 'clean' before uploading.	<a href="#">Project Integrity Check</a> <sup>[597]</sup>

**Transfer the project data to the repository**

Step	Action	See also
1	Open Enterprise Architect. (If the Open Project screen displays, click on the <b>Cancel</b> button to open with no project loaded.)	
2	Select the <b>Tools   Data Management   Project Transfer</b> menu option. The Project Transfer dialog displays.	
3	In the Transfer Type panel, select <b>.EAP to DBMS</b> .	
4	In the <b>Source Project</b> field, type the name of the project file to copy to Oracle. If the .EAP file has Replication enabled, this must be removed before performing the transfer.	<a href="#">Remove Replication</a> 
5	At the right of the <b>Target Project</b> field, click on the ( ... ) (Browse) button. The Datalink Properties dialog displays.	
6	Select <b>Oracle Provider for OLE DB</b> from the list. Click on the <b>Next</b> button.	
7	On the Connection page of the Datalink Properties dialog, enter the Oracle service name in the <b>Data Source</b> field, and the user name and password as required. Click on the <b>OK</b> button. The Project Transfer dialog redisplay.	
8	If required, select the <b>Logfile</b> checkbox and type a path and filename for the data transfer log file.	
9	Click on the <b>Transfer</b> button to begin the data transfer process.	
10	When the process is complete, you have created a project on a Oracle database and can now connect to it and open it from Enterprise Architect.	<a href="#">Connect to an Oracle Data Repository (ODBC)</a>  <a href="#">Connect to an Oracle Data Repository (OLE DB)</a> 

### **Notes**

- When transferring a project to Oracle you must have access rights to execute the **CREATE SEQUENCE** command

### **Learn more**

- [Perform a Project Data Transfer](#)<sup>[504]</sup>

#### **3.3.7.1 Create an Oracle Data Repository**

Creating an Oracle database is fully documented in the product information provided with your Oracle installer.

Before creating an Oracle data repository, install the appropriate version of Oracle (9i, 10g or 11g) and MDAC 2.6 or higher, and obtain access permission to create a new database.

Having created your Oracle database, use the SQL script (**Oracle\_EASchema.sql**) provided by Sparx Systems to create the required table definitions and indexes for Enterprise Architect. You can obtain the script from the Sparx Systems website, on the:

- Registered Corporate edition Resources page (Registered users), or
- Corporate edition Resources page (Trial users)

When you eventually connect to the Oracle database from Enterprise Architect, you can do so using either OLE DB or ODBC; if you intent to use ODBC, after creating the Oracle data repository you set up the Oracle ODBC driver.

### **Third Party Tools**

If you are unfamiliar with Oracle and DBMS systems in general, you might want to consider a suitable front end tool. You could connect to the database and execute the scripts with a program such as Oracle SQL\*Plus or SQL Plus Worksheet.

### **Notes**

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions
- With Windows 7 the OLE/ODBC connection is set using the 32 bit ODBC driver, installed using C: \Windows\SysWOW64\odbcad32.exe; the connection can be set using either ODBC or OLE
- If you intend to connect via the OLE connection, download the 32-bit ODBC driver from <http://www.oracle.com/technetwork/topics/dotnet/utilsoft-086879.html>
- Set the collation to the alphabet you use, such as Latin or Cyrillic

### **Learn more**

- [Create a project in an Oracle database](#)<sup>[250]</sup>
- [Registered User's Scripts page](#)
- [Trial User's Scripts Page](#)
- [Set up an Oracle ODBC Driver](#)<sup>[254]</sup>
- [Connect to an Oracle Data Repository \(ODBC\)](#)<sup>[255]</sup>

- [Connect to an Oracle Data Repository \(OLE DB\)](#) <sup>[257]</sup>

### 3.3.7.2 Set up an Oracle ODBC Driver

After you have created a repository on your Oracle database, you set up the Oracle ODBC driver so that you can connect to a project on that repository from Enterprise Architect (in preference to using an OLE DB connection).

#### Prerequisites

Install:

- Oracle DBMS and repository

#### Set up the Oracle ODBC driver

Step	Action
1	<p>If you are using a 32-bit operating system:</p> <ul style="list-style-type: none"> <li>• Select the Windows™ <b>Control Panel   Administrative Tools   Data Sources (ODBC)</b> option</li> </ul> <p>If you are using a 64-bit operating system:</p> <ul style="list-style-type: none"> <li>• Enterprise Architect requires 32-bit ODBC drivers to connect to a repository through ODBC; to set up the ODBC configuration on <b>64-bit clients</b>, run the 32-bit ODBC Data Source Administrator from <code>C:\Windows\SysWOW64\odbcad32.exe</code></li> <li>• You can quickly load the correct 32-bit ODBC Data Source Administrator by selecting <b>Tools   ODBC Data Sources</b> in the Enterprise Architect menu bar</li> </ul> <p>The ODBC Data Source Administrator window displays.</p>
2	<p>Click on the <b>Add</b> button.</p> <p>The Create New Data Source dialog displays, enabling you to add a new DSN.</p>
3	<p>Select <b>Oracle in OraDB11g_home1</b> from the list (or similar, depending on the ODBC installation).</p>
4	<p>Click on the <b>Finish</b> button.</p> <p>The Oracle ODBC Driver Configuration dialog displays.</p>
5	<p>Enter the following configuration details:</p> <ul style="list-style-type: none"> <li>• A data source name for the connection</li> <li>• A description (optional)</li> <li>• The TNS Service Name (click on the drop down arrow and select from the list)</li> <li>• User ID</li> </ul>
6	<p>Click on the <b>Test Connection</b> button and enter the Oracle user password to confirm that the details</p>

Step	Action
	are correct.
<b>7</b>	If the test succeeds, click on the <b>OK</b> button to complete the configuration. If the test does not succeed, review your settings.

Your Oracle driver is now available to connect to the repository from Enterprise Architect.

#### Learn more

- [Create a Project in an Oracle Database](#)<sup>[250]</sup>
- [Create an Oracle Data Repository](#)<sup>[253]</sup>
- [Connect to an Oracle Data Repository \(ODBC\)](#)<sup>[255]</sup>

### **3.3.7.3 Connect to an Oracle Data Repository (ODBC)**

To access a project in your Oracle 9i, 10g or 11g data repository, you need to connect to the data repository from Enterprise Architect.

#### Prerequisites

- The Oracle repository and the project already exist
- You have **SELECT**, **UPDATE**, **INSERT** and **DELETE** access permissions
- The Oracle ODBC driver has been set up

Alternatively, you can connect to your Oracle project using OLE DB - see *Connect to an Oracle Data Repository (OLE DB)*.

**Access**   **File | Open Project (Ctrl+O)**

#### Connect to an Oracle Data Repository using an ODBC Driver

Step	Action
<b>1</b>	In the Open Project dialog, select the <b>Connect to Server</b> checkbox.
<b>2</b>	Click on the ( ... ) (Browse) button. The Data Link Properties dialog displays.
<b>3</b>	Select <b>Microsoft OLE DB Provider for ODBC Drivers</b> from the list.

Step	Action
4	Click on the <b>Next&gt;&gt;</b> button. The Connection tab displays.
5	In the <b>Data source</b> field, click on the drop-down arrow and select the data source name.
6	Type in the <b>User</b> name and <b>Password</b> .
7	Click on the <b>Test Connection</b> button to confirm that the details are correct.
8	If the test does not succeed, revise your settings. If the test succeeds, click on the <b>OK</b> button; Oracle prompts you for the password.
9	Type in the password. The Connection Name and Type dialog displays.
10	Give the connection a suitable name so that you can recognize it in the Recent Projects panel on the Open Project dialog.
11	If you want to encrypt and hide the connection details of the database from the users that the connection string is given to, select the <b>Encrypt Connection String</b> checkbox.
12	The Lazy Load facility does not load the full project view when the model is loaded; instead, it loads only the parts that are necessary to display the visible portion of the tree. With this set, the model loads faster and users can begin work sooner, but at the expense of later small delays as Enterprise Architect loads specific portions of the model.  If you want to use the Lazy load facility, select the <b>Lazy Load</b> checkbox.
13	To improve performance over a Wide Area Network, remote database calls can be routed through a WAN Optimizer that compresses the data returned from the repository, reducing transfer time. If you are using a WAN, and want to apply this facility, select the <b>Use WAN Optimization</b> checkbox; otherwise go to step 16.  If you select this checkbox, complete the next two fields (see your administrator for the correct values).
14	In the <b>Server</b> field, type the network name or address of the optimizer server.
15	In the <b>Port</b> field, type the port on which the server is running on the remote machine.



Step	Action
16	<p>Click on the <b>OK</b> button to complete the configuration and open the project.</p> <p>This also adds the project name to the <b>Recent Projects</b> list on the Start Page; from now on, you can open the project again just by clicking on this name.</p>

### Notes

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions

### Learn more

- [Create a Project in an Oracle Database](#)<sup>[250]</sup>
- [Create an Oracle Data Repository](#)<sup>[253]</sup>
- [Set up an Oracle ODBC Driver](#)<sup>[254]</sup>
- [Open a Project](#)<sup>[202]</sup>
- [The WAN Optimizer](#)<sup>[259]</sup>
- [Connect to an Oracle Data Repository \(OLE DB\)](#)<sup>[257]</sup>

### 3.3.7.4 Connect to an Oracle Data Repository (OLE DB)

To access a project in your Oracle 9i, 10g or 11g data repository, you need to connect to the data repository from Enterprise Architect.

### Prerequisites

- The Oracle repository and the project already exist
- You have **SELECT**, **UPDATE**, **INSERT** and **DELETE** access permissions

Alternatively, you can connect to the repository using an Oracle ODBC driver - see *Connect to an Oracle Data Repository (ODBC)*.

**Access** **File | Open Project (Ctrl+O)**

### Connect to an Oracle Repository using OLE DB

Step	Action
1	In the Open Project dialog, select the <b>Connect to Server</b> checkbox.
2	<p>Click on the ( ... ) (Browse) button.</p> <p>The Data Link Properties dialog displays.</p>

Step	Action
3	<p>Select <b>Oracle Provider for OLE DB</b> from the list.</p> <p>Do not select <b>Microsoft OLE DB Provider for Oracle</b>; Enterprise Architect might not work as expected.</p>
4	<p>Click on the <b>Next&gt;&gt;</b> button.</p> <p>The Connection tab displays.</p>
5	In the <b>Data source</b> field, click on the drop-down arrow and select the data source name (the service name of the Oracle database).
6	Type in the <b>User</b> name and <b>Password</b> .
7	Click on the <b>Test Connection</b> button to confirm that the details are correct.
8	<p>If the test does not succeed, revise your settings.</p> <p>If the test succeeds, click on the <b>OK</b> button; the Connection Name and Type dialog displays.</p>
9	Give the connection a suitable name so that you can recognize it in the Recent Projects panel on the Open Project dialog.
10	If you want to encrypt and hide the connection details of the database from the users that the connection string is given to, select the <b>Encrypt Connection String</b> checkbox.
11	<p>If required, select the <b>Lazy Load</b> checkbox to not load the full project view when the model is loaded; instead, only the parts that are necessary to display the visible portion of the tree are loaded.</p> <p>With this setting, the model loads faster and users can begin work sooner, but at the expense of later small delays as Enterprise Architect loads specific portions of the model.</p>
12	<p>To improve performance over a Wide Area Network, remote database calls can be routed through a WAN Optimizer that compresses the data returned from the repository, reducing transfer time. If you are using a WAN, and want to apply this facility, select the <b>Use WAN Optimization</b> checkbox; otherwise go to step 15.</p> <p>If you select this checkbox, complete the next two fields (see your administrator for the correct values).</p>
13	In the <b>Server</b> field, type the network name or address of the optimizer server.
14	In the <b>Port</b> field, type the port on which the server is running on the remote machine.

Step	Action
15	<p>Click on the <b>OK</b> button to complete the configuration and open the project.</p> <p>This also adds the project name to the <b>Recent Projects</b> list on the Start Page; from now on, you can open the project again just by clicking on this name.</p>

#### Notes

- This feature is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions

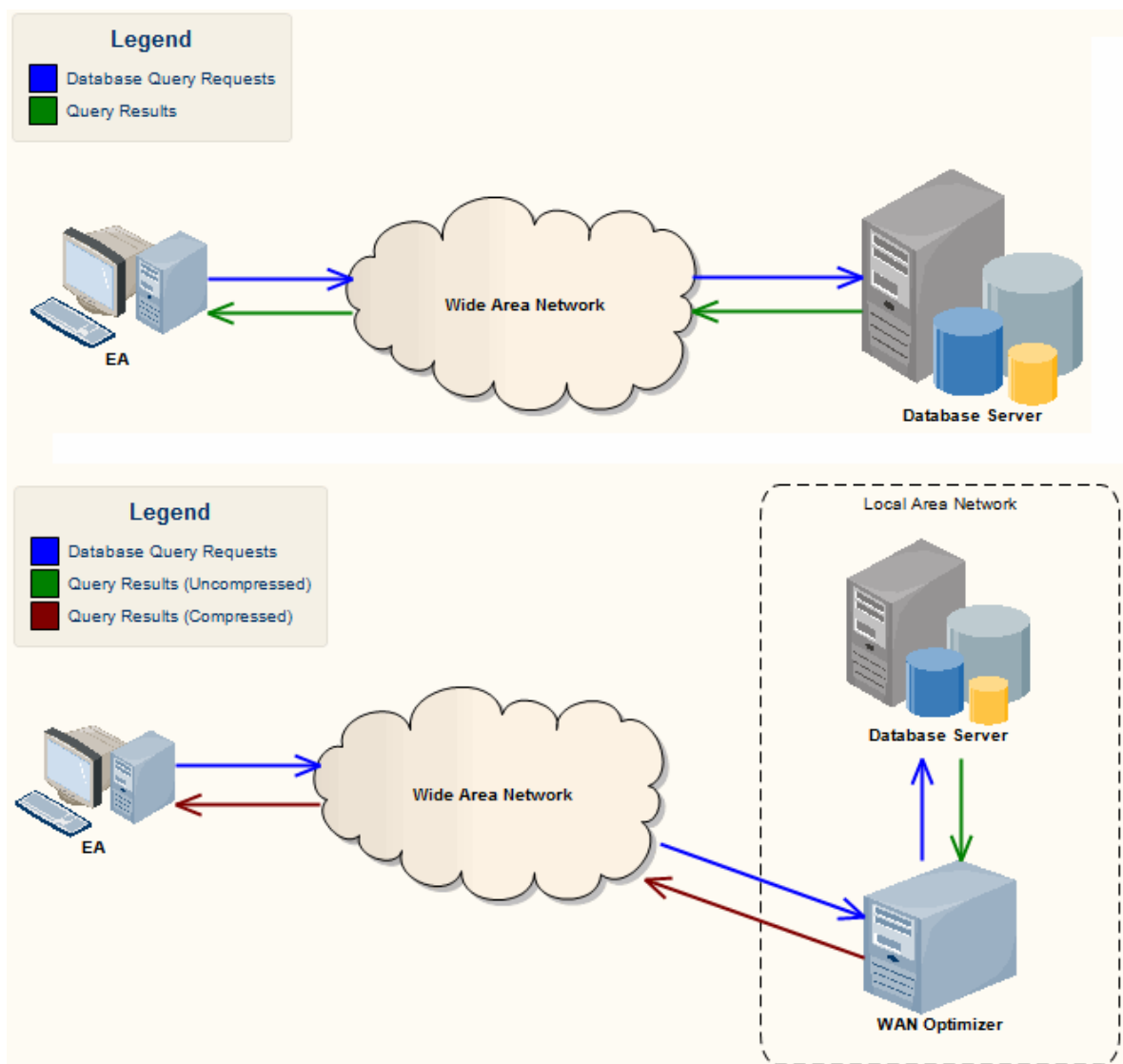
#### Learn more

- [Create a Project in an Oracle Database](#)<sup>[250]</sup>
- [Create an Oracle Data Repository](#)<sup>[253]</sup>
- [Open a Project](#)<sup>[202]</sup>
- [The WAN Optimizer](#)<sup>[259]</sup>
- [Connect to an Oracle Data Repository \(ODBC\)](#)<sup>[255]</sup>

### 3.3.8 The WAN Optimizer

You can significantly improve Enterprise Architect's performance in a Wide Area Network (WAN) by reducing the amount of data transmitted and, in turn, the number of network calls made. To achieve this, you can use the Sparx Systems *Wide Area Network (WAN) Optimizer*, a lightweight server installed on a Local Area Network (LAN) connection to a Database Management System (DBMS) that hosts an Enterprise Architect repository. You can configure the server to listen for client connections on a particular port; it acts as a local proxy to execute queries and return the results in a compressed format to the client.

In the following diagram, transmission between Enterprise Architect and a DBMS is depicted first without and then with the WAN Optimizer.



You can download the WAN Optimizer installer from the Downloads page of the Registered Users section of the Sparx Systems website. The Wan Optimizer Service installer package provides two installable features for the target machine:

- WAN Optimizer Service - the installer also helps register and start the service on the target machine, and add it to the Windows Startup folder
- WAN Optimizer Admin Client - to enable an administrator to administer and configure the service from a remote client

The Optimizer has its own *Sparx WAN Optimizer User Guide*. See that User Guide for more information on:

- WAN Optimizer Components
- Installing and Starting the WAN Optimizer Service
- Configuring the Service
- Troubleshooting

**Learn more**

- [WAN Optimizer](#) (Online Resource)
- [Connect to a SQL Server Data Repository](#)<sup>[218]</sup>
- [Connect to a MySQL Data Repository](#)<sup>[225]</sup>
- [Connect to a PostgreSQL Data Repository](#)<sup>[233]</sup>
- [Connect to an ASA Data Repository](#)<sup>[239]</sup>
- [Connect to a Progress OpenEdge Data Repository](#)<sup>[248]</sup>
- [Connect to an Oracle Data Repository \(ODBC\)](#)<sup>[255]</sup>
- [Connect to an Oracle Data Repository \(OLE DB\)](#)<sup>[257]</sup>

### 3.4 Connecting to Projects Via the Cloud

With a simple connection to the internet or web - **the 'Cloud'** - you can access and work on projects stored in repositories at **remote locations anywhere in the world**, or access local repositories from such locations, using the Enterprise Architect installation **on your machine**. The advantages of working in this way include:

- You, as a user, do not need to have database drivers installed on your machine or to create a database connection, as you do to work directly on projects held on a DBMS server; your system administrator performs the driver set-up and connection once only, during server configuration
- The http and, especially, https connections apply a firewall and all the security facilities provided by the web server - whether Apache or IIS - to your project work, such as URL authorization, domain authentication and IP security
- The Cloud Server can be configured to encrypt all communications using standard TLS/SSL protocols; you can be certain that your data is not intercepted during transmission across insecure networks

Using the Cloud can help to reduce lag time for common tasks such as opening Packages and editing model data. Requests are cached and sent together, rather than individually, which reduces communication delays to a minimum.

Using the Cloud server is most beneficial when:

- You want to minimize the set-up requirements each user has on their workstation
- You want to expose models held outside your private network
- Your users are accessing their models over slow connections

#### Learn more

- [Server Based Repositories](#) <sup>[214]</sup>
- [Setting up the Cloud Server](#) <sup>[262]</sup>
- [Connect to a Project Via the Cloud](#) <sup>[266]</sup>
- [Additional Functionality Using the Cloud](#) <sup>[267]</sup>

#### 3.4.1 Setting up the Cloud Server

One of the benefits of using the Cloud server is that you only need to perform the set-up once on the server, rather than separately on each user's workstation. This single-set-up process of installation, configuration and connection is described here.

##### Installation

To run the installer for Sparx Systems Cloud Services you must have administration permissions for installation on your system.

- Run the installer as an administrator
- Accept the license agreement
- Specify the location of the target directory
- Apply the options for installing the service itself, the management client and the IIS integration files

##### IIS Integration

IIS integration is not set up by default. Only the files are copied to the install target location.

### Configuration File

In the service installation directory is the **SCloudServices.config** file. Edit this file to set the ports that the server will 'listen' on, and other configuration options (below). When you make changes to this file, you must restart the server for those changes to take effect; you can do this using the Windows Services list.

The first group of settings are for the **TCP connection**, followed by a list of global, or **General**, settings that are applied across the entire service.

### **TCP Connection**

The TCP settings control the port used for administration tasks.

Setting	Description
<b>SERVER_PORT=803</b>	Sets the port to use when you connect to the administration client or opt to use the IIS integration instead of the integrated web server. It is recommended that this port is not exposed to external networks, as encryption cannot be applied to the port.
<b>SERVER_PASSWORD=</b>	The password to protect the administration functions of the server. This can also be changed directly within the Admin client.

### **General Settings**

Setting	Description
<b>DBMAN_DEFAULTMAXSIMQUERIES=10</b>	The default maximum number of queries that can be run at a time for any configured database. This can be changed directly within the Admin client.
<b>AUDIT_TIME_PERIOD=3600</b>	The number of seconds between the recording activities on the system logs on each database.
<b>TEMP_DIRECTORY=%SERVICE_PATH %Temp</b>	The location to which temporary files are written before they are sent to clients. You generally do not need to change this.
<b>LOGGING_LEVEL=SYSTEM</b>	Indicates how much text the server should provide when writing log files. The valid values are: <ul style="list-style-type: none"><li>• <b>OFF</b></li><li>• <b>FATAL</b></li><li>• <b>WARNING</b></li></ul>

Setting	Description
	<ul style="list-style-type: none"> <li>• <b>INFO</b> and</li> <li>• <b>SYSTEM</b></li> </ul> <p>The value can be changed directly within the Admin client.</p>
<b>LOGGING_DIRECTORY=%SERVICE_PATH%\Logs</b>	Collectively, identify where log files are written and how much history will be kept.
<b>LOGGING_FILECOUNT=3</b>	
<b>LOGGING_FILESIZE=1048576</b>	

### HTTP Ports

In the Cloud server you can define a number of different ports to listen to http connections, **each** with a different configuration and each denoted in the config file with an open and close parenthesis on their own line.

```
(
SERVER_PORT=804
REQUIRE_SSL=0
DEFAULT_MODEL=
MODEL_AUTHENTICATION=
GLOBAL_AUTHENTICATION=user model
OSLC_SUPPORT=1
)
```

Setting	Description
<b>SERVER_PORT</b>	The port that the server will listen on for http connections. Each port must be unique and not be used by any other services on the machine.
<b>REQUIRE_SSL</b>	When set to <b>1</b> , enables <b>https</b> on this port. This should be set to <b>1</b> for all connections that are being exposed on public networks, but it requires a private key ( <b>server.pem</b> ) to be included in the same directory before it will run.
<b>DEFAULT_MODEL</b>	Enables a single model to be exposed on a port, making it possible to use a different port for each model.
<b>MODEL_AUTHENTICATION</b>	Can be set to <b>1</b> to request http authorization using the list of security users in the model being connected to.  Note that if you are not using SSL to connect, the usernames and



Setting	Description
	passwords will be sent in plain text. This option is mutually exclusive with <b>GLOBAL_AUTHENTICATION</b> .
<b>GLOBAL_AUTHENTICATION</b>	Can be set to the name of a model with security enabled that will provide the list of users for <b>all</b> models provided by the connection. This is helpful if you want to provide multiple models but only manage one list of users. This option is mutually exclusive with <b>MODEL_AUTHENTICATION</b> .
<b>OSLC_SUPPORT</b>	Enabled by default (set to <b>1</b> ) to enable models to be queried using the Open Services for Lifecycle Collaboration ( <b>OSLC</b> ) standard. (See <i>OSLC Requirements Management</i> )  Set to <b>0</b> to disable the use of OSLC.

### Configure Models

Once your **service** is configured, you can connect to the Admin client to configure any **databases** you want to provide using the Cloud server.

Step	Action	See also
<b>1</b>	Open the Admin client.  An empty list of Database Management Systems displays.	
<b>2</b>	Click on the <b>Add</b> button to configure a new DBMS.  A dialog displays in which you provide the connection string that the server should use when connecting to a model. If you are running the Admin client on the same machine as the server you can click the ellipsis (...) button to open the Data Link Properties dialog to build the connection string.  This dialog also helps you to create a new <b>Firebird</b> database with all tables set up. This is the easiest way to get a connection running; just enter the model name followed by <b>.fdb</b> .	<a href="#">Server Based Repositories</a> [214]
<b>3</b>	Once you have added one or more Database Managers, they are included in the list in the main dialog.  Select any of these and click on the <b>Configure</b> button to allow connections to the model. A number of options are provided for the selected model.  <ul style="list-style-type: none"> <li>• <b>Accept Queries</b> - must be set to enable users to connect to this database</li> <li>• <b>Max Simultaneous Queries</b> - allows you to control the maximum number of simultaneous connections that will be created to this model; when the</li> </ul>	

Step	Action	See also
	<p>database was created the maximum number came from the system setting for this option</p> <p>To tweak constraints of system performance against resource usage, you can look at the audit history for each database to see how many connections have been used in the specified time period</p> <ul style="list-style-type: none"> <li>• <b>Run Scheduled Tasks</b> - allows the server to run periodic updates to this model</li> <li>• <b>Read-only connection</b> - allows a model to be shared but blocks any changes</li> <li>• <b>Require a secure and authenticated connection</b> - flags that security is required for this model; no connections will be accepted unless via https, with either <b>Model Authentication</b> or <b>Global Authentication</b> set</li> </ul>	<a href="#">Additional Functionality Using the Cloud</a> <a href="#">[267]</a>

### Security Considerations

As with any web connected service, you should consider a number of security concerns when setting up a new service.

- If any data is considered private, always use an **https** connection and require user authentication; there is an option on the database itself to set this
- There is an implicit trust in sharing a model with anyone; security is available to prevent users doing things that they shouldn't, but because Enterprise Architect allows user-written SQL to be used in queries in a number of places, any information can be at least retrieved

Be aware, this includes user names and hashes of their passwords, although this can be prevented by using **Global Authentication** instead of **Model Authentication** (see above)

### Learn more

- [Connecting to Projects Via the Cloud](#) [\[262\]](#)
- [Connect to a Project Via the Cloud](#) [\[266\]](#)
- [OSLC Requirements Management](#) [\[268\]](#)

## **3.4.2 Connect to a Project Via the Cloud**

A connection to the 'Cloud' is set up once; thereafter you can open the model via the link name in the Start Page **Recent** list. To create the link to a project via the Cloud, you need the access URL and model name as stored on the server; if you do not have this information, see your System Administrator.

Your link name can itself be a shortcut file that you provide to other users who need not know it is a Cloud connection. The Cloud connection makes no difference to the actions of any such Project Shortcut that you create; using the shortcut users can not only access a remote project simply, but also see it opened at specific diagrams, views and/or windows.

Access    **Start Page | Connect to Cloud**

Connect to Your Project

Field/Option/ Button	Action	See also
<b>Name</b>	Type in a name by which <b>you</b> can recognize and open the model in the <b>Recent</b> list on <b>your</b> Start Page or <b>File</b> drop-down menu.  This name does not affect the model loaded (see <b>Model Name</b> , below).	
<b>URL</b>	Type in the address or name of the computer hosting the model.  This starts with the protocol ( <i>http://</i> or <i>https://</i> ) followed by the name or IP address of the server and the port number; for example:  <code>http://localhost:804</code>  If you do not enter the port number, the system uses the default ports ( : <b>80</b> for http, : <b>443</b> for https).	
<b>Model Name</b>	Type in the model name by which the model is identified <b>on the server</b> .	
<b>Username</b>	If the host server is configured to request a username and password for authentication, you can type those values in here.  If you need to provide a user name and password and you require greater security (for example, if the password changes frequently or several users are using the connection) you can omit these values here, so that each person using this connection is prompted to provide them each time they access the model.	
<b>Password</b>		

Learn more

- [Connecting to Projects Via the Cloud](#)<sup>[262]</sup>
- [Setting up the Cloud Server](#)<sup>[262]</sup>
- [Project Shortcuts](#)<sup>[204]</sup>

**3.4.3 Additional Functionality Using the Cloud**

The Cloud services provide the facility of connecting to and using models over a secure http connection. They also make it possible to use three further facilities on or within your models:

- Open Services for Lifecycle Collaboration (OSLC)
- Reusable Asset Service (RAS)
- Scheduled Tasks

**OSLC**

OSLC is an initiative to allow easier integration between Requirement Management tools. It uses HTTP to

list, add, modify and delete requirements. The service provider definition to direct any OSLC client to is:

```
<protocol>://<server>:<port>/<model_name>/oslc/sp/
```

For example, if you were connecting to a server running on your own machine using the default settings, the connection would be:

```
http://localhost:804/model/oslc/sp/
```

## **RAS**

The RAS portion of the Cloud server enables you to define Packages that can be used in **any** model. When a Package is requested, Enterprise Architect and the Cloud server will track cross-Package dependencies and make available everything required by that Package.

## **Scheduled Tasks**

The Cloud server includes optional support for running time-based updates to data. Currently, this is applied to updating a **Time Series chart** automatically to provide a dynamic view of how a model is changing over time. You can set different time scales for these updates, such as daily, weekly or monthly.

## **Learn more**

- [OSLC Requirements Management](#)<sup>[268]</sup>
- [Reusable Asset Service](#)<sup>[282]</sup>
- [Define a Time Series Chart](#)<sup>[2767]</sup>

### **3.4.4 OSLC Requirements Management**

The following text is derived from the OSLC Primer:

**Open Services for Lifecycle Collaboration (OSLC)** is an open community creating specifications for integrating tools. These specifications allow conforming independent software and product lifecycle tools to integrate their data and workflows in support of end-to-end lifecycle processes.

OSLC is based on the W3C Linked Data. One of the primary techniques for integrating tools using OSLC is **Linking data via HTTP**, which specifies creating, retrieving, updating and deleting (CRUD) lifecycle artifacts based on internet standards like HTTP and RDF using Linked Data model. Each artifact in the lifecycle, such as a requirement, is an HTTP resource that is manipulated using the standard methods of the HTTP specification (like GET, POST).

Enterprise Architect acts as an OSLC **Provider** and supports the Requirements Management 2.0 specification of OSLC, which allows for creating, retrieving and querying the Requirements in a model accessed via a Cloud connection. With OSLC support, Requirements in an Enterprise Architect model can be identified and accessed using a unique URL that can be linked to resources in other lifecycle products and tools.

Enterprise Architect complies with the following Requirements Management 2.0 base requirements:

- Resource Operations
- Service Provider Resource
- Partial Resource Representations
- Creation Factory

- Query Capability
- Query Syntax
- Error Responses
- RDF/XML Representations

#### \*Beta Implementation\*

This is a **BETA** release of OSLC Requirements Management and associated tools. As such, it is supplied without warranty of any kind and is subject to change without notice during the Beta period. Sparx Systems welcomes any feedback, issues, suggestions and comments on this implementation. As with all Beta software, please take due care when using OSLC Requirements Management in a production environment.

#### Learn More

- [OSLC](#) (Online Resource)
- [OSLC Requirements Management 2.0](#) (Online Resource)
- [Linked Data](#) (Online Resource)
- [Requirements](#)<sup>[1763]</sup>
- [Connecting to Projects via the Cloud](#)<sup>[262]</sup>
- [Service Provider Resource](#)<sup>[269]</sup>
- [Resource Shape](#)<sup>[271]</sup>
- [Query Capability](#)<sup>[273]</sup>
- [Partial Resource Representations](#)<sup>[278]</sup>
- [Creation Factory](#)<sup>[280]</sup>

### **3.4.4.1 Service Provider and Service Provider Resource**

Enterprise Architect acts as an OSLC **Provider** that other OSLC-compliant tools (OSLC **Consumer/Clients**) access to link to its **Resources**. All OSLC Resources live in a **Service Provider**, which is a central organising concept of OSLC. In Enterprise Architect, each **model** that can be accessed via the Cloud connection is treated as an OSLC Service Provider. The Requirement elements in the model are the OSLC Resources.

The services offered by the Service Provider can be retrieved using the **Service Provider Resource**. A Service Provider Resource specifies the:

- URL to which you can POST representations to create new resources
- URL you can use to GET a list of existing resources

To retrieve the Service Provider Resource from an Enterprise Architect model connected via the Cloud, use the URL:

`http://<server>/<model_name>/oslc/sp/`

For example, the Service Provider Resource for a model called *firebird\_model*, connected via the Cloud, would be accessed using the URL:

`http://localhost:480/firebird_model/oslc/sp/`

This retrieved resource would resemble this:

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:dcterms="http://purl.org/dc/terms/" xmlns:oslc="http://open-services.net/ns/core#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <oslc:ServiceProvider rdf:about="http://localhost:480/firebird_model/oslc/sp/">
    <dcterms:title>firebird:anonymous@firebird_model</dcterms:title>
    <dcterms:publisher>
      <oslc:Publisher>
        <dcterms:title>Sparx Systems</dcterms:title>
        <dcterms:identifier>http://www.sparxsystems.com.au/</dcterms:identifier>
      </oslc:Publisher>
    </dcterms:publisher>
    <oslc:service>
      <oslc:Service>
        <oslc:domain rdf:resource="http://open-services.net/ns/rm#">
          <oslc:creationFactory>
            <oslc:CreationFactory>
              <dcterms:title>Requirements Creation</dcterms:title>
              <oslc:creation rdf:resource="http://localhost:480/firebird_model/oslc/cf/">
                <oslc:resourceShape rdf:resource="http://localhost:480/firebird_model/oslc/rs/">
                  </oslc:resourceShape>
              </oslc:creationFactory>
            </oslc:creationFactory>
            <oslc:queryCapability>
              <oslc:QueryCapability>
                <dcterms:title>Requirements Query</dcterms:title>
                <oslc:queryBase rdf:resource="http://localhost:480/firebird_model/oslc/qc/">
                  <oslc:resourceShape rdf:resource="http://localhost:480/firebird_model/oslc/rs/">
                    </oslc:resourceShape>
                  </oslc:queryCapability>
                </oslc:QueryCapability>
              </oslc:queryCapability>
            </oslc:Service>
          </oslc:Service>
        </oslc:domain>
      </oslc:Service>
    </oslc:service>
    <oslc:prefixDefinition>
      <oslc:PrefixDefinition>
        <oslc:prefix>foaf</oslc:prefix>
        <oslc:prefixBase rdf:resource="http://xmlns.com/foaf/0.1/">
          </oslc:prefixBase>
        </oslc:PrefixDefinition>
      </oslc:PrefixDefinition>
      <oslc:PrefixDefinition>
        <oslc:PrefixDefinition>
          <oslc:prefix>rdfs</oslc:prefix>
          <oslc:prefixBase rdf:resource="http://www.w3.org/2000/01/rdf-schema#">
            </oslc:prefixBase>
          </oslc:PrefixDefinition>
        </oslc:PrefixDefinition>
        <oslc:PrefixDefinition>
          <oslc:PrefixDefinition>
            <oslc:prefix>dcterms</oslc:prefix>
            <oslc:prefixBase rdf:resource="http://purl.org/dc/terms/">
              </oslc:prefixBase>
            </oslc:PrefixDefinition>
          </oslc:PrefixDefinition>
          <oslc:PrefixDefinition>
            <oslc:PrefixDefinition>
              <oslc:prefix>oslc_rm</oslc:prefix>
              <oslc:prefixBase rdf:resource="http://open-services.net/ns/rm#">
                </oslc:prefixBase>
              </oslc:PrefixDefinition>
            </oslc:PrefixDefinition>
            <oslc:PrefixDefinition>
              <oslc:PrefixDefinition>
                <oslc:prefix>ss</oslc:prefix>
                <oslc:prefixBase rdf:resource="http://www.sparxsystems.com.au/">
                  </oslc:prefixBase>
                </oslc:PrefixDefinition>
              </oslc:PrefixDefinition>
              <oslc:PrefixDefinition>
                <oslc:PrefixDefinition>
                  <oslc:prefix>oslc</oslc:prefix>
                  <oslc:prefixBase rdf:resource="http://open-services.net/ns/core#">
                    </oslc:prefixBase>
                  </oslc:PrefixDefinition>
                </oslc:PrefixDefinition>
                <oslc:PrefixDefinition>
                  <oslc:PrefixDefinition>
                    <oslc:prefix>rdf</oslc:prefix>
                    <oslc:prefixBase rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
                      </oslc:prefixBase>
                    </oslc:PrefixDefinition>
                  </oslc:PrefixDefinition>
                </oslc:PrefixDefinition>
              </oslc:PrefixDefinition>
            </oslc:PrefixDefinition>
          </oslc:PrefixDefinition>
        </oslc:PrefixDefinition>
      </oslc:PrefixDefinition>
    </oslc:ServiceProvider>
  </rdf:RDF>
```

→ Create new Requirements in the Model by sending POST request to this URL

→ Access the Requirements in the Model by sending GET request to this URL

→ Requirements representation is available in this URL

### Elements of the Service Provider Resource

Element	Description
<b>oslc:Publisher</b>	Specifies the OSLC Provider.
<b>oslc:Service</b>	Specifies the services offered by the OSLC Provider. Enterprise Architect supports the following OSLC Requirement Management services: <ul style="list-style-type: none"> <li>• <b>Creation Factory</b> - Used to create new Requirements in the model, by passing</li> </ul>

Element	Description
	<p>the Requirement's representation in RDF format using HTTP POST to the URL:</p> <p><code>http://&lt;server&gt;/&lt;model_name&gt;/oslc/cf/</code></p> <ul style="list-style-type: none"> <li>• <b>Query Capability</b> - Used to list/query the Requirements in the model, by passing the query using HTTP GET to the URL:</li> </ul> <p><code>http://&lt;server&gt;/&lt;model_name&gt;/oslc/qc/</code></p>
<b>oslc:resourceShape</b>	<p>Specifies the Requirement's metadata; that is, its properties and constraints. These are available in the URL:</p> <p><code>http://&lt;server&gt;/&lt;model_name&gt;/oslc/rs/</code></p>
<b>oslc:prefixDefinition</b>	Specifies the namespace prefixes and their namespace definitions.

#### Learn more

- [Requirements](#)<sup>[1763]</sup>
- [Resource Shape](#)<sup>[271]</sup>
- [Query Capability](#)<sup>[273]</sup>
- [Creation Factory](#)<sup>[280]</sup>

### 3.4.4.2 Resource Shape

Resource Shape specifies a Requirement's metadata; that is, the set of standard OSLC properties and custom Enterprise Architect properties. Resource Shape can be accessed using the URL:

`http://<server>/<model_name>/oslc/rs/`

The properties of both types specified in the Resource Shape map to specific fields in the Requirement's Properties dialog in Enterprise Architect. For each property, the Resource Shape can also specify constraints.

#### OSLC Properties

(These are as defined in the *Dublin Core Metadata Element Set*.)

Property Name	Requirement element Properties dialog field
<b>title</b>	Short Description
<b>description</b>	Notes
<b>subject</b>	Key Words

Property Name	Requirement element Properties dialog field
<b>creator</b>	Author
<b>created</b>	Created
<b>modified</b>	Last Updated

#### Custom Enterprise Architect properties

Property Name	Requirement element Properties dialog field
<b>alias</b>	Alias
<b>status</b>	Status
<b>difficulty</b>	Difficulty
<b>priority</b>	Priority
<b>type</b>	Type
<b>phase</b>	Phase
<b>version</b>	Version

#### Constraints

Constraint	Meaning
<b>name</b>	The name of the property.
<b>valueType</b>	The type of value the property can have, such as <i>string</i> , <i>dateTime</i> or <i>integer</i> .
<b>occurs</b>	The cardinality of the property; that is, whether the property is optional or



Constraint	Meaning
	required.
<b>maxSize</b>	The maximum number of characters for a <i>string</i> valueType.
<b>allowedValue</b>	The list of values that can be assigned for the property.
<b>readOnly</b>	Determines whether a value for the property can be set by the client.

#### Notes

- The property **identifier** refers to a Requirement's unique Enterprise Architect GUID
- The property **packageID** refers to the ID of the Package under which the Requirement exists in the model

#### Learn more

- [Dublin Core 1.1](#) (Online Resource)
- [Requirement Properties](#) <sup>[1774]</sup>

### 3.4.4.3 Query Capability

Clients can query a model and retrieve the Requirements that match a specific criteria. In Enterprise Architect, the base URI for accessing the Query Capability is:

`http://<server>/<model_name>/oslc/qc/`

A query string expressing the specific criteria should be added to the base URI and addressed to the model using an HTTP GET request. The response for this request will be in RDF/XML format. For example, all the Requirements in a model called *firebird\_model* connected by the Cloud can be retrieved using the URL:

`http://localhost:480/firebird_model/oslc/qc/`

The response for the request will resemble this:

```
<?xml version="1.0" encoding="UTF-8"?>
- <rdf:RDF xmlns:ss="http://www.sparxsystems.com.au/" xmlns:foaf="http://xmlns.com/foaf/0.1/" xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:oslc_rm="http://open-services.net/ns/rm#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
-   <rdf:Description rdf:about="http://localhost:480/firebird_model/oslc/qc/">
-     <rdfs:member>
-       <oslc_rm:Requirement rdf:about="http://localhost:480/firebird_model/oslc/re/{58089E4F-E705-46ae-992E-0D876D7F5DF5}/">
-         <rdf:type>http://open-services.net/ns/rm#Requirement</rdf:type>
-         <dcterms:title>Requirement2</dcterms:title>
-         <dcterms:description>Requirement2 Notes</dcterms:description>
-         <dcterms:identifier>{58089E4F-E705-46ae-992E-0D876D7F5DF5}</dcterms:identifier>
-         <dcterms:subject>risk, loading factor</dcterms:subject>
-         <dcterms:creator>
-           <foaf:Person>
-             <foaf:name>User1</foaf:name>
-           </foaf:Person>
-         </dcterms:creator>
-         <dcterms:created>2014-01-06 11:29:23</dcterms:created>
-         <dcterms:modified>2014-01-06 13:20:05</dcterms:modified>
-         <ss:alias>Requirement2Alias</ss:alias>
-         <ss:status>Proposed</ss:status>
-         <ss:difficulty>Medium</ss:difficulty>
-         <ss:priority>Medium</ss:priority>
-         <ss:type>Display</ss:type>
-         <ss:phase>1.0</ss:phase>
-         <ss:version>1.0</ss:version>
-         <ss:packageID>355</ss:packageID>
-       </oslc_rm:Requirement>
-     </rdfs:member>
-   </rdfs:member>
- </rdf:Description>
</rdf:RDF>
```

### Elements of the response

Element	Description
<b>oslc_rm:Requirement</b>	A Requirement and its properties (as defined in the Resource Shape).  The attribute <b>rdf:about</b> on this element specifies the URL for accessing the Requirement. The format for the URL is:  <code>http://&lt;server&gt;/&lt;model_name&gt;/oslc/re/&lt;requirement_GUID&gt;/</code>
<b>Elements with namespace prefix <i>dcterms</i></b>	The standard properties from the Dublin Core Metadata Element Set.
<b>Elements with namespace prefix <i>ss</i></b>	The custom Enterprise Architect properties.

In Enterprise Architect, the Query Capability supports these parameters:

- **oslc.where**
- **oslc.select**
- **oslc.properties**
- **oslc.prefix**

**Learn more**

- [Dublin Core 1.1](#) (Online Resource)
- [Resource Shape](#)<sup>[271]</sup>
- [WHERE Parameter](#)<sup>[275]</sup>
- [SELECT Parameter](#)<sup>[276]</sup>
- [Combine WHERE and SELECT Parameters](#)<sup>[277]</sup>
- [PROPERTIES Parameter](#)<sup>[278]</sup>
- [PREFIX Parameter](#)<sup>[279]</sup>

**3.4.4.3.1 WHERE Parameter**

The **oslc.where** Query parameter specifies the conditions that the resources must satisfy; it is similar to the WHERE clause of an SQL statement. This parameter must be appended to the base URI of the Query Capability in Enterprise Architect:

```
http://<server>/<model_name>/oslc/qc/
```

The syntax for the **oslc.where** Query parameter (defined in BNF grammar and as specified in the OSLC Query Syntax Specification) is:

```
oslc_where    ::= "oslc.where=" compound_term
compound_term ::= simple_term (space? boolean_op space? simple_term)*
simple_term    ::= term | scoped_term
space         ::= " " /* a space character */
boolean_op    ::= "and"
term          ::= identifier_wc comparison_op value | identifier_wc space in_op space? in_val
scoped_term   ::= identifier_wc "{" compound_term "}"
identifier_wc ::= identifier | wildcard
identifier    ::= PrefixedName
PrefixedName ::= /* see "SPARQL Query Language for RDF", http://www.w3.org/TR/rdf-sparql-query/#rPrefixedName */
wildcard      ::= "*"
comparison_op ::= "=" | "!=" | "<" | ">" | "<=" | ">="
in_op         ::= "in"
in_val        ::= "[" value ("," value)* "]"
value         ::= uri_ref_esc | literal_value
uri_ref_esc   ::= /* an angle bracket-delimited URI reference in which > and \ are \-escaped. */
literal_value ::= boolean | decimal | string_esc (LANGTAG | ("^^" PrefixedName))?
boolean       ::= "true" | "false"
decimal       ::= /* see "XML Schema Part 2: Datatypes Second Edition", http://www.w3.org/TR/xmlschema-2/ */
string_esc    ::= /* a string enclosed in double quotes, with certain characters escaped. See below. */
LANGTAG       ::= /* see "SPARQL Query Language for RDF", http://www.w3.org/TR/rdf-sparql-query/#rLANGTAG */
```

These example queries act on a model called *firebird\_model* connected by the Cloud:

Query	Response
<code>http://localhost:480/firebird_model/oslc/qc/?oslc.where= dcterms: title="Requirement1"</code>	Retrieves all the OSLC properties of all the Requirements having the <b>title Requirement1</b> .
<code>http://localhost:480/firebird_model/oslc/qc/?oslc.where= dcterms:title in ["Requirement1","Requirement2"]</code>	Retrieves all the OSLC properties of all the

Query	Response
	Requirements having the <b>title</b> <i>Requirement1</i> or <i>Requirement2</i> .
<code>http://localhost:480/firebird_model/oslc/qc/?oslc.where=dterms:title="Requirement1" and dterms:creator{foaf:name}="User1"</code>	Retrieves all the OSLC properties of all the Requirements having the <b>title</b> <i>Requirement1</i> , created by <i>User1</i> .
<code>http://localhost:480/firebird_model/oslc/qc/?oslc.where=dterms:creator{foaf:name}="User1" and dterms:created&lt;"2014-01-07"</code>	Retrieves all the OSLC properties of all the Requirements created by <i>User1</i> prior to <i>7th January 2014</i> .
<code>http://localhost:480/firebird_model/oslc/qc/?oslc.where=name="Requirement1"</code>	Returns an error response as <b>name</b> is not a valid property in Enterprise Architect.
<code>http://localhost:480/firebird_model/oslc/qc/?oslc.where=dterms:created&lt;"01-07-2014"</code>	Returns an error response as the value of the property <b>created</b> is not in a valid format.

#### Notes

- Dates must be specified in the format **YYYY-MM-DD**

#### Learn more

- [OSLC WHERE](#) (Online Resource)
- [Resource Shape](#)<sup>[27]</sup>
- [Query Capability](#)<sup>[27]</sup>

#### 3.4.4.3.2 SELECT Parameter

The **oslc.select** Query parameter specifies the Requirement properties to be retrieved; it is similar to the SELECT clause of an SQL statement. This parameter must be appended to the base URI of the Query Capability in Enterprise Architect:

`http://<server>/<model_name>/oslc/qc/`

The syntax for the **oslc.select** Query parameter (defined in BNF grammar and as specified in the OSLC Query Syntax Specification) is:

```

oslc_select ::= "oslc.select=" properties
properties ::= property ("," property)*
property    ::= identifier | wildcard | nested_prop
nested_prop ::= (identifier | wildcard) "{" properties "}"

```

These example queries act on a model called *firebird\_model* connected by the Cloud:

Query	Response
<code>http://localhost:480/firebird_model/oslc/qc/?oslc.select= dterms: title</code>	Retrieves the OSLC property <b>title</b> of all the Requirements.
<code>http://localhost:480/firebird_model/oslc/qc/?oslc.select=dterms: title,dterms:created,dterms:creator{foaf:name},ss:version</code>	Retrieves the OSLC properties <b>title</b> , <b>creator</b> and <b>version</b> of all the Requirements.
<code>http://localhost:480/firebird_model/oslc/qc/?oslc.select=*</code>	Retrieves all the OSLC properties of all the Requirements.
<code>http://localhost:480/firebird_model/oslc/qc/?oslc.select=dterms: name</code>	Returns an error response, as <b>name</b> is not a valid property in Enterprise Architect.

#### Learn more

- [OSLC SELECT](#) (Online Resource)
- [Resource Shape](#)<sup>[27]</sup>
- [Query Capability](#)<sup>[27]</sup>

#### 3.4.4.3.3 Combine WHERE and SELECT Parameters

The **oslc.where** and **oslc.select** Query parameters can be combined in the base URI of the Query Capability to retrieve the required properties of all those Requirements that satisfy the specified condition. This is similar to using WHERE and SELECT clauses together in a SQL statement.

These example queries act on a model called *firebird\_model* connected by the Cloud:

Query	Response
<code>http://localhost:480/firebird_model/oslc/qc/?oslc.where= dterms: title="Requirement1" &amp; oslc.select=dterms:title</code>	Retrieves the OSLC property <b>title</b> of all the Requirements that have the <b>title Requirement1</b> .
<code>http://localhost:480/firebird_model/oslc/qc/?oslc.where= dterms: title in ["Requirement1","Requirement2"] &amp; oslc.select=dterms: title,dterms:created,dterms:creator{foaf:name}</code>	Retrieves the OSLC properties <b>title</b> , <b>created</b> and <b>creator</b> of all the Requirements that have the title

Query	Response
	<i>Requirement1</i> or <i>Requirement2</i> .
<code>http://localhost:480/firebird_model/oslc/qc/?oslc.where=dcterms:title="Requirement1" and dcterms:creator{foaf:name}="User1" &amp; oslc.select=*</code>	Retrieves all the OSLC properties of all the Requirements that have the <b>title</b> <i>Requirement1</i> , created by <i>User1</i> .
<code>http://localhost:480/firebird_model/oslc/qc/?oslc.where=dcterms:created&lt;"01-07-2014" oslc.select=dcterms:title</code>	Returns an error response, as the value of the property <b>created</b> is not in the valid format.

### Notes

- Dates must be specified in the format **YYYY-MM-DD**

### Learn more

- [Resource Shape](#)<sup>[27]</sup>
- [WHERE Parameter](#)<sup>[27]</sup>
- [SELECT Parameter](#)<sup>[27]</sup>

#### 3.4.4.3.4 PROPERTIES Parameter

Enterprise Architect supports a technique called **Selective Properties**, through which clients can retrieve selected OSLC properties of a Requirement. This technique accepts a partial representation of the OSLC properties - that is, all properties or only some of them. The base URI for accessing Selective Properties of a Requirement in Enterprise Architect is :

`http://<server>/<model_name>/oslc/re/<requirement_GUID>/`

The syntax for the **oslc.properties** Query parameter (defined in BNF grammar and as specified in the OSLC Core Specification) is:

```
oslc_properties ::= "oslc.properties=" properties
properties      ::= property ("," property)*
property        ::= identifier | wildcard | nested_prop
nested_prop     ::= (identifier | wildcard) "{" properties "}"
wildcard        ::= "*"
identifier      ::= PrefixedName
PrefixedName    ::= /* see "SPARQL Query Language for RDF", http://www.w3.org/TR/rdf-sparql-query/#rPrefixedName */
```

These example queries act on a Requirement with the GUID {7104C13D-841C-4068-B7EE-FB998C5BA4B7} in a model called *firebird\_model* connected by the Cloud:

Query	Response
<code>http://localhost:480/firebird_model/oslc/re/{7104C13D-841C-4068-B7EE-FB998C5BA4B7}/?oslc.properties=*</code>	Retrieves all the OSLC properties of the specified Requirement.

Query	Response
<code>http://localhost:480/firebird_model/oslc/re/{7104C13D-841C-4068-B7EE-FB998C5BA4B7}/?oslc.properties=dcterms:title,dcterms:creator{foaf:name}</code>	Retrieves the OSLC properties <b>title</b> and <b>creator</b> of the specified Requirement.
<code>http://localhost:480/firebird_model/oslc/re/{7104C13D-841C-4068-B7EE-FB998C5BA4B7}/?oslc.properties=dcterms:title,dcterms:creator{}</code>	Returns an error response as the property <b>creator</b> is incomplete; it should be:  <code>dcterms:creator{foaf:name}</code>

#### Learn more

- [OSLC PROPERTIES](#) (Online Resource)
- [Resource Shape](#)<sup>[27]</sup>
- [Query Capability](#)<sup>[27]</sup>

#### 3.4.4.3.5 PREFIX Parameter

Clients can use the **oslc.prefix** parameter to specify URI prefixes as used in OSLC property names. This parameter is appended to the base URI of the Selective Properties of a requirement:

`http://<server>/<model_name>/oslc/re/<requirement_GUID>/`

The syntax for the **oslc.prefix** parameter (defined in BNF grammar and as specified in the OSLC Core Specification) is:

```
oslc_prefix ::= "oslc.prefix=" prefix_defs
prefix_defs ::= prefix_def ("," prefix_def)*
prefix_def  ::= prefix "=" uri_ref_esc
prefix      ::= PN_PREFIX
PN_PREFIX   ::= /* see "SPARQL Query Language for RDF", http://www.w3.org/TR/rdf-sparql-query/#rPN_PREFIX */
uri_ref_esc ::= /* an angle bracket-delimited URI reference in which > and \ are \-escaped. */
```

These example queries act on a Requirement with the GUID `{7104C13D-841C-4068-B7EE-FB998C5BA4B7}` in a model called `firebird_model` connected by the Cloud:

Query	Response
<code>http://localhost:480/firebird_model/oslc/re/{7104C13D-841C-4068-B7EE-FB998C5BA4B7}/?oslc.prefix=otrm=&lt;http://purl.org/dc/terms/&gt; &amp; oslc.properties=otrm:title</code>	Retrieves the OSLC property <b>title</b> of the specified Requirement.  Note that the prefix for the namespace <code>http://purl.org/dc/terms/</code> has been specified as <b>otrm</b> in the query.
<code>http://localhost:480/firebird_model/oslc/re/{7104C13D-841C-4068-B7EE-FB998C5BA4B7}/?oslc.prefix=otrm=&lt;http://purl.org/dc/terms/&gt; &amp; oslc.properties=otrm:title,otrm:alias</code>	Retrieves the OSLC properties <b>title</b> and <b>alias</b> of the specified Requirement.

Query	Response
org/dc/terms/>>,spx=<http://www.sparxsystems.com.au/>&oslc.properties=otrm:title,spx:alias	Note that the prefix for the namespace <i>http://purl.org/dc/terms/</i> has been specified as <b>otrm</b> and that of <i>http://www.sparxsystems.com.au/</i> has been specified as <b>spx</b> in the query.
http://localhost:480/firebird_model/oslc/re/{7104C13D-841C-4068-B7EE-FB998C5BA4B7}?oslc.prefix=otrm=<http://purl.org/dc/terms/>>,spx=<http://www.sparxsystems.com.au/>&oslc.properties=otrm:title,sx:alias	Returns an error response as the namespace prefix <b>sx</b> on the property <b>alias</b> is undefined.

#### Learn more

- [OSLC PREFIX](#) (Online Resource)
- [Resource Shape](#)<sup>[27]</sup>
- [Query Capability](#)<sup>[27]</sup>
- [PROPERTIES Parameter](#)<sup>[27]</sup>

#### 3.4.4.4 Creation Factory

Enterprise Architect supports the **Creation Factory** OSLC service, through which clients can create new Requirements via HTTP **POST**. To create a new Requirement, the client POSTs a representation of the Requirement in **RDF** format to the Creation Factory URL. If the POST is successful, the HTTP location header of the response will contain the URL of the created Requirement. An unsuccessful POST will generate an error response.

The Creation Factory URL has the format:

`http://<server>/<model_name>/oslc/cf/`

Some example representations of a Requirement in RDF format are:

RDF Representation	Result
<pre>&lt;rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"   xmlns:oslc_rm="http://open-services.net/ns/rm#"   xmlns:dcterms="http://purl.org/dc/terms/"   xmlns:foaf="http://xmlns.com/foaf/0.1/"   xmlns:ss="http://www.sparxsystems.com.au/"&gt;   &lt;oslc_rm:requirement&gt;   &lt;dcterms:title&gt;Requirement3&lt;/dcterms:title&gt;   &lt;dcterms:description&gt;Requirement Notes     &lt;/dcterms:description&gt;   &lt;dcterms:creator&gt;     &lt;foaf:Person&gt;       &lt;foaf:name&gt;User1&lt;/foaf:name&gt;     &lt;/foaf:Person&gt;   &lt;/dcterms:creator&gt;   &lt;ss:type&gt;Functional&lt;/ss:type&gt;   &lt;ss:packageID&gt;355&lt;/ss:packageID&gt;</pre>	Creates a new Requirement with the specified <b>name</b> , <b>notes</b> , <b>author</b> and <b>stereotype</b> .



RDF Representation	Result
<pre> &lt;/oslc_rm:requirement&gt; &lt;/rdf:RDF&gt; </pre>	
<pre> &lt;rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"   xmlns:oslc_rm="http://open-services.net/ns/rm#"   xmlns:dct="http://purl.org/dc/terms/"   xmlns:foaf="http://xmlns.com/foaf/0.1/"   xmlns:ss="http://www.sparxsystems.com.au/"&gt;   &lt;oslc_rm:requirement&gt;     &lt;dct:title&gt;Requirement4&lt;/dct:title&gt;     &lt;ss:packageID&gt;355&lt;/ss:packageID&gt;     &lt;ss:difficulty&gt;High&lt;/ss:difficulty&gt;     &lt;ss:priority&gt;High&lt;/ss:priority&gt;   &lt;/oslc_rm:requirement&gt; &lt;/rdf:RDF&gt; </pre>	Creates a new Requirement with the specified <b>name</b> , <b>difficulty</b> and <b>priority</b> .
<pre> &lt;rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"   xmlns:oslc_rm="http://open-services.net/ns/rm#"   xmlns:dct="http://purl.org/dc/terms/"   xmlns:foaf="http://xmlns.com/foaf/0.1/"   xmlns:ss="http://www.sparxsystems.com.au/"&gt;   &lt;oslc_rm:requirement&gt;     &lt;dct:title&gt;Requirement4&lt;/dct:title&gt;     &lt;ss:packageID&gt;355&lt;/ss:packageID&gt;     &lt;ss:difficulty/&gt;   &lt;/oslc_rm:requirement&gt; &lt;/rdf:RDF&gt; </pre>	Produces an error response, as the property <b>difficult</b> has an invalid (empty) value.

**Note**

- The OSLC properties **title** and **packageID** of the Requirement are mandatory and must be supplied in the RDF representation
- The value of the OSLC property **packageID** should already exist in the model, so before creating a Requirement using Creation Factory, use the Query Capability URL to determine the existing values of **packageID**
- The OSLC properties **identifier**, **created** and **modified** of the Requirement are read-only and cannot be set using the Creation Factory service

**Learn More**

- [Resource Shape](#)<sup>[27]</sup>
- [Query Capability](#)<sup>[273]</sup>

### 3.5 Reuseable Asset Service

Within a large organization, groups of users - such as model developers - can be separated by geographical distance and/or being on different networks. This can make it difficult to share common data, standards and modeling structures easily, without the complexity of using external version control tools or manually distributing XML files between projects. However, within Enterprise Architect the **Reusable Asset Service (RAS)** provides a simple and convenient mechanism for modelers to distribute or download **reusable** model structures, information, corporate directives and standards, through a **shared repository**, accessible via a **Cloud Service** connection. The person who sets up the reusable data can retain ownership and management of the resource - or **asset** - whilst their distant colleagues can quickly review the currency of the information and download the latest versions into their models or file folders.

The RAS gives distributed teams convenient access to a single 'source of truth' for shared data, including project milestones, architectural frameworks and industry standards.

#### Reusable Assets

Reusable Assets consist of:

- **Packages** containing elements, diagrams and structures (as drawn from any point within the Project Browser), and
- **Files** in a range of text, code and graphic formats, including .eap files

An asset can be, for example:

- A common Class library or framework
- A set of common Requirements or Use Cases
- A draft specification document
- Marketing collateral

#### Storage Structure and Use

Reusable Assets are held in a remote **registry**, accessed through a Cloud Service connection. The registry contains any number of **Storages**, which any user can create. Each Storage can contain any number of **Asset Packages** holding modeling structures, and **files** containing textual or graphical information and data. When a user creates a Storage, they can protect the contents from being updated in the Registry or downloaded into a model, using **password protection** defined by that user.

For each Package, the RAS automatically identifies the:

- Version of the Package held in the registry
- Diagrams and elements (including child Package elements) contained by the Asset Package
- Dependencies on parent Packages of any external elements that the Asset Package references
- Dependencies on MDG Technologies

Any user, regardless of password protection, can freely browse and identify the contents of a Package held in a Storage - including displaying the diagrams - without importing the material into a model. A user that has imported an Asset Package into their model can compare their model Package against any version of the Asset Package, to check for and assess any differences between them.

#### Notes

- The Reusable Asset Service is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect

#### Learn more


- [Connect to the Asset Service](#) <sup>[283]</sup>
- [Browse Assets](#) <sup>[284]</sup>
- [Import an Asset into the Model](#) <sup>[291]</sup>
- [Set Up the Asset Service](#) <sup>[293]</sup>
- [Register New Assets](#) <sup>[298]</sup>
- [Update an Asset](#) <sup>[304]</sup>

### 3.5.1 Connect to the Asset Service

The Reusable Asset Service provides model structures and documents to you from one or more registries on remote systems. You access these registries through a Cloud connection to the appropriate server. The connection details should be provided to you by your System Administrator or Reusable Asset Service Administrator.

#### Access    **Project | Reusable Asset Service**

#### Connect to Assets

Field/Option	Detail	See also
<b>Registry</b>	<p>This field normally defaults to the last-used registry name. If the field is empty, or if you want to switch to a different registry, click on the  (Browse) button to the right of the field.</p> <p>The Cloud Connection dialog displays.</p> <p>Using the information provided by your administrator, type in the server name, URL and the name of the model hosting the registry data.</p> <p>If advised by your Administrator, type in the user name and password for the server.</p> <p>Click on the <b>OK</b> button; the <b>Registry</b> field now shows the server name.</p>	<p><a href="#">Connect to a Project Via the Cloud</a> <sup>[266]</sup></p>
<b>Storage</b>	<p>If there is more than one Storage available in the Registry, this field defaults to the first listed in the Registry.</p> <p>To check for or select other Storages, click on the drop-down arrow at the right of the field. Click on the Storage to review.</p> <p>The Registry Browser tab displays the Packages registered in the selected Storage.</p>	<p><a href="#">Browse Assets</a> <sup>[284]</sup></p>

#### Notes

- The Reusable Asset Service is available in the Corporate, Business and Software Engineering,

System Engineering and Ultimate editions of Enterprise Architect


#### Learn more

- [Reuseable Asset Service](#)<sup>[282]</sup>
- [Set Up the Asset Service](#)<sup>[293]</sup>

### 3.5.2 Browse Assets

When you select a **Storage** in the Reusable Asset Service, the **Registry Browser** tab immediately displays a list of the **Packages** registered in that Storage. You can review and use these Packages using the context menu and buttons on the tab.

You can review the properties and contents of the selected Package in the Storage, and access any files in the Storage, using the tabs in the **lower** half of the Reusable Asset Service view, underneath the Registry Browser tab.

If you have left the Registry Browser open for a while and there is a possibility that the Registry has been changed, you can click on the  icon in the Reusable Asset Service toolbar to refresh the Browser with the latest contents of the Registry.

Access    **Project | Reusable Asset Service > Registry Browser**

#### Review Assets

Field/Option	Detail	See also
<b>Toggle Filter Bar</b>	Right-click on the column headings and select this option to show or hide the <b>Filter Bar</b> on the display.	
<b>&lt;filter bar fields&gt;</b>	Type in the appropriate characters to list only Packages that have that string of characters in the values in the corresponding column.	<a href="#">List Header</a> <sup>[678]</sup>
<b>Package</b>	This column shows the <b>name</b> of the Packages held in the selected Storage.	
<b>Version</b>	This column defaults to the most recent <b>version</b> of each Package held in the selected Storage.  You can click on the drop-down arrow at the end of a field and select a <b>different version</b> of the corresponding Package, if any have been registered.	
<b>Last Registered</b>	This column shows the <b>date and time</b> at which the currently-listed version of each Package was registered.	
<b>Registered By</b>	This column shows the <b>user name</b> of the person who registered the currently-listed version of each Package.	

Field/Option	Detail	See also
<b>Import</b>	<p>Click on a Package name and click on this button to begin to <b>import</b> either the Package alone or the Package and its dependent structures into your <b>model</b>.</p> <p>You can perform the same action by right-clicking on the Package name and selecting the <b>Import from Registry</b> menu option.</p> <p>If the Package from the Registry <b>already exists</b> in your model, it is deleted and replaced by this import.</p>	<a href="#">Import an Asset into the Model</a> <sup>[291]</sup>
<b>Compare with Package in Model</b>	<p>If you have previously imported a Package from the Registry into your model, you can right-click on the Package name in the Registry Browser and select this option to <b>compare</b> the imported Package with the Registry version, and show any differences.</p> <p>The system automatically selects the model Package from the Project Browser.</p>	<a href="#">Compare an Asset to the Model</a> <sup>[291]</sup>
<b>Find in Project Browser</b>	<p>If you have previously imported a Package from the Registry into your model, you can right-click on the Package name in the <b>Registry Browser</b> and select this option to highlight the corresponding Package in the <b>Project Browser</b>.</p>	

### Notes

- Some of the other options available on the Registry Browser are used to **set up** assets in the Registry
- If the Storage is password-protected, a prompt for you to enter a password displays when you select to perform an operation on the data; in this situation, you need the Read-only password to **process** the information from the Storage, and the All-access password to **change** the information in the Storage

### Learn more

- [Reuseable Asset Service](#) <sup>[282]</sup>
- [Connect to the Asset Service](#) <sup>[283]</sup>
- [Asset Properties](#) <sup>[286]</sup>
- [Package Contents](#) <sup>[286]</sup>
- [Package Dependencies](#) <sup>[288]</sup>
- [Package Technologies](#) <sup>[289]</sup>
- [Storage Files](#) <sup>[290]</sup>
- [Set Up the Asset Service](#) <sup>[293]</sup>
- [Register New Assets](#) <sup>[298]</sup>
- [Update an Asset](#) <sup>[304]</sup>

### 3.5.2.1 Asset Properties

When you select a Storage and a Package in the Registry Browser, the first four tabs in the lower half of the Reusable Asset Service view are updated with information from the Package. The **Asset Properties** tab is a **read-only** view of the properties of the selected **Package** itself.

**Access**    **Project | Reusable Asset Service | <Package name> > Asset Properties**

#### Review Package Properties

Field	Detail	See also
<b>Package</b>	Displays the name of the selected Package.	
<b>GUID</b>	Displays the Global Unique Identifier of the Package.	
<b>Version</b>	Displays the version reference (such as <b>1.1</b> or <b>Beta</b> ) of the Package.	
<b>Author</b>	Displays the user ID of the person who created the Package (the Package Author).	
<b>Created Date</b>	Displays the date and time at which the selected version of the Package was created (that is, registered) <b>in the Registry</b> .	<a href="#">Register New Assets</a> <sup>[298]</sup>
<b>Modified Date</b>	Displays the date and time at which the selected version of the Package was last modified (that is, <b>re-registered</b> ) in the Registry.	<a href="#">Update an Asset</a> <sup>[304]</sup>
<b>Comments</b>	Displays any comment recorded against the Package when it was registered in the Storage.	
<b>Notes</b>	Displays the notes recorded against the Package when it was registered in the Storage.	

#### Learn more

- [Reuseable Asset Service](#)<sup>[282]</sup>
- [Browse Assets](#)<sup>[284]</sup>

### 3.5.2.2 Package Contents

When you select a Storage and a Package in the Registry Browser, the first four tabs in the lower half of the Reusable Asset Service view are updated with information from the Package. The **Contents** tab lists the **diagrams** and **elements** (including child Packages) held in the selected Package, listing the two types of object separately. You can organize the information within a column into alphabetical or reverse-

alphabetical order for ease of reference, and use the Filter bar to filter the display to show only items with values containing specific characters or digits.

Access **Project | Reusable Asset Service > Contents**

#### Review Package Contents

Field/Option	Detail	See also
<b>Content Type</b>	Click on this button to toggle between listing the Package <b>diagrams</b> first and listing the Package <b>elements</b> first.	
<b>Toggle Filter Bar</b>	Right-click on the column headings and select this option to show or hide the <b>Filter Bar</b> on the display.	
<b>&lt;filter bar fields&gt;</b>	Type in the appropriate characters to list only elements and diagrams that have that string of characters in the values in the corresponding column.	<a href="#">List Header</a> <sup>[678]</sup>
<b>Content Type: Diagram</b> <b>Content Type: Element</b>	These are the headers for the two types of object listed in this tab. Click on the appropriate expansion box to hide or show the list of diagrams or elements under the heading.	
<b>Name</b>	Displays the name of the element or diagram available in the Package.	
<b>Type</b>	Displays the type of the element or diagram, such as UseCase or Use Case Diagram.	
<b>GUID</b>	Displays the Global Unique Identifier of the element or diagram.	
<b>Content Type</b>	Displays the object type of the item - <b>Element</b> or <b>Diagram</b> .	
<b>Find in Project Browser</b>	Right-click on an element or diagram line and select this option to see if the element or diagram also exists in your model and, if it does, to highlight it in the Project Browser.	
<b>View Diagram</b>	Right-click on a diagram name and select this option to display the diagram within a labeled frame, as an image. Alternatively, double-click on the diagram name.	

Learn more

- [Reuseable Asset Service](#)<sup>[282]</sup>
- [Browse Assets](#)<sup>[284]</sup>

**3.5.2.3 Package Dependencies**

A Package held as an asset in the Reusable Asset Service registry might contain elements and diagrams that have relationships with objects in other Packages. The Packages containing these 'external' objects will also have been added to the Registry, to support the first Package, and will be listed in the Registry Browser tab along with other, unrelated Packages. You can establish whether the selected Package in the Registry Browser has links to other Packages in the Registry, and which they are, by reviewing the **Dependency** tab.

Package A depends on Package B if any of the following constructs (or their Tagged Values) in Package A references elements in Package B:

- Elements
- Attributes
- Operations
- Operation Parameters
- Diagrams
- Connectors

Access    **Project | Reusable Asset Service > Dependency**

Check Package Dependencies

Field/Option	Detail	See also
<b>Package</b>	Displays the name of the related Package.	
<b>Version</b>	Displays the latest registered version of the related Package that has links to the selected Package.	
<b>GUID</b>	Displays the Global Unique Identifier of the related Package.	
<b>Find in Project Browser</b>	Right-click on the Package line and select this option to see if the Package also exists in your model and, if it does, to highlight it in the Project Browser.	

Notes

- If one Package depends on another, and that second Package itself depends on another Package, both the second **and** third Packages are shown in the Dependency tab

Learn more



- [Reuseable Asset Service](#)<sup>[282]</sup>
- [Browse Assets](#)<sup>[284]</sup>

### 3.5.2.4 Package Technologies

It is possible that the asset Package in the Registry is associated with one or more Technologies, especially if the Package comes from a model developed as an extension or customization of the UML. You can check whether an asset Package has any associated Technologies, and what they are, by viewing the **Technology** tab of the Reusable Asset Service view.

**Access**    **Project | Reusable Asset Service > Technology**

#### Review Technologies

Field/Option	Detail	See also
<b>Name</b>	Displays the name of the Technology as recorded in the Registry.	
<b>Version</b>	Displays the required version of the Technology to support the asset Package, available through the Registry.	
<b>ID</b>	Displays the unique identifier of the Technology.	
<b>Type</b>	Identifies whether the Technology is system-supplied and <b>built in</b> to Enterprise Architect (such as BPMN 2.0 or ArcGIS), or <b>user-defined</b> .  A user-defined technology can be introduced from an external source or developed by a user within the system.	<a href="#">Using MDG Technologies</a> <sup>[1475]</sup>
<b>Registry Status</b>	Indicates whether the Technology file is: <ul style="list-style-type: none"> <li>• <b>Available</b> - the Technology file is registered in the current Storage, or</li> <li>• <b>Unavailable</b> - either: <ul style="list-style-type: none"> <li>• The Technology file is not registered in the Storage, or</li> <li>• The Technology is a <b>built-in</b> Technology and is already integrated with your system, or must be purchased under license</li> </ul> </li> </ul>	
<b>Import From Registry</b>	If your local system does not have one of the <b>Available</b> Technologies, you can quickly import it from the Registry. You cannot import <b>Unavailable</b> Technologies.  Right-click on the Technology name and select this context menu option.  A prompt displays for you to confirm the import to your system. Click on the <b>Yes</b> button.  A message displays to confirm that the Technology has been imported. Click on the <b>OK</b> button.	

Field/Option	Detail	See also
	You might need to restart Enterprise Architect to activate the Technology.	

### Notes

- You can also import **Available** Technologies from the registry as part of the process of importing the asset Package
- If the Storage is **password protected**, a prompt to enter your password displays (this can be a **Read-Only** or a **Complete Access** password) after you confirm that you want to import the Technology; you cannot import the Technology without this password

### Learn more

- [Reuseable Asset Service](#)<sup>[282]</sup>
- [Browse Assets](#)<sup>[284]</sup>
- [Import an Asset into the Model](#)<sup>[291]</sup>
- [Set Up the Asset Service](#)<sup>[293]</sup>

### 3.5.2.5 Storage Files

The **Storage Files** tab of the Reusable Asset Service view simply lists the **files** that have been uploaded to the specified **Storage**. The files can be of any type, uploaded from any directory on any system connected to the registry. Typically they would be document or graphics files that provide standards, guidelines or design information on a model structure. The tab displays the file name and extension, and a comment on the nature or purpose of the file.

Files held in the Registry are compressed, so to view the contents you download them onto your local system.

**Access**    **Project | Reusable Asset Service > Storage Files**

### Review Storage Files

On the Storage Files tab, check the **Comments** field against each file that interests you.

To view the contents of a file, right click on the file name and select the **Import From Registry** context menu option. The Save As browser displays, through which you select the directory into which to copy the file.

Click on the **Open** button. The file is copied into the selected location, from which you can open it.

### Notes

- If the **Storage** is **password-protected**, the system prompts you to enter a password before you begin importing a file from the Storage; in this case you require either a **Read-Only** or **Complete Access** password

#### Learn more

- [Reuseable Asset Service](#)<sup>[282]</sup>
- [Browse Assets](#)<sup>[284]</sup>
- [Set Up the Asset Service](#)<sup>[293]</sup>

### 3.5.2.6 Compare an Asset to the Model

If you are developing a Package in your model against a standard structure, or using common elements from the Reusable Asset Service, you can check that your model conforms to the standard or incorporates any changes to the common elements, by performing a comparison between the asset Package and your model.

**Access** **Project | Reusable Asset Service > Registry Browser | right-click Package | Compare with Package in Model**

#### Compare Asset Package and Model

A prompt displays for you to confirm the comparison. Click on the **Yes** button.

The Baseline Comparison view displays, showing the element hierarchy in which differences have been detected between the Package in the model and the asset Package in the Registry (as the Baseline).

All facilities of the standard Baseline Comparison are available, including the ability to 'roll back' differences so that the element or Package in the model matches the Registry.

#### Notes

- If the **Storage** is **password-protected**, the system prompts you to enter a password before you begin comparing information in the Storage and in the model; in this case you require either a **Read-Only** or a **Complete Access** password

#### Learn more

- [Reuseable Asset Service](#)<sup>[282]</sup>
- [Package Baselines](#)<sup>[457]</sup>
- [Example Comparison](#)<sup>[468]</sup>

### 3.5.3 Import an Asset into the Model

The Reusable Asset Service provides common or standard information and data that you can import into your local model. The unit that you import is a selected version of the Asset Package; you can also, optionally, import:

- Other Packages on which the selected Asset Package depends, and/or
- Available Technologies that support full use of the model structures in the Asset Package

The Asset Package (with, if selected, any needed Packages) is imported into the **currently-selected** model Package in the Project Browser, **unless** the Asset Package already exists anywhere else in the **Project**. In this case, the system locates the existing Package and overwrites it with the imported Asset Package.

**Access** **Project | Reusable Asset Service | Click on Asset Package: Import** or

### Project | Reusable Asset Service > Registry Browser | Right-click on Asset Package | Import from Registry

#### Import Asset Package

Before selecting an Asset Package to import into your model, click on the **Version** drop-down arrow and select the appropriate version of the Package to import.

When you select to import an Asset Package into your model, a short menu displays from which you select to import:

- The Package alone or
- The Package with the Packages on which it is dependent

A prompt then displays for you to confirm the import, and to warn you that if the Package already exists in the model, it will be overwritten by the import. Click on the **Yes** button to continue, or the **No** button to cancel the import.

If the Storage is **password protected**, after you confirm the import a prompt displays for your Read-Only or Complete-Access password; enter this and click on the **OK** button.

The Import from Registry dialog displays, followed by confirmation that the import of the Package or Packages is complete. Click on the **OK** button.

#### If the Package depends on Technologies

If the Technologies associated with an Asset Package are not already on your system and/or enabled in your model, during the display of the Import from Registry dialog the Import Technology From Registry dialog also displays. Process this dialog as described.

Field/Option	Detail	See also
<b>Name</b>	Displays the name of the Technology as recorded in the Registry.  The check box against each name is selected by default. If you want to import one technology but not another, <b>deselect</b> the checkbox of the technology to ignore.	
<b>Version</b>	Displays the required version of the Technology to support the Asset Package, available through the Registry.	
<b>Registry Status</b>	Indicates whether the Technology file is: <ul style="list-style-type: none"> <li>• <b>Available</b> - the Technology file is registered in the current Storage, or</li> <li>• <b>Unavailable</b> - either: <ul style="list-style-type: none"> <li>• The Technology file is not registered in the Storage, or</li> <li>• The Technology is a <b>built-in</b> Technology and is already integrated with your system, or must be purchased under license</li> </ul> </li> </ul>	

Field/Option	Detail	See also
<b>Model Status</b>	Indicates whether the Technology is already in your model and disabled.  (If the Technology is in your model and enabled, there is no need to import it and it is not listed in the dialog.)	
<b>OK</b>	Click on this button to import and/or enable the selected Technology or Technologies in your system.  A status message displays to indicate that the Package and Technologies have been imported and the Technologies enabled. You might have to restart Enterprise Architect for the Technologies to take full effect.  Click on the <b>OK</b> button to clear the message.  In the Project Browser, the Package has been added to the model with (if selected) the Packages on which it is dependent.	

**Notes**

- You can also import Technologies separately from the Package that is dependent on them, using the Technology Tab of the Reusable Asset Service view

**Learn more**

- [Reuseable Asset Service](#) <sup>[282]</sup>
- [Register New Assets](#) <sup>[298]</sup>
- [Package Technologies](#) <sup>[289]</sup>

**3.5.4 Set Up the Asset Service**

The process of setting up assets and files in the Reusable Asset Service has a number of simple stages, typically:

- Identify the Registry
- Create the Storages, including copying an existing Storage as template for a new one
- Set password protection on each Storage
- Register the Asset Packages and Storage Files, and update them as necessary

**Access Project | Reusable Asset Service****Identify Registry**

The **Registry server** holds the **Storages** of the Reusable Asset Service. It will have been previously configured by your system administrator, to be accessed through a **Cloud Connection**. Your system administrator will provide the connection details for you to use.


In the Reusable Asset Service view, click on the [ ... ] Browse button to the right of the **Registry** field. The Cloud Connection dialog displays.

Using the information provided by your administrator, type in the server name, URL and the name of the model hosting the registry data. If advised by your administrator, also type in the user name and password for the server.

Click on the **OK** button; the Registry field now shows the server name.

### Create Storages

A Storage is a container within the Registry, holding related assets and files. There can be several Storages in the Registry, to support the different areas of work or purposes that your organization might define.

Step	Action	See also
1	Click on the  icon in the Reusable Asset Service toolbar. The New Storage dialog displays.	
2	In the <b>Name</b> field, type a name that indicates the purpose or content of the Storage.	
3	In the <b>Type</b> field, select: <ul style="list-style-type: none"> <li>• <b>Draft</b> if you intend to make significant changes to the content of the Storage, such as populating it with assets and/or files; you <b>can</b> overwrite existing Packages or files in a Draft Storage</li> <li>• <b>Complete</b>; if the Storage is ready for use; you <b>cannot</b> overwrite any Packages or files in a Complete Storage, although you can delete Packages and files and add new ones</li> </ul> <p>You can modify a Storage at a later time, and change this setting.</p>	<a href="#">Change Storage Details</a> <sup>[296]</sup>
4	If you want to set <b>Storage Access Passwords</b> on this Storage, click on the check box against one or both password types, and provide and confirm the <b>Complete Access</b> and/or <b>Read-Only Access</b> passwords.	<a href="#">Password Protection</a> <sup>[294]</sup> (below)
5	Click on the <b>OK</b> button. A prompt displays for you to confirm the creation of the Storage, with the type of access you have set on the Storage.	
6	Click on the <b>Yes</b> button. A confirmation message displays; click on the <b>OK</b> button. The name of the new Storage displays in the <b>Storage</b> field.	

### Password Protection

Access to a Storage is unrestricted, but you can apply password protection to restrict users in processing and modifying the contents of the Storage either intentionally or unintentionally. Password protection can be applied at two levels:

- **Complete Access** - users enter their password and can modify or delete the Storage itself, and within the Storage can register, update and delete Packages and files in the Registry, view Packages and import assets into their models
- **Read-Only Access** - users enter their password and can import assets into their models

You set the **Complete Access** password **before** you set the **Read-Only Access** password.

If password protection is set and a user has no password, they can only view the contents of the Storage. If **no** password protection is set, all users can freely access and modify the Storage and its contents.

You set the passwords on each Storage as you create that Storage, using the New Storage dialog. Once the Storage has been created, you can change or remove an existing password using the Modify Storage Access dialog, but you cannot **add** a password.

If a Storage is password protected, the system displays a prompt for the password when the user begins to:

- Change or delete a password on the Storage
- Copy the Storage to create a new Storage
- Delete the Storage
- Register a Package or file in the Storage
- Import a Package, file or Technology from the Storage
- Delete a Package or file from the Storage
- Compare an Asset Package with a Package in the model

The system displays the password prompt when any of those operations is performed for the **first time** after the user:

- Connects to a Registry
- Reloads the Registry Browser
- Selects a Storage in the **Storage** field in the Reusable Asset Services view

If the user enters the correct password and therefore establishes their credentials, the system does not display a password prompt again for any valid operation that user performs until they:

- Select a different Storage in the **Storage** field in the Reusable Asset Services view or
- Reload the Registry Browser from the Registry.


### Register Assets and Files

Whilst a Storage exists and, if the Storage is under password protection, you have the **Complete Access** password, you can register Packages from a model and files from your system as assets in that Storage. If the Asset Package was developed using one or more MDG Technologies, you can optionally register those technologies in the Storage as well.

See the *Register New Assets* Help topic.

### Copy Storages

It is possible to copy a Storage as the basis for creating another Storage, for example if you want to use the same set of assets in the context of larger set for a different department, work area or development phase.

Step	Action	See also
1	In the <b>Storage</b> field, select the Storage to be copied.	
2	Click on the  icon in the Reusable Asset Service toolbar.	
3	If necessary, enter your <b>Complete Access</b> password and click on the <b>OK</b> button. The Save Storage As dialog displays.	
4	In the <b>Name</b> field, type a name that indicates the purpose or content of the new Storage.	
5	In the <b>Type</b> field, select: <ul style="list-style-type: none"> <li>• <b>Draft</b> if you intend to make significant changes to the content of the Storage, such as populating it with assets and/or files; you <b>can</b> overwrite existing Packages or files in a Draft Storage</li> <li>• <b>Complete</b>; if the Storage is ready for use; you <b>cannot</b> overwrite any Packages or files in a Complete Storage, although you can delete Packages and files and add new ones</li> </ul> <p>You can modify a Storage at a later time, and change this setting.</p>	<a href="#">Change Storage Details</a> <sup>[296]</sup>
6	If you want to set <b>Storage Access Passwords</b> on this Storage, click on the check box against one or both password types, and provide and confirm the <b>Complete Access</b> and/or <b>Read-Only Access</b> passwords.	<a href="#">Password Protection</a> <sup>[294]</sup> (above)
7	Click on the <b>OK</b> button. A prompt displays for you to confirm the creation of the Storage, with the type of access you have set on the Storage.	
8	Click on the <b>Yes</b> button. A confirmation message displays; click on the <b>OK</b> button. The name of the new Storage displays in the <b>Storage</b> field.	

### Change Storage Details


After you have set up a Storage, you can update it at a later stage to:

- Change the status, or **Type**,
- Change one or both of the existing **Complete Access** and **Read-Only Access** passwords
- Delete one or both of the passwords




It is not possible to **add** a password where one has not previously been set.

You cannot change or delete passwords unless you have the **Complete Access** password yourself.

Step	Action	See also
1	On the Reusable Asset Service view, in the <b>Storage</b> field click on the drop-down arrow and select the Storage to update.	
2	Click on the  icon in the Reusable Asset Service view toolbar.	
3	If necessary, enter your <b>Complete Access</b> password and click on the <b>OK</b> button. The Modify Storage Access dialog displays.	
4	In the <b>Type</b> field, click on the drop-down arrow and select: <ul style="list-style-type: none"> <li>• <b>Complete</b> if the Storage has now been set up and is ready for use; you cannot overwrite any Packages or files in the Storage, although you can delete them and add new Packages and files</li> <li>• <b>Draft</b> if you intend to make significant changes to the content of the Storage; you can overwrite existing Packages or files</li> </ul>	
5	The <b>Action</b> field defaults to <b>None</b> , to indicate no changes to the passwords. If you are changing or deleting the passwords, click on the drop-down field and select: <ul style="list-style-type: none"> <li>• <b>Change Password</b> to enable both password type fields ready for editing</li> <li>• <b>Remove Storage Access Password</b> to simply delete the Complete Access password altogether</li> <li>• <b>Remove Read-Only Access Password</b> to simply delete the Read-Only Access password altogether</li> </ul>	<a href="#">Password Protection</a> <sup>[294]</sup>
6	If you have selected the <b>Change Password</b> option, select the appropriate password checkbox or both checkboxes, and type in the new password(s). Re-type the password(s) in the <b>Confirm Password</b> field(s).	
7	Click on the <b>OK</b> button to save your changes and close the Modify Storage Access dialog.	

### Delete a Storage

If a Storage has been created by mistake, or is no longer necessary in the Registry, you can delete it and its entire contents of Package and files in one action.

Step	Action	See also
1	On the Reusable Asset Service view, in the <b>Storage</b> field click on the drop-down arrow and select the Storage to delete.	
2	Click on the  icon in the Reusable Asset Service view toolbar. A prompt displays to confirm the deletion.	
3	Click on the <b>Yes</b> button. If necessary, enter your <b>Complete Access</b> password and click on the <b>OK</b> button. A message displays to confirm that the Storage has been deleted.	
4	Click on the <b>OK</b> button to clear the message. The Storage and its contents have been cleared from the Registry.	

#### Learn more

- [Reuseable Asset Service](#)<sup>[282]</sup>
- [Connecting to Projects Via the Cloud](#)<sup>[262]</sup>
- [Register New Assets](#)<sup>[298]</sup>
- [Update an Asset](#)<sup>[304]</sup>
- [Browse Assets](#)<sup>[284]</sup>

#### 3.5.4.1 Register New Assets

After you have set up your Storages in the Reusable Asset Service Registry, you can register assets in them. The assets include:

- Packages of modeling structures, held in the Project Browser of your model; these Packages can include diagrams and subordinate Packages
- Any Packages containing model components on which the primary Asset Packages depend
- Any MDG Technologies that you used in developing the Asset Packages, and that you decide to register with the Packages
- Any text or graphics files you want to make available to your corporate community; these files are compressed before being added to the Storage

You register the first three items together. You can register files in the Storage during that same process, or separately.

You can register separate versions of the same Package. If you register a Package that already exists in the Registry under the **same** version reference, and the Storage has **Draft** status, the existing version is **overwritten**. If you register the same package with a **different** version number or reference, it is registered **separately** and users can access **both** versions of the Package from the Registry.

**Access** **Project | Reusable Asset Service** and  
**View | Project Browser** (Alt+0)

### Register Package as Asset

Step	Action	See also
1	In the <b>Reusable Asset Service</b> view, select the appropriate Registry and Storage.	<a href="#">Set Up the Asset Service</a> <sup>[293]</sup>
2	In the <b>Project Browser</b> for your model, click on the Package to register as an Asset.	
3	On the <b>Registry Browser</b> tab, click on the <b>Register</b> button.  If the Storage is password protected, a prompt displays for your <b>Complete Access</b> password. Enter this and click on the <b>OK</b> button.	<a href="#">Set Up the Asset Service</a> <sup>[294]</sup>
4	<p>The Register Package(s) dialog displays, showing the:</p> <ul style="list-style-type: none"> <li>Name and type of the Storage</li> <li>Name and status of the selected Package (the status at this point being <b>Pending</b>)</li> <li>The name, GUID and current version of the selected Package; you can type a different version number or text string if necessary</li> </ul> <p>If there is a version of the Package already in the Registry, and the Storage has <b>Draft</b> status, you can either:</p> <ul style="list-style-type: none"> <li>Overwrite the registered Package with an update of that version, or</li> <li>Add the updated Package as another Asset under a different version number; in this case, the Packages to Register panel shows the version number, registration date and comments on the latest version of the Asset Package</li> </ul> <p>You cannot overwrite an existing Package if the Storage has <b>Complete</b> status. In this case, add the Package under a different version number.</p>	
5	In the <b>Comments</b> field type a short comment and, in the <b>Notes</b> field, a longer explanation of the Package being registered.	
6	Click on the <b>Check Dependency</b> button and resolve the dependencies of the Package.  At the end of this process, the selected Package has the status <b>Ready</b> (for registration).	<a href="#">Check Package Dependencies</a> <sup>[302]</sup>

Step	Action	See also
7	<p>You can, at this point, also register <b>files</b> as assets of the Storage. See Step 4 of the <i>Register File as Asset</i> procedure below.</p> <p>However, if you prefer you can follow the complete procedure separate from registering Packages.</p>	
8	<p>Click on the <b>Register</b> button.</p> <p>A prompt displays to confirm registering the package. Click on the <b>Yes</b> button.</p> <p>If the <b>same</b> version of the selected Package already exists in the Registry (if the Storage is a <b>Draft</b>) a prompt displays to confirm overwriting it. Click on the <b>Yes</b> button.</p> <p>A progress dialog displays. If the Package does <b>not</b> have any associated MDG Technologies, a message displays confirming that Package registration is complete. Go to step <b>10</b>.</p>	
9	<p>If the Package has one or more associated MDG Technologies, the Register MDG Technology dialog displays, listing the Technologies and prompting you to register (<b>OK</b>) or <b>Ignore</b> all of them.</p> <p>If you want to register some Technologies but not others, click on those you do not want to register, and click on the <b>Remove</b> button for each one in turn.</p> <p>Click on the <b>OK</b> or <b>Ignore</b> button.</p> <p>A message displays confirming that Package and Technology registration is complete.</p>	
10	<p>Click on the <b>OK</b> button.</p> <p>The Registry Browser tab is updated with the newly-added details of the Package and any dependent Packages. You can review the details on the tabs on the lower half of the Reusable Asset Server view.</p>	<a href="#">Browse Assets</a> <sup>[284]</sup>

### Register File as Asset

Step	Action	See also
1	In the <b>Reusable Asset Service</b> view, select the appropriate Registry and Storage.	
2	<p>On the <b>Registry Browser</b> tab, click on the <b>Register</b> button.</p> <p>If the Storage is password protected, a prompt displays for your <b>Complete Access</b> password. Enter this and click on the <b>OK</b> button.</p> <p>The Register Package(s) dialog displays.</p>	<a href="#">Set Up the Asset Service</a> <sup>[293]</sup>

Step	Action	See also
3	If any Packages are listed in the Packages to Register panel, and you do not want to re-register them, clear the checkbox against each Package name.	
4	Click on the <b>Additional Files</b> button. The Select Additional Files dialog displays.	
5	Click on the <b>Add</b> button. A Select screen displays, on which you browse for and click on the required file. You can only select one file at a time on this browser.	
6	Click on the <b>Open</b> button. The Add Comment dialog displays.	
7	Type a brief comment on the nature or purpose of the file in the Registry, and click on the <b>OK</b> button. Focus returns to the Select Additional Files dialog, which now lists the selected file.	
8	Repeat steps 5-7 for each Asset file to register. When you have finished, click on the <b>OK</b> button. The Register Package(s) dialog redisplays.	
9	Click on the <b>Register</b> button. A prompt displays to confirm that you want to register the files.	
10	Click on the <b>Yes</b> button. A Progress dialog displays while the files are registered, followed by a message confirming that the (Package and) File registration is complete.	
11	Click on the <b>OK</b> button to clear the message, and to return focus to the Reusable Asset Service view. You can review the files on the Storage Files tab in the lower half of the view.	<a href="#">Browse Assets</a> <sup>[284]</sup> <a href="#">Storage Files</a> <sup>[290]</sup>

### Learn more

- [Reuseable Asset Service](#) <sup>[282]</sup>

- [Browse Assets](#) <sup>[284]</sup>

#### 3.5.4.1.1 Check Package Dependencies

As you register a Package in the Reusable Asset Service, the system provides a check on whether that Package is **dependent on** any external content; that is, whether it links to any modeling component held in Packages **other than** its own child Packages. Package A is dependent on Package B if Package A contains any of these constructs (or their Tagged Values) that reference elements in Package B:

- Elements
- Attributes
- Operations
- Operation parameters
- Diagrams
- Connectors

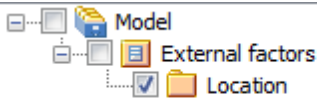
You perform this dependency check as an integral part of the Asset Registration process (specifically, at step 6 of the *Register Package as Asset* process). The check automatically identifies, firstly, the Packages directly needed by the Asset Package, and then any Packages needed by **those** initial Packages, and so on until the needed Packages are not dependent on any other.

In practice, such dependency chains are likely to be quite short. If you know that the original Asset Package is dependent on two other Packages, one of which is subordinate to the other, you can simplify the process further by selecting the Parent or Ancestor package first and thus avoiding the extra steps of selecting the two Packages separately.

**Access**    **Project | Reusable Asset Service > Registry Browser: Register**

#### Check Package Dependencies

Step	Action	See also
1	Select the appropriate Registry, Storage and version of the model Package to register, and provide any appropriate comments or notes.	<a href="#">Register New Assets</a> <sup>[298]</sup> to Step 6.
2	<p>Click on the <b>Check Dependency</b> button.</p> <p>If there are no Packages on which the Asset Package depends, the system sets that Package to <b>Ready</b>. Go to step 6.</p> <p>The Select Needed Package: &lt;dependent package name&gt; dialog displays, showing the model hierarchy containing the Packages needed by the Asset Package. Each needed Package has a selected checkbox against it.</p> <p>If you know that a needed Package is a child or grandchild of another needed Package, click on the checkbox next to the 'higher' Package name so that the child is validated as part of the parent. In this example, if the Asset Package has dependencies on both <b>External factors</b> and <b>Location</b>, click on the <b>External factors</b> check box to register both dependencies as <b>External factors</b>.</p>	

Step	Action	See also
		
3	<p>Click on the <b>OK</b> button.</p> <p>The Register Package(s) dialog updates to show the initial Asset Package status as <b>Ready</b>, and to list the needed Packages each with the status of <b>Pending</b>.</p>	
4	<p>Click on the <b>Check Dependency</b> button again.</p> <p>If there are no Packages on which the needed Package depends, the system sets that Package to <b>Ready</b> and sets the focus on the next <b>Pending</b> Package. If there are no more <b>Pending</b> Packages, go to step 6.</p> <p>If the needed Package is dependent on other Packages, the Select Needed Package: &lt;dependent package name&gt; dialog displays again, showing the model hierarchy containing <b>those</b> Packages, each with a selected checkbox next to it. As in step 2, you can select a 'higher' Package if appropriate.</p>	
5	<p>Click on the <b>OK</b> button.</p> <p>The Register Package(s) dialog updates to show the first needed Package status as <b>Ready</b>, and to list any further needed Packages with the status of <b>Pending</b>.</p>	
6	<p>If there are further needed Packages in the chain, repeat step 4. The system automatically identifies them and lists them for validation as in step 5.</p> <p>Otherwise the system sets the final needed Package status to <b>Ready</b> and you can continue to register files and MDG Technologies.</p>	<a href="#">Register New Assets</a> <sup>[298]</sup> from Step 7.

### Notes

- A Package with status **Pending** cannot be registered; only Packages with status **Ready** can be registered
- If you are certain that there are no parent-child relationships between the needed Packages, you can work through these steps quickly by just clicking on the **Check Dependency** and **OK** buttons alternately until all Packages in the **Package to Register** list have the status **Ready**
- The system does not allow you to register an Asset Package that has dependencies on its own parent or grandparent Package; such dependencies would cause the Asset Package to be registered twice - by itself and as part of the parent Package - creating the risk of differences developing between the two instances

### Learn more

- [Reuseable Asset Service](#) <sup>[282]</sup>

- [Browse Assets](#) <sup>[284]</sup>

### 3.5.4.2 Update an Asset

Over time, you might need to revise the contents of the Registry. Such updates can include:


- Registering additional Asset Packages and files
- Registering a new version of an Asset Package
- Registering a new edition of a Storage file
- Deleting an Asset Package
- Deleting a Storage file
- Changing the details of a Storage
- Deleting an entire Storage and the assets and files within it

**Access** [Project | Reusable Asset Service > Registry Browser](#)

#### Update Tasks

Task	Detail	See also
<b>Register new Asset Packages and Storage files</b>	You can add further Asset Packages and files to a Storage at any time.	
<b>Register a new version of an Asset Package</b>	<p>You register a <b>new version</b> of an Asset Package as if it were a new Asset. In the <b>Current Version</b> field on the Register Package(s) dialog, you enter a different number or reference to that assigned to the currently-registered asset.</p> <p>You can overwrite an existing Asset Package under the <b>same</b> version number only if the Storage has <b>Draft</b> status. If the Storage has <b>Complete</b> status, you cannot overwrite the existing Asset Package under the same version number. To replace the <b>current</b> version, you must delete the Asset from the Storage (see below) and register the Package again.</p>	<a href="#">Register New Asset</a> <sup>[298]</sup>
<b>Delete Asset Package from Storage</b>	<p>Click on the Package name in the Registry Browser tab and either:</p> <ul style="list-style-type: none"> <li>• Click on the <b>Delete</b> button or</li> <li>• Right-click and select the <b>Delete From Registry</b> menu option</li> </ul> <p>In either case a short submenu displays, prompting you to select to delete:</p> <ul style="list-style-type: none"> <li>• The Asset Package only, or</li> <li>• The Asset Package and the Packages on which it is directly or indirectly dependant</li> </ul> <p>When you select the appropriate option, a prompt displays for you to confirm the deletion.</p> <p>Click on the <b>Yes</b> button. A message displays to confirm that the</p>	

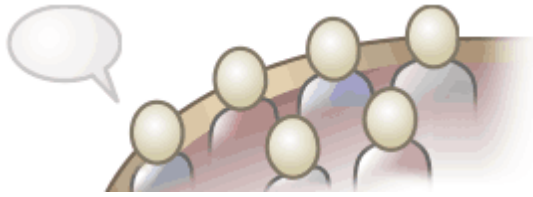


Task	Detail	See also
	<p>Package has been deleted from the Registry, and the name of the Package is removed from the Registry Browser tab.</p> <p>If the <b>Storage</b> is <b>password-protected</b>, the system prompts you to enter a <b>Complete Access</b> password before it begins deleting the Package (s).</p>	
<b>Register a new version of a Storage File.</b>	<p>You register a <b>new version</b> of a Storage file as if it were a new file under a different file name.</p> <p>You can overwrite an existing file under the <b>same</b> file name only if the Storage has <b>Draft</b> status. If the Storage has <b>Complete</b> status, you cannot overwrite the existing file. To replace the <b>current</b> version, you must delete the file from the Storage (see below) and register the file again.</p>	<a href="#">Register New Asset</a> <sup>[298]</sup>
<b>Delete Storage Files from Storage</b>	<p>On the Storage Files tab, right-click on the file name and select the <b>Delete From Registry</b> context menu option. A prompt displays for you to confirm the deletion.</p> <p>Click on the <b>Yes</b> button. A status message confirms the deletion, the file is removed from the Registry and the file name is removed from the Storage Files tab.</p> <p>If the <b>Storage</b> is <b>password-protected</b>, the system prompts you to enter a <b>Complete Access</b> password before it begins deleting the file.</p>	
<b>Change Storage Details</b>	<p>These tasks are as documented for setting up the Registry. Note that whilst you can <b>copy</b> a Storage <b>as a new</b> Storage, you cannot <b>transfer</b> assets directly between existing Storages.</p>	<a href="#">Setup Asset Service</a> <sup>[293]</sup>
<b>Delete a Storage and its contents</b>		
<b>Refresh Registry Browser</b>	<p>If there is a possibility that another user has updated the Registry while you have had it open, click on the  icon in the Reusable Asset Service toolbar to refresh your display from the latest updates to the Registry.</p>	

#### Learn more

- [Reuseable Asset Service](#) <sup>[282]</sup>
- [Browse Assets](#) <sup>[284]</sup>
- [Storage Files](#) <sup>[290]</sup>

## 3.6 Team Development



Using Enterprise Architect, you can develop a project across a team of people so that each person can access the latest data without the risk of damaging or losing that data. The system provides a number of facilities for managing, accessing and discussing the model data within the team.

### Making project data available in a distributed environment

Enterprise Architect offers a diverse set of functionality designed specifically for sharing projects in team-based and distributed development environments; for example: **network deployment** of model repositories, **replication** and **XMI Import/Export**.

### Applying security to the model

**User Security** is a means of improving collaborative design and development by preventing concurrent editing, and limiting the possibility of inadvertent model changes by users not designated as model authors.

### Using an internal discussion forum

The **Project Team Review** facility is a **discussion forum** within your development team community to discuss the development and progress of a project. You can link to the model elements that are the subject of a post thread in the forum.

Similarly, you can develop a discussion about a specific element in the model as a component of the element, using the **Element Discussions** facility.

### Building company policy and project development guidelines into the project

You can create **workflow scripts** that provide a robust approach to applying company policy and strengthening project development guidelines, by validating against the policy and procedures within the model itself.

### Standardizing and re-using project data

You can import and export **Reference data** (including Glossary and Issue information) from .XML files of another iteration of the **same** model, or of a **different** model.

### Learn more

- [Project Sharing](#)<sup>[307]</sup>
- [User Security](#)<sup>[316]</sup>
- [Team Review Tools](#)<sup>[343]</sup>
- [Element Discussions](#)<sup>[365]</sup>

- [Workflow Scripts](#) <sup>[367]</sup>
- [Sharing Reference Data](#) <sup>[374]</sup>

### 3.6.1 Project Sharing

Enterprise Architect offers a diverse set of functionality designed specifically for sharing projects in team-based and distributed development environments, through network deployment of model repositories, replication and XML Import/Export.

#### Topics

Topic	Detail	See also
<b>Network Deployment</b>	<p>Network deployment is possible under two different schemas:</p> <ul style="list-style-type: none"> <li>• File based repositories or</li> <li>• DBMS server based repositories</li> </ul> <p>DBMS server based repositories offer better:</p> <ul style="list-style-type: none"> <li>• Response times than files on networks, due to the inherent structure of the DBMS</li> <li>• Solutions when networking problems are encountered, as they have the ability to backtrack transactions caused by external breakdowns</li> </ul>	<a href="#">Share a Project</a> <sup>[308]</sup>
<b>Replication</b>	<p>Replication enables data interchange between .EAP based repositories and is suitable for where many different users work independently in parallel development.</p> <p>Modelers merge their changes into a Design Master only as required; it is recommended that a backup is carried out prior to replication.</p> <p>Replication cannot be performed on repositories stored on a DBMS server.</p>	<a href="#">Replication</a> <sup>[310]</sup>
<b>XML Import and Export</b>	<p>XML Import/Export can be used to export and share discrete packages between developers; XML enables the export of packages into XML files which can then be imported into any model.</p> <p>Package control can be used to set up packages for version control and to enable batch export of packages using XML; Version Control enables a repository to be maintained by a third-party source code control application that is used to control access and record revisions.</p>	<a href="#">XML Import/Export</a> <sup>[473]</sup> <a href="#">Version Control</a> <sup>[383]</sup> <a href="#">Controlled Packages</a> <sup>[484]</sup>

#### Notes

- DBMS Repository support is available with the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect

### Learn more

- [Deployment of Enterprise Architect white paper](#)

### 3.6.1.1 Share Enterprise Architect Projects

The most efficient way of using Enterprise Architect to manage a team development is to *share* a project amongst a team of designers, developers and analysts.

### Guide


Topic	Detail	See also
<b>By Sharing a project</b>	<ul style="list-style-type: none"> <li>• Many people can work on the model at the same time and contribute their particular skill</li> <li>• Team members can always see what the latest changes are, keeping the team informed and up to date with the project status</li> </ul>	<a href="#">Refresh View of Shared Project</a> <sup>[308]</sup>
<b>You can share a project in three ways</b>	<ul style="list-style-type: none"> <li>• Place the project in a shared network directory</li> <li>• Use replication</li> <li>• Use a shared DBMS-based repository</li> </ul>	<a href="#">Share Projects on Network Drive</a> <sup>[309]</sup> <a href="#">Replication</a> <sup>[310]</sup> <a href="#">Server Based Repositories</a> <sup>[214]</sup>

### Notes

- Project Sharing and Replication are available in the Professional, Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect
- DBMS repositories are supported in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect

#### 3.6.1.1.1 Refresh View of Shared Project

When a user of a shared model checks out a package and makes changes, other users can see those changes by refreshing their view of the package or the changed diagram within the package, in a number of ways.

Refresh	Detail	See also
<b>Project</b>	<ul style="list-style-type: none"> <li>• Right-click on the package name in the Project Browser and select the <b>Contents   Reload Current Package</b> context menu option</li> <li>• Select the <b>File   Reload Project</b> menu option</li> <li>• Select the  (<b>Reload Project</b>) icon in the <b>Project</b> toolbar</li> <li>• Press ( <b>Ctrl+Shift+F11</b> )</li> <li>• Close the project and reopen it</li> </ul>	<a href="#">Contents Submenu</a> <sup>[655]</sup> <a href="#">File Menu</a> <sup>[79]</sup> <a href="#">Project Toolbar</a> <sup>[136]</sup>

Refresh	Detail	See also
<b>Diagram</b>	<ul style="list-style-type: none"> <li>Select the <b>Window   Reload Current View</b> menu option</li> <li>Right-click on the opened diagram tab in the diagram view, and select the <b>Reload &lt;diagram name&gt;</b> context menu option</li> </ul>	<a href="#">Window Menu</a> <sup>[127]</sup> <a href="#">Diagram Tabs</a> <sup>[790]</sup>

### 3.6.1.2 Share Projects on Network Drive

The easiest way to share a project amongst a work group of developers and analysts is to place the project file on a shared network drive, to which people connect from their workstations.

Individual developers and analysts can then open and work on the project concurrently.

Network Issues	See also
<p>Enterprise Architect accepts a number of concurrent connections without issue; however:</p> <ul style="list-style-type: none"> <li>There can be occasional 'lock-outs' when one user tries to access or update something another user is in the process of modifying</li> <li>Changes to the Project Browser (and other project views) are not automatically updated; to compensate for this, users must occasionally reload their project to see changes made by other users</li> <li>If two or more people work on the same diagram concurrently, unexpected results can occur; it is best to allow only one analyst to work on a diagram at a time</li> <li>(.EAP files only) If a user's machine crashes, the network suffers an outage or a machine is turned off unexpectedly, the project file might require repair to compensate for the sudden inconsistency; a repair facility is provided (<b>Tools   Data Management   Manage .EAP File   Repair .EAP File</b>) to carry out this task</li> </ul>	<a href="#">Refresh Model View</a> <sup>[308]</sup>  <a href="#">Repair a Project</a> <sup>[602]</sup>

#### Notes

- Firebird-based projects (.feap files) are not suitable for sharing over a network

### 3.6.1.3 Distributed Development

Enterprise Architect supports distributed development using two different techniques.

Topic	Detail	See also
<b>Replication</b>	Enables geographically separated analysts to update and modify parts of the model in replicas, then merge these back together at a central location.	<a href="#">Replication</a> <sup>[310]</sup>
<b>XMI Import/Export</b>	Use XMI-based import/export to export discrete packages to XML to share among the development team; this has several	<a href="#">Import/Export</a> <sup>[473]</sup>

Topic	Detail	See also
	<p>benefits over replication:</p> <ul style="list-style-type: none"> <li>You can assemble a model from only the parts necessary to get your job done</li> <li>You can assemble a full model if required</li> <li>You can assemble a model from different package versions for different purposes (such as customer visible, internal release only)</li> <li>You can roll-back parts of a model as required</li> <li>There is less chance of 'collisions' between developers if each works on a discrete package</li> <li>The process is controllable using a version control system, or through package control</li> </ul> <p>XMI based import/export is UML1.3 / XMI1.1 compliant; you can also write XML based tools to manipulate and extract information from XML files to enhance the development process.</p> <p>XMI-based Import/Export is accessed through <b>Project   Model Import/Export</b>.</p>	<p><a href="#">Version Control</a> <sup>[383]</sup></p> <p><a href="#">Controlled Packages</a> <sup>[484]</sup></p> <p><a href="#">XMI Import Export Options</a> <sup>[90]</sup></p>

### 3.6.1.4 Replication

Apart from sharing Enterprise Architect projects in real time over a network, you can also share projects using replication.

#### Guide

Topic	Detail	See also
<b>Replication</b>	<p>Replication is a powerful means of sharing projects between isolated or mobile users; it enables different users to work independently of one another, and to merge their changes at a later time.</p> <p>In Replication:</p> <ul style="list-style-type: none"> <li>A project is converted to a design master, then replicas are made of the master</li> <li>Users take the replicas away, modify the project, then bring their replicas back to be synchronized with the master file</li> </ul>	
<b>Use Replication</b>	<p>To use replication</p> <ol style="list-style-type: none"> <li>Convert the base project to a design master.</li> <li>Create replicas from the design master.</li> <li>Take the replica away and work on it as required, then bring it back for synchronization with the design master.</li> <li>Synchronize the replicas, during which all changes to both the</li> </ol>	<p><a href="#">Design Master</a> <sup>[311]</sup></p> <p><a href="#">Create Replicas</a> <sup>[312]</sup></p> <p><a href="#">Synchronize Replicas</a> <sup>[313]</sup></p>

Topic	Detail	See also
	master and the replica are propagated in both directions, so they both finally contain the same information.	
<b>Enterprise Architect Merge Rules</b>	<p>Enterprise Architect follows these rules in merging:</p> <ul style="list-style-type: none"> <li>• Additions are cumulative; that is, two replicas each creating three new Classes result in six new Classes after merging</li> <li>• Deletions prevail over modifications; if one replica changes a Class name and other deletes the Class, merging replicas results in both files losing the Class</li> </ul> <p>Conflicting modifications appear in the Resolve Replication Conflicts dialog.</p>	<a href="#">Resolve Conflicts</a> <sup>[315]</sup>
<b>Upgrades and Replicas</b>	When you upgrade your version of Enterprise Architect, you must not open a replica until you have opened the design master and then synchronized the replicas with the master; you cannot directly upgrade a replica.	<a href="#">Upgrade Replicas</a> <sup>[315]</sup>
<b>Avoid Change Collisions</b>	<p>If two or more people make changes to the same element, Enterprise Architect arbitrarily overwrites one person's change with the other's; to avoid this, different users should work on different packages.</p> <p>However, since Enterprise Architect does not enforce this rule, it is possible for users' work to conflict; to minimize difficulties, please note the following guidelines:</p> <ul style="list-style-type: none"> <li>• If users are likely to have worked in the same area of the model, they should both witness the synchronization and confirm that they are happy with the net result</li> <li>• If small pieces of information have been lost, they should be typed into one of the merged models after synchronization</li> <li>• If a large piece of information has been lost (for example, an overwritten large Class note) use the Resolve Replication Conflicts dialog</li> </ul>	<a href="#">Resolve Conflicts</a> <sup>[315]</sup>
<b>Disable or Remove Replication Features</b>	If you have converted a project to a design master but now want to disable the replication features, you can remove Replication; ensure that you back up all your files first.	<a href="#">Remove Replication</a> <sup>[313]</sup>

### 3.6.1.4.1 Design Masters

A design master is the first converted Enterprise Architect project that supports replication. You create the master project from which you create replicas that can be modified independently of the master project and re-merged later.

**Access** [Tools](#) | [Data Management](#) | [Manage .EAP File](#) | [Make Design Master](#)

**Create a design master**

Step	Action
1	Take a back-up of the required Enterprise Architect project.
2	Select the project in the Project Browser.
3	Select the <b>Make Design Master</b> menu option and follow the on-screen instructions.

**Learn more**

- [Create Replicas](#) <sup>[312]</sup>

**3.6.1.4.2 Create Replicas**

A replica is one of several copies of the design master of an Enterprise Architect project. You create the copy of the master project for you or another user to modify independently and re-merge later.

**Access**   **Tools | Data Management | Manage .EAP File | Create New Replica**

**Create a replica**

Step	Action
1	First create a design master, then select the <b>Create New Replica</b> menu option and follow the on-screen instructions.
2	Edit the replica over time and, when required, return the file for merging with the design master.

**Notes**

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Replicas** permission to create a replica

**Learn more**

- [Design Masters](#) <sup>[311]</sup>
- [Manage Replicas User Permissions](#) <sup>[329]</sup>



### 3.6.1.4.3 Synchronize Replicas

Synchronizing replicas combines the changes made to each file. You can:

- Merge the changes made to each replica with the design master, so that a new set of replicas with all changes can be generated and distributed
- Combine the changes made to two replicas, should it be necessary for two team members to combine their work

**Access** [Tools](#) | [Data Management](#) | [Manage .EAP File](#) | [Synchronize Replicas](#)

#### Synchronize replicas

Step	Action
1	Open the design master project file (or the first required replica).
2	Select the <b>Synchronize Replicas</b> menu option.
3	Locate and select the (second) required replica to merge the open project and the replica.

#### Notes

- Information is copied both ways, including deletes, updates and inserts; both projects end up containing identical information
- If this process generates 'conflicting changes' errors, you should review and, if necessary, resolve these conflicts
- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Replicas** permission to create a replica

#### Learn more

- [Resolve Conflicts](#) <sup>[315]</sup>
- [Manage Replicas User Permissions](#) <sup>[329]</sup>

### 3.6.1.4.4 Remove Replication

Replication makes many changes to the database structure of your model, so the model file becomes considerably larger with additional information; you might, therefore, decide not to use the replication feature any more.

**Access** [Tools](#) | [Data Management](#) | [Manage .EAP File](#) | [Remove Replication](#)

#### Remove replication from your model

Step	Action
1	Ensure that you have open a temporary repository (not the one having replication removed) - the menu option is not available if no repository is open.
2	Select the <b>Remove Replication</b> menu option. The Remove Replication Wizard displays.
3	Enter the full path and file name of the project to have replication removed. Click on the <b>Next</b> button.
4	Enter the full path and file name of the base Enterprise Architect model (with no replication) to act as template. Click on the <b>Next</b> button.
5	Enter the full path and required file name for the output file. Click on the <b>Next</b> button.
6	Select whether to have a log file created and, if so, enter a file name for the log file.
7	Click on the <b>Run</b> button to begin removing replication. Enterprise Architect creates a new project containing all the model information. Your model has now had replication removed, and should be considerably smaller.

### Notes

- You cannot remove replication from a model with Auditing enabled - if you want to remove replication:
  - Disable Auditing
  - If prompted to do so, allow Enterprise Architect to roll back the database version
  - Remove replication




### Learn more

- [Auditing](#)<sup>[446]</sup>
- [Auditing Settings](#)<sup>[448]</sup>

### 3.6.1.4.5 Upgrade Replicas

A new release of Enterprise Architect might contain changes to the underlying project structure, such as more tables or changed queries. If you use Replication, you must take care with your upgrade.

#### Considerations

Consideration	Detail	See also
<b>Open Design Master Project first</b>	After installing the new version of Enterprise Architect, it is very important that you open the <i>design master</i> before opening any of the replicas with the updated version.	<a href="#">Design Masters</a> 
<b>Changes to the database design</b>	Changes to the database design in a replicated project can ONLY be done to the design master; trying to update a replica at best does nothing, and at worst causes the update of the master to fail.	
<b>Propagate Changes</b>	Design changes are propagated through to the replicas the next time the replicas are synchronized with the master.	<a href="#">Synchronize Replicas</a> 
<b>Alternatively</b>	One other strategy is to remove replication from a copy of the replica set, upgrade that project and convert it into a new design master from which new replicas are created.	<a href="#">Remove Replication</a> 

### 3.6.1.4.6 Resolve Conflicts

If two or more people each work on the same model object in their respective replicas between synchronizations, the replication engine has problems in resolving which change is the master. You need to Select which of two conflicting changes you should save to the Design Master and/or replicas, where a substantial piece of information has been overridden by a user and you want to retrieve it.

Consideration	Detail	See also
<b>Avoid the problem</b>	Ensure that each team member always works in a separate area of the model within their replica.	
<b>Check for conflicts</b>	After synchronizing replicas, open the Resolve Conflicts dialog (see below this table) and check if there were any conflicts.	
<b>Response to conflicts</b>	When a project record has been modified in different ways by different users, the replication engine selects one of the conflicting values based on rules within the JET replication manager.  However, the replication engine stores the discarded changes and flags the conflict on the Resolve Conflicts dialog so that you can choose to roll in the discarded change instead.	

Consideration	Detail	See also
	<p>Normally it is not necessary or desirable to examine conflicts, since they represent relatively inconsequential pieces of information that can very easily be modified through the normal Enterprise Architect interface; for example, by moving a diagram element.</p> <p>The only case in which the Resolve Conflicts dialog should be used is where a substantial piece of information has been overridden by a user, and you want to retrieve it.</p>	

**Access** [Tools](#) | [Data Management](#) | [Manage .EAP File](#) | [Resolve Replication Conflicts](#)

#### Resolve conflicts in changes to the same model object in two separate replicas

Step	Action
1	Synchronize a replica with the Design Master, and display the Resolve Conflicts dialog.
2	In the Table with Conflicts list, click on the entry that is likely to contain the lost information.
3	<p>Click on each entry in the Conflicting Records list.</p> <p>When the lost information appears in the Conflict Details list, click on the <b>Overwrite with Conflict</b> button.</p>
4	<p>In the Conflicting Records list, you can also copy each Row ID to the clipboard by right-clicking on the row and selecting the <b>Copy to Clipboard</b> menu option.</p> <p>You can locate the object that is in conflict in the Project Browser or diagrams, by right-clicking on it and selecting the <b>Find in Project Browser</b> or <b>Find in Diagrams</b> context menu option.</p>

### 3.6.2 User Security

**User Security** in Enterprise Architect is a means of blocking the use of model **update functions** across the model by means of **access permissions** for each function, and protecting specific elements and diagrams from change by means of **user locks**. The intent is not to prevent access to information, but to prevent inadvertent changes to data.

Security is an **optional** facility in the system. If required it is **enabled** by the Security Administrator, who at the same time sets the security **policy** to either:

- **Require User Lock to Edit** - the whole project is blocked against editing and the user locks an object to open it and edit it, or
- **User/group locking** - the whole project is open for editing and the user locks an object to protect it from being edited

The Security Administrator also sets up the user and group **IDs and passwords**, which every user requires

to log in to the model when security is enabled. Access permissions are assigned to the user IDs to determine which update functions the user can apply. The users can still **view** any information in the project. If security is **not** enabled in the project, no login is required and users do not have to have access permissions to perform update functions.

## Access   **Project | Security**

### Security Operations

Operations For	Detail	See also
<b>Administrators</b>	<p>A number of security tasks can be performed only by users with Administrative permissions to the security operations. The person who <b>enables</b> security receives online instructions to login as <b>Admin</b>. This login ID automatically:</p> <ul style="list-style-type: none"> <li>• Has access permissions to perform all security operations</li> <li>• Is a member of an Administrators user group, which also has access permissions to perform all security operations</li> </ul>	<p><a href="#">Enable/Disable Security</a> <sup>[318]</sup></p> <p><a href="#">Set Security Policy</a> <sup>[319]</sup></p> <p><a href="#">Maintain Users</a> <sup>[323]</sup></p> <p><a href="#">Import User IDs From Active Directory</a> <sup>[324]</sup></p> <p><a href="#">Assign User To Groups</a> <sup>[326]</sup></p> <p><a href="#">Set Up Single Permissions</a> <sup>[327]</sup></p> <p><a href="#">View All User Permissions</a> <sup>[328]</sup></p> <p><a href="#">Maintain Groups</a> <sup>[329]</sup></p> <p><a href="#">Set Group Permissions</a> <sup>[329]</sup></p> <p><a href="#">View and Manage Locks</a> <sup>[332]</sup></p> <p><a href="#">Password Encryption</a> <sup>[333]</sup></p> <p><a href="#">Change Password</a> <sup>[334]</sup> (of any user)</p>
<b>Users</b>	<p>Other security tasks can be performed by users who do not have Administrative rights, on work performed under their own user ID. These users must still have the appropriate access permissions to perform many of these 'user' tasks.</p>	<p><a href="#">Lock Model Elements</a> <sup>[336]</sup></p> <p><a href="#">Lock Packages</a> <sup>[337]</sup></p> <p><a href="#">Apply a User Lock</a> <sup>[339]</sup></p> <p><a href="#">Identify Who Has Locked An Object</a> <sup>[342]</sup></p> <p><a href="#">Locked Element Indicators</a> <sup>[340]</sup></p> <p><a href="#">Manage Your Own Locks</a> <sup>[342]</sup></p> <p><a href="#">Change Password</a> <sup>[334]</sup> (your own password)</p>

### Notes

- User Security can be enabled in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect

### 3.6.2.1 Enable/Disable Security

User security is not automatically enabled on the system. If your organization requires the security facilities, the Security Administrator **enables** them using an **authorization key** obtained from the Sparx Systems website. Similarly, if security facilities are no longer required, the Security Administrator explicitly **disables** security, again using the authorization key.

**Access**    **Project | Security | Enable Security**

### Enable and Disable User Security

Step	Action	See also
1	<p>Obtain the authorization key from the Sparx Systems website on:</p> <ul style="list-style-type: none"> <li>the Team Modeling Resources section (Trial User) or</li> <li>the Registered Users section (Registered User; you must also have your Registered Users login and password)</li> </ul> <p>The two authorization keys are not interchangeable - the Trial User key does not work on a registered user installation.</p>	<a href="#">Team Modeling Resources web pages</a> <a href="#">Registered Users web pages</a>
2	<p>In Enterprise Architect, select the <b>Enable Security</b> menu option.</p> <p>The Enter authorization dialog displays.</p>	
3	<p>In the <b>Enter authorization key</b> field, type the authorization key from the Sparx Systems website.</p>	
4	<p>If required, select the <b>Automatically apply Exclusive Edit Locks to diagrams</b> checkbox.</p> <p>In standard (<i>User/Group Locking</i>) mode, this option blocks multiple users from simultaneously attempting to modify the same diagram (see <i>Notes</i> below).</p> <p>This option is ignored in <i>Require User Lock to Edit</i> security mode.</p>	<a href="#">Set Security Policy</a> <sup>[319]</sup>
5	<p>Click on the <b>OK</b> button.</p> <p>Security is enabled, and an <b>Admin</b> user and <b>Administrators</b> user group are created, both with all access permissions; the Admin user has the password of <b>password</b>.</p>	<a href="#">List of Available Permissions</a> <sup>[329]</sup>
6	<p>Select the <b>Project   Security   Login as Another User</b> menu option, and</p>	

Step	Action	See also
	log in as <b>Admin</b> with the initial password of <b>password</b> . It is recommended that you change the <b>Admin</b> password immediately.	<a href="#">Change Password</a> [334]
7	Set up users and permissions as required.	<a href="#">Maintain Groups</a> [320] <a href="#">Set Group Permissions</a> [321] <a href="#">Maintain Users</a> [323] <a href="#">Set Up Single Permissions</a> [327] <a href="#">Assign User To Groups</a> [326]

### Notes

- Once security has been enabled, you must have **Security - Enable/Disable** permission to turn it off - the initial **Admin** administrator and **Administrators** group automatically have this permission; the system prompts you to log off the project and log on again, but this is not strictly necessary
- If you re-enable security, be aware that any changes you have made to the **Admin** user (password and reduced access permissions) are reset to **password** and full access; similarly, the **Administrators** user group is reinstated with full access permissions
- The **Automatically apply Exclusive Edit Locks to diagrams** option is not displayed when disabling security, therefore to toggle the setting whilst security is enabled you must disable security and re-enable it; security settings (users, groups and permissions) and locks on elements, are **not** affected by this action
- If the **Automatically apply Exclusive Edit Locks to diagrams** option is selected, as a user modifies a diagram the system automatically applies a User Lock to the diagram, preventing any other user from modifying it

It is creating difference between the database and buffer versions of the diagram that triggers the temporary lock, and elimination of difference that releases the lock; therefore, the system releases the lock when:

- The user saves the changes to the diagram, with the **Save** icon or keyboard keys
- The user undoes the last remaining action in the **Undo** list
- The user saves or discards changes via the system prompt when they close the diagram

If the diagram already has a User Lock or Group Lock that does not exclude the current user, this lock is set aside and saved when the temporary User Lock is applied; when the temporary User Lock is released, the pre-existing lock is restored

### 3.6.2.2 Set Security Policy

The security policy determines how security mechanisms are applied and interpreted on the system. There are two possible security policies in Enterprise Architect:

- User/Group Locking** mode - All elements and diagrams are considered unlocked and anyone can edit any part of the model; however, when you edit a diagram, package or element, you lock the element or set of elements at either the user level or group level

This mode is good for cooperative work groups where there is a solid understanding of who is working on which part of the model, and locking is used mainly to prevent further changes or to limit who has write access to a part of the model

- **Require User Lock to Edit** mode - More rigorous: the model is read-only, and everything is locked so that nobody can edit anything unless they explicitly check out the object with a user lock; a single 'check out' function operates on a diagram to check out the diagram and all contained elements in one go.

There are also functions on the context (right-click) menus of packages, diagrams and elements in the Project Browser to apply a user lock when this mode is in use

You would use this mode when there is a strict requirement to ensure only one person can edit a resource at one time; this is suitable for much larger projects where there might be less communication between users

**Access** **Project | Security | Require User Lock to Edit** (select for **Require User Lock to Edit** mode, **deselect for User/Group Locking** mode)

#### Notes

- Only the Admin Security Administrator, with **Admin** permissions, is able to set the security policy applied
- When you add new elements in **User/Group Locking** mode (elements editable by default), no user lock is created automatically for the newly created element
- When you add new elements in **Require User Lock to Edit** mode (elements locked by default), a user lock is created on the new element to enable instant editing

### 3.6.2.3 Maintain Groups

Whilst you can apply access permissions to each user individually, it is easier and more convenient to assign all users with the same access permission(s) to a **security group**, and assign the permissions to all the members of that group in a **single** action.

The security group also acts as a **mailbox** for Model Mail, where the group name can be selected as the addressee; when an internal mail is sent to the group, all members of the group receive that email in their Model Mail window. The group name can act as either:

- A mail **list**, in which case each group member receives their **own** copy of the message, or
- A mail **box**, in which case the email is a **single entity** and the group members do not receive separate instances of it; if one group member responds to or deletes the email, the other group members see that action as if they had performed it themselves

**Access** **Project | Security | Manage Groups**

#### Set up a security group

Field/ Button	Action	See also
<b>New</b>	Click on this button to clear the fields ready to define a new group.	
<b>Group</b>	Type the security group name.	



Field/ Button	Action	See also
<b>Name</b>		
<b>Description</b>	Type a description of the group.	
<b>Save</b>	Click on this button to save the group definition and add it to the <b>Groups</b> list.	
<b>Set Group Permissions</b>	Click on this button to immediately assign group access permissions to the group.	<a href="#">Set Group Permissions</a> [321]
<b>Shared Mail</b>	To make the group name act as a mail <b>box</b> , select this checkbox against the name in the list.  To use the group name as a mail <b>list</b> , leave the checkbox unselected.	<a href="#">Model Mail</a> [565]
<b>Close</b>	Click on this button to close the dialog.	

### Notes

- You must have **Security - Manage Users** permission to manage user groups; the initial **Admin** administrator and **Administrators** group automatically have this permission
- You do not define groups as group logins with passwords; if you intend to use a group login, you can define a single-user login and password that all group members use (that is, Enterprise Architect allows multiple logins under one user ID)
- Emails already sent to a group as a mail list and those sent to a group as a mailbox cannot be interchanged; if you change the status of the **Shared Mail** checkbox, the only way to change the distribution of past emails is to forward them to the group name again
- You can subsequently edit the group name; changes are automatically reflected in the internal Model Mail mail list or mail box

### Learn more

- [Set Group Permissions](#) [321]
- [Maintain Users](#) [323]
- [Assign User To Groups](#) [326]
- [List of Available Permissions](#) [329]

### 3.6.2.4 Set Group Permissions

Once you have set up some user groups, you can assign access permissions to each group to define what update functions in the system the group members have access to.

Access    **Project | Security | Manage Groups**

**Assign access permissions to a security group**

Step	Action	See also
1	Select the user group from the <b>Groups</b> list, so that the name displays in the <b>Group Name</b> field.	
2	Click on the <b>Set Group Permissions</b> button.  The Group Permissions dialog displays, listing all available access permissions.	<a href="#">List of Available Permissions</a> <sup>[329]</sup>
3	Scroll through the list and select the checkbox against any access permissions you want to assign to the members of this group. You can also: <ul style="list-style-type: none"> <li>Click on the <b>Select All</b> button, then <b>deselect</b> the checkbox against any access permissions you do <b>not</b> want to assign to the members of this group or</li> <li>Click on the <b>Deselect All</b> button to clear all selected permissions, then select the checkbox against any you want to keep selected</li> </ul>	
4	Click on the <b>OK</b> button to assign the permissions.  All of the users assigned to this group share in this set of permissions.	
5	Click on the <b>Close</b> button.	

**Notes**

- You must have **Security - Manage Users** permission to assign permissions to user groups; the initial **Admin** administrator and **Administrators** group automatically have this permission
- You can transport the group permissions between models as Reference Data, using the **Project | Model Import/Export | Export Reference Data** and **Import Reference Data** options

**Learn more**

- [Maintain Groups](#)<sup>[320]</sup>
- [Assign User To Groups](#)<sup>[326]</sup>
- [Export Reference Data](#)<sup>[376]</sup>
- [Import Reference Data](#)<sup>[380]</sup>

### 3.6.2.5 Maintain Users

When you have enabled security, you create the user definition for each user that has access to the model. The user definition initially consists of the user ID and the user's password. You can either set up the user manually, as below, or import the user IDs from a Windows Active Directory.

**Access**   **Project | Security | Manage Users: New**

#### Set up a user for your model

Step	Action	See also
1	<p>In the:</p> <ul style="list-style-type: none"> <li>• <b>Login</b> field, type the user ID</li> <li>• <b>Firstname</b> field, type the user's first name</li> <li>• <b>Surname</b> field, type the user's last name</li> </ul> <p>Optionally, in the <b>Department</b> field, type the name of the user's department.</p>	
2	<p>Click on the <b>Save</b> button, and then click on the <b>Change Password</b> button.</p> <p>The Change Password dialog displays.</p>	
3	<p>In the <b>New password</b> field, type the user's password.</p> <p>This can be any number of characters in length.</p> <p>(As this is a new user, the <b>Enter old password</b> field is disabled.)</p>	
4	<p>In the <b>Retype new</b> field, type the user's password again, for confirmation.</p>	
5	<p>Click on the <b>OK</b> button.</p> <p>A <i>Password Changed</i> message displays.</p> <p>Click on the <b>OK</b> button to return to the Security Users dialog.</p>	
6	<p>The user definition is complete. You can now either:</p> <ul style="list-style-type: none"> <li>• Assign the user to an existing user group</li> <li>• Assign access permissions to update functions for the new user</li> <li>• Click on the <b>New</b> button to add another user, or</li> <li>• Click on the <b>Close</b> button to exit the Security Users dialog</li> </ul>	<p><a href="#">Assign User IDs to Groups</a> <sup>[326]</sup></p> <p><a href="#">Single Permissions</a> <sup>[327]</sup></p>

#### Notes

- You must have **Security - Manage Users** permission to maintain users; the initial **Admin** administrator and **Administrators** group automatically have this permission
- You can transport the user definitions between models as Reference Data, using the **Project | Model Import/Export | Export Reference Data** and **Import Reference Data** options

#### Learn more

- [Import User IDs from a Windows Active Directory](#)<sup>[324]</sup>
- [List of Available Permissions](#)<sup>[329]</sup>
- [Export Reference Data](#)<sup>[376]</sup>
- [Import Reference Data](#)<sup>[380]</sup>

### 3.6.2.6 Import User IDs From Active Directory

Whilst you can define each of your model users individually and specifically for Enterprise Architect security, you can also import your **Windows user IDs** from Windows Active Directory and use those as the security user IDs with **Windows Authentication**. If you set up your security user IDs in this way, when a user opens the model the system checks the users database for their Windows ID and, if it matches, automatically logs the user in without prompting for a password.

As a pre-requisite, you create an appropriate user group into which to import the user IDs; you can also use this to assign appropriate permissions to the user IDs as a whole.

**Access**    **Project | Security | Manage Users**

#### Import user IDs from Windows Active Directory

Step	Action	See also
1	On the Security Users dialog do not complete any user details fields. Instead, select the <b>Accept Windows Authentication</b> checkbox and click on the <b>Import</b> button. The Import Users dialog displays.	
2	Click on the down arrow in the <b>Security Group</b> field and select the appropriate security group for the imported user IDs.	
3	Click on the <b>Add</b> button. The Select Users dialog displays.	
4	If the <b>Select this object type</b> field has not defaulted to <b>Users</b> : <ol style="list-style-type: none"> <li>1. Click on the <b>Object Types</b> button; the Object Types dialog displays</li> <li>2. Select the checkbox against <b>Users</b> (the type of object to import from the Active Directory).</li> <li>3. Click on the <b>OK</b> button to return to the Select Users dialog.</li> </ol>	
5	Click on the <b>Locations</b> button.	

Step	Action	See also
	The Locations dialog displays.	
6	Browse for and select the location to import from within the Active Directory. Click on the <b>OK</b> button to return to the Select Users dialog.	
7	In the <b>Enter the object names to select</b> field, either: <ul style="list-style-type: none"> <li>Type in the user IDs individually (click on the <b>examples</b> link to see examples of the correct formats) and go to <b>step 13</b>, or</li> <li>Click on the <b>Advanced</b> button to search for IDs; the Select Users dialog redisplay shows a Common Queries tab</li> </ul>	
8	In the <b>Name</b> and <b>Description</b> fields, type any characters or text that help identify the IDs you are searching for.	
9	In the <b>Starts with</b> field, click on the drop-down arrow and, if necessary, select a different qualifier.	
10	Optionally, to further filter the IDs to search for, select the <b>Disabled accounts</b> or <b>Non-expiring password</b> checkboxes, and/or select a value in the <b>Days since last logon</b> field.	
11	Click on the <b>Find Now</b> button to initiate the search, and to display a list of IDs in the bottom panel of the dialog.  You can vary the types of information shown here by clicking on the <b>Columns</b> button and selecting the column headings to display, then dragging the column titles into the sequence you prefer.	
12	When you have identified the IDs to import, click on a required ID (or press <b>(Ctrl)</b> or <b>(Shift)</b> while you click to select several) and click on the <b>OK</b> button.  The Select Users dialog redisplay, with the selected ID or IDs listed in the <b>Enter the object names to select</b> field.	
13	Click on the <b>OK</b> button to redisplay the Import Users dialog with the selected users' names listed in the Users panel.	
14	Click on the <b>Import</b> button to add the user IDs to the Security Users dialog.	
15	Click on a user ID to populate the dialog fields with the user ID details.  Optionally, you can start to set group membership and/or single permissions.	<a href="#">Group Membership</a> 326

Step	Action	See also
		<a href="#">Set Single Permissions</a> <sup>327</sup>

### Notes

- You must have **Security - Manage Users** permission to maintain users; the initial **Admin** administrator and **Administrators** group automatically have this permission
- As a security measure, the Windows Authentication is automatically deactivated if the project file is moved to a different location; once the file has been relocated, you can toggle the **Accept Windows Authentication** checkbox to reactivate Windows Authentication.
- Enterprise Architect generates random passwords for Windows user IDs; however, if necessary you can assign a new password to an imported user ID

### Learn more

- [Maintain Groups](#) <sup>320</sup>
- [Change Password](#) <sup>334</sup>

## 3.6.2.7 Assign User To Groups

Once you have created your user IDs and the security groups for assigning sets of access permissions to users, you can **assign each user ID** to the **security group or groups** that define the access permissions needed by that user. A user can belong to more than one security group and also have individual access permissions, appropriate to the work that they perform.

**Access**   **Project | Security | Manage Users**

### Assign a user to user groups

Step	Action	See also
1	On the Security Users dialog, select the user ID in the <b>Users</b> list so that the details display in the User Details panel.  Click on the <b>Group Membership</b> button. The User Groups dialog displays, listing the security groups that exist in the project.	
2	Select the checkbox against each security group that this user belongs to. If you select a group by accident, deselect the checkbox against it.	
3	Click on the <b>OK</b> button to assign the user to the selected groups.	
4	On the Security Users dialog either select another user to assign to security groups, or	

Step	Action	See also
	click on the <b>Close</b> button.	

#### Notes

- To assign users to groups, you must have **Security - Manage Users** permission; the initial **Admin** administrator and **Administrators** group automatically have this permission

#### Learn more

- [Maintain Groups](#) <sup>[320]</sup>
- [List of Available Permissions](#) <sup>[329]</sup>
- [Set Up Single Permissions](#) <sup>[327]</sup>

### 3.6.2.8 Set Up Single Permissions

A user commonly belongs to one or more work groups that give that user group permissions to use update facilities in various areas of the system. You can also assign additional, individual user permissions to that user ID.

**Access**   **Project | Security | Manage Users**

#### Set up single permissions for a user

Step	Action	See also
1	On the Security Users dialog, select the user ID in the <b>Users</b> list so that the details display in the User Details panel.  Click on the <b>Single Permissions</b> button.  The User Permissions dialog displays.	
2	Scroll through the list and select the checkbox against any access permissions you want to assign to this user ID. You can also: <ul style="list-style-type: none"> <li>Click on the <b>Select All</b> button, then <b>deselect</b> the checkbox against any access permissions you do <b>not</b> want to assign to this user, or</li> <li>Click on the <b>Deselect All</b> button to clear all selected permissions, then select the checkbox against any you want to keep selected</li> </ul>	<a href="#">List of Available Permissions</a> <sup>[329]</sup>
3	Click on the <b>OK</b> button to assign the selected access permissions to the user.	
4	Click on the <b>Close</b> button.	

**Notes**

- You must have **Security - Manage Users** permission to assign permissions to users; the initial **Admin** administrator and **Administrators** group automatically have this permission
- A user's total permissions are those granted by Group Membership plus those granted by specific permission assignment
- You can transport the user permissions between models as Reference Data, using the **Project | Model Import/Export | Export Reference Data** and **Import Reference Data** options

**Learn more**

- [Maintain Groups](#)<sup>[320]</sup>
- [Assign User To Groups](#)<sup>[326]</sup>
- [Export Reference Data](#)<sup>[376]</sup>
- [Import Reference Data](#)<sup>[380]</sup>

**3.6.2.9 View All User Permissions**

When a user has been set up on the system with their appropriate access permissions, you can periodically monitor their list of permissions to verify what they are able to do and check that their profile is up to date. The access permissions shown are derived from their individual profile and from their membership of security groups.

**Access**   **Project | Security | Manage Users**

**Display the permissions assigned to a user ID**

Step	Action	See also
1	On the Security Users dialog, select the user ID in the <b>Users</b> list so that the details display in the User Details panel.	
2	Click on the <b>View All</b> button.  The All user permissions dialog displays, listing all permissions available on the system.	
3	Scroll through the list. The user has all access permissions for which the checkbox is selected, and does not have the permissions for which the checkbox is clear.	

**Notes**

- You must have **Security - Manage Users** permission to maintain users; the initial **Admin** administrator and **Administrators** group automatically have this permission
- You can select and deselect checkboxes on the All user permissions dialog, but you cannot save these changes; if you want to edit the user's access permissions, click on the **Cancel** button and on the **Single Permissions** button



Learn more

- [Set Up Single Permissions](#)<sup>[327]</sup>

**3.6.2.10 List of Available Permissions**

In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions, if security is enabled, users can update information if they have the appropriate access permissions for the data update tasks for that type of data. The tasks that a user can perform with each access permission are listed below.

Some permissions take precedence over others. For example, if a user has **Use Version Control** permission, they can modify model elements on import even if they do not have **Update Element** permission.

Permissions

Permission	Enables the user to	See also
<b>Administer Database</b>	Compact and repair a project database.	<a href="#">Compact a Project</a> <sup>[601]</sup> <a href="#">Repair a Project</a> <sup>[602]</sup>
<b>Admin Workflow</b>	Develop and manage workflow scripts.	<a href="#">Workflow scripts</a> <sup>[367]</sup>
<b>Audit Settings</b>	Change the audit settings in the Audit Settings dialog.	<a href="#">Audit Settings</a> <sup>[448]</sup>
<b>Audit View</b>	Enable auditing and display data in the Audit View and Audit History tab.	<a href="#">Audit View Controls</a> <sup>[453]</sup> <a href="#">Audit History Tab</a> <sup>[455]</sup>
<b>Baselines - Manage</b>	Create, delete, import and export Baselines.	<a href="#">Package Baselines</a> <sup>[457]</sup>
<b>Baselines - Restore model</b>	Merge data into the project model from a Baseline or XML file.	<a href="#">Package Baselines</a> <sup>[457]</sup>
<b>Change Password</b>	Change your own password.	<a href="#">Change Password</a> <sup>[334]</sup>
<b>Check Data Integrity</b>	Check and repair project data integrity.	<a href="#">Check Project Data Integrity</a> <sup>[597]</sup>
<b>Configure Datatypes</b>	Add, modify and delete datatypes.	<a href="#">Data Types</a> <sup>[117]</sup>
<b>Configure Images</b>	Configure alternative element images.	<a href="#">Using The Image Manager</a> <sup>[860]</sup>
<b>Configure</b>	Configure controlled packages and package properties.	<a href="#">Configure Packages</a> <sup>[488]</sup>

Permission	Enables the user to	See also
<b>Packages</b>		
<b>Configure Resources</b>	Create and manage Resources window items: document templates, patterns, profiles and favorites.	<a href="#">Resources</a> <sup>[1173]</sup>
<b>Configure Stereotypes</b>	Add, modify and delete Stereotypes.	<a href="#">Stereotype Settings</a> <sup>[1147]</sup>
<b>Configure Version Control</b>	Set up version control options for the current model.	<a href="#">Version Control Settings</a> <sup>[415]</sup>
<b>Export XML</b>	Export a model to XML.	<a href="#">Export to XML</a> <sup>[475]</sup>
<b>Generate Documents</b>	Generate document and web reports from model packages.	<a href="#">Document Reports</a> <sup>[2640]</sup> <a href="#">Web Reports</a> <sup>[2744]</sup>
<b>Generate Source Code and DDL</b>	Generate source code and DDL from a model element, and synchronize code against model elements if it already exists.	<a href="#">Generate Source Code</a> <sup>[2111]</sup> <a href="#">Generate DDL for a Table</a> <sup>[2380]</sup> <a href="#">Update Package Contents</a> <sup>[2117]</sup>
<b>Import XML</b>	Import a model from XML.	<a href="#">Import from XML</a> <sup>[478]</sup>
<b>Lock Elements</b>	Lock an element or package.	<a href="#">Lock Model Elements</a> <sup>[336]</sup>
<b>Manage Diagrams</b>	Create new diagrams, copy and delete existing diagrams, and save a diagram as a UML Pattern.	<a href="#">Create a Pattern</a> <sup>[1464]</sup> <a href="#">Add New Diagrams</a> <sup>[822]</sup> <a href="#">Copy a Diagram</a> <sup>[842]</sup> <a href="#">Delete Diagram</a> <sup>[839]</sup>
<b>Manage Glossary</b>	Create, edit and delete glossary items in the Project Glossary.	<a href="#">Project Glossary</a> <sup>[533]</sup>
<b>Manage Issues</b>	Update and delete model Issues.	<a href="#">Project Issues</a> <sup>[528]</sup>
<b>Manage Project Calendar</b>	Add, update and delete Project Calendar events; those without this permission can view calendar items.	<a href="#">Calendar</a> <sup>[574]</sup>

Permission	Enables the user to	See also
<b>Manage Project Settings</b>	Update and manage project-wide settings including: <ul style="list-style-type: none"> <li>The available project resources, metrics and risks</li> <li>The default element font for the model and</li> <li>The model default diagram</li> </ul>	<a href="#">Project Resources</a> <sup>[512]</sup> <a href="#">Set Default Fonts</a> <sup>[624]</sup> <a href="#">Diagram Advanced Menu</a> <sup>[95]</sup> <a href="#">Set the Default Diagram</a> <sup>[844]</sup>
<b>Manage Reference Data - Update</b>	Update and delete reference items.	<a href="#">Reference Data</a> <sup>[1146]</sup>
<b>Manage Replicas</b>	Create and synchronize replicas.	<a href="#">Create Replicas</a> <sup>[312]</sup> <a href="#">Synchronize Replicas</a> <sup>[313]</sup>
<b>Manage Tests</b>	Update and delete Test records.	<a href="#">Working on Test Records</a> <sup>[2605]</sup>
<b>Reverse Engineer from DDL and Source Code</b>	Reverse engineer from source code or ODBC, and synchronize model elements against code.	<a href="#">Import Source Code</a> <sup>[2136]</sup>
<b>Security - Enable/Disable</b>	Disable user security in Enterprise Architect.	<a href="#">Enable/Disable Security</a> <sup>[318]</sup>
<b>Security - Manage Locks</b>	View and delete element locks.	<a href="#">View and Manage Locks</a> <sup>[332]</sup>
<b>Security - Manage Users</b>	Maintain users, groups and assigned permissions.	<a href="#">Maintain Users</a> <sup>[323]</sup> <a href="#">Import User IDs From Active Directory</a> <sup>[324]</sup> <a href="#">Assign User to Groups</a> <sup>[326]</sup> <a href="#">Set Up Single Permissions</a> <sup>[327]</sup> <a href="#">Maintain Groups</a> <sup>[320]</sup> <a href="#">Set Group Permissions</a> <sup>[327]</sup> <a href="#">Change Password</a> <sup>[334]</sup>

Permission	Enables the user to	See also
<b>Spell Check</b>	Spell check package and set spell check language.	<a href="#">Using the Spell Checker</a> [553]
<b>Transfer Data</b>	Transfer model between different repositories.	<a href="#">Perform a Project Data Transfer</a> [504]
<b>Transform Package</b>	Perform transformations of packages and elements.	<a href="#">Model Transformations</a> [2013]
<b>Update Diagrams</b>	Update diagram appearance, properties and layout, including the Page Setup dialog.	<a href="#">Diagram Properties</a> [823] <a href="#">Set Up Diagram Page</a> [870]
<b>Update Elements</b>	Save model changes (including delete) for elements, packages, and relationships.	<a href="#">Element Tasks</a> [900] <a href="#">Packages</a> [772] <a href="#">Connector Tasks</a> [1108]
<b>Use Version Control</b>	Check files in and out using version control.	<a href="#">Check In and Check Out Packages</a> [422]
<b>View Locks</b>	Display and clear locks that you have set.	<a href="#">Manage Your Own Locks</a> [342]

### 3.6.2.11 View and Manage Locks

When security is enabled, users lock and unlock model elements in order to work on them, which can require monitoring and control. You can periodically **view** and, if necessary, **delete** the active locks placed on elements by users.

**Access** [Project | Security | Manage Locks](#)

#### Manage user locks

Step	Action	See also
1	<p>On the Active Locks dialog, in the View Locks For panel, click on the radio button for the type of lock to view.</p> <ul style="list-style-type: none"> <li>• All</li> <li>• Groups Only</li> <li>• Users Only</li> </ul> <p>Locks of the selected type are listed in the Active Locks panel.</p>	

Step	Action	See also
2	<p>To remove a lock, click on it and click on the <b>Unlock Selected</b> button.</p> <p>You can select (and deselect) multiple locks by pressing <b>Ctrl</b> as you click on each one, or <b>Shift</b> as you click on the last block in a range.</p> <p>You can also select all locks in the list by clicking on the <b>Select All</b> button, and clear that selection by clicking on the <b>Select None</b> button (or by clicking off the list).</p>	
3	When you have finished reviewing the locks, click on the <b>Close</b> button to close the dialog.	

### Notes

- You must have **Security - Manage Locks** permission to view and delete user locks; the initial **Admin** administrator and **Administrators** group automatically have this permission
- The Active Locks dialog does not currently show any **Full** locks set in the model
- If you want to display the resulting information in a more readable layout, you can resize the dialog and its columns

### Learn more

- [List of Available Permissions](#)<sup>[329]</sup>

### 3.6.2.12 Password Encryption

In releases of Enterprise Architect prior to **7.1**, Security Administrators of projects on SQL Server or Oracle repositories can **encrypt** the password used to set up the connection between Enterprise Architect and the repository. The system user does not have the real password, thereby preventing them from accessing the repository using other tools such as Query Analyzer or SQLPlus.

For password encryption for all repositories at and beyond release 7.1, you provide the connection but protect the password using **model shortcuts**.

**Access**    **Project | Security | Encrypt Password**

### Encrypt a repository password

Step	Action	See also
1	<p>In the <b>Password</b> field, enter the password to access the repository.</p> <p>The <b>Encrypted</b> field displays the modified password to be provided to users.</p>	
2	Instruct users to connect Enterprise Architect to the repository using the encrypted password prefixed with <b>\$\$</b> .	<a href="#">Connect to Oracle Data Repository</a> <sup>[255]</sup>

Step	Action	See also
		<a href="#">Connect to SQL Server Data Repository</a> <sup>[218]</sup>

### Notes

- Do not click on the **Test Connection** button as it can cause an error with encrypted passwords
- For SQL Server repositories, you must enter the **Initial Catalog** details from the All tab of the Data Link Properties dialog

### Learn more

- [Model Shortcuts](#)<sup>[204]</sup>

### 3.6.2.13 Change Password

It is recommended that users of any computer system change their passwords - or have them changed - at regular intervals. When security is enabled in Enterprise Architect, users can change their own passwords or, if the user is unable or not authorized to do this, a Security Administrator can set or change the user's password.

**Access**    **Project | Security | Change Password** (User)  
**Project | Security | Manage Users** (Administrator)

#### User - Change your own user password

Step	Action	See also
1	On the Change Password dialog, in the <b>Enter old password</b> field, type your <b>current</b> password.	
2	In the <b>New password</b> field, type your new password. This can be any number of characters in length.	
3	In the <b>Retype new</b> field, type your new password again, for confirmation.	
4	Click on the <b>OK</b> button. The <i>Password Changed</i> message displays.	
5	Click on the <b>OK</b> button to clear the message. Use your new password next time you log in.	

Step	Action	See also

#### **Administrator - Set or change any user's password**

Step	Action	See also
<b>1</b>	On the Security Users dialog, in the <b>Users:</b> panel, click on the user's name. The user's details display in the dialog fields.	
<b>2</b>	Click on the <b>Change Password</b> button. The Change Password dialog displays.	
<b>3</b>	In the <b>New password</b> field, type the user's new password. This can be any number of characters in length. You do not have to enter the user's current password.	
<b>4</b>	In the <b>Retype new</b> field, type the user's password again, for confirmation.	
<b>5</b>	Click on the <b>OK</b> button. The <i>Password Changed</i> message displays.	
<b>6</b>	Click on the <b>OK</b> button.	
<b>7</b>	By secure means, notify the user of their new password.	

#### **Notes**

- A user must have **Change Password** permission to change their own password; the initial **Admin** administrator and **Administrators** group automatically have this permission
- A Security Administrator must have **Security - Manage Users** permission to change other users' passwords; the initial **Admin** administrator and **Administrators** group automatically have this permission

#### **Learn more**

- [List of Available Permissions](#) 

### 3.6.2.14 Lock Model Elements

If you need to set a lock on a **package**, **element** or **diagram**, or clear that lock, you can do so from either the Project Browser or - for an element or diagram - from within a diagram. You follow one of three procedures, depending on whether you are:

- Locking an element or diagram under the User/Group Locking security policy
- Locking a package under the User/Group Locking security policy
- Locking an element or diagram under the Require User Lock to Edit security policy

#### Notes

- You must have **Lock Elements** permission to lock an element or diagram

#### Learn more

- [Set Security Policy](#)<sup>[319]</sup>
- [Lock Objects Under User/Group Locking](#)<sup>[336]</sup>
- [Lock Packages Under User/Group Locking](#)<sup>[337]</sup>
- [Lock Objects Under Require User Lock to Edit](#)<sup>[339]</sup>
- [Locked Element Indicators](#)<sup>[340]</sup>
- [List of Available Permissions](#)<sup>[329]</sup>

#### 3.6.2.14.1 Lock Objects Under User/Group Locking

Under the **User/Group Locking** security policy, if you need to set or release a lock on an **element** or **diagram**, you can do so from either the Project Browser or from within the diagram.

**Access** Select one of:

- **Project Browser | right-click on diagram | Lock**
- **Project Browser | right-click on element | Lock**
- **Diagram | right-click on diagram background | Lock** or
- **Diagram | right-click on element | Lock**

#### Set a lock on an element or diagram

Step	Action	See also
1	<p>On the Lock Element or Lock Diagram dialog, in the Lock Type panel, select the radio button for the required option:</p> <ul style="list-style-type: none"> <li>• <b>No lock</b> - do not set a lock on this object; clear any existing lock that <b>you</b> have set, or clear any <b>Full lock</b> that <b>other users</b> have set</li> <li>• <b>Full lock</b> - lock this object so that no-one can edit it without specifically clearing the lock</li> <li>• <b>User lock</b> - lock this object so that only you can make further edits; other users cannot unlock or edit the object</li> <li>• <b>Group lock</b> - lock this object so that any member of the specified group (below) can edit the object; other users cannot unlock or edit the object</li> </ul>	



Step	Action	See also
2	<p>If you have selected the <b>Group lock</b> option, in the <b>GroupID</b> field click on the drop-down arrow and select the group containing users that can edit the object.</p> <p>The <b>GroupID</b> drop-down list only includes groups which you are a member of.</p>	<a href="#">Assign User To Groups</a> <a href="#">[326]</a>
3	Click on the <b>OK</b> button.	

### Notes

- To check whether the project security is in **User/Group locking** mode, select **Project | Security**; the **Require User Lock to Edit** option should be **deselected**
- You must have **Lock Elements** permission to lock an element or diagram
- If the item already has a lock, only the corresponding lock option and the **No lock** option are highlighted; you have to release the lock in order to set a different type of lock
- If a **diagram** is locked and you select an element on it, the element border displays in **red**, indicating that you cannot move it or resize it
- If an **element** is locked and you click on it on a diagram, the element border displays in **black**; you can display the properties but not change them
- If you select the **Full Lock , no-one may edit** option, a red exclamation mark displays against the object in the Project Browser
- If you select the **User Lock, locking user may still edit** or **Group lock, locking group may still edit** options, a blue exclamation mark displays against the object in the Project Browser; **other** users see a red exclamation mark

### Learn more

- [Set Security Policy](#)<sup>[319]</sup>
- [Lock Packages Under User/Group Locking](#)<sup>[337]</sup>
- [Lock Objects Under Require User Lock to Edit](#)<sup>[339]</sup>
- [List of Available Permissions](#)<sup>[329]</sup>
- [Locked Element Indicators](#)<sup>[340]</sup>

#### 3.6.2.14.2 Lock Packages Under User/Group Locking

If, in **User/Group locking** mode, you want to lock or unlock the contents of a **package**, you can do so in a single operation. You can set the lock on the entire contents (including child packages), just on the top-level package content, or just on the elements or diagrams in the package.

**Access** **Project Browser | Package context menu | Lock Package**

### Lock or Unlock a package

Step	Action	See also
1	<p>In the Lock Type panel, select the appropriate radio button for the lock to apply:</p> <ul style="list-style-type: none"> <li>• <b>No lock</b> - do not set a lock on this package; clear any existing lock that <b>you</b> have set, or clear any <b>Full lock</b> that <b>other users</b> have set</li> <li>• <b>Full lock</b> - lock this package so that no-one can edit it without specifically clearing the lock</li> <li>• <b>User lock</b> - lock this package so that only you can make further edits; other users cannot unlock or edit the package</li> <li>• <b>Group lock</b> - lock this package so that any member of the specified group (below) can edit the object; other users cannot unlock or edit the package</li> </ul>	
2	<p>If you have selected the <b>Group lock</b> option, in the <b>GroupID</b> field click on the drop-down arrow and select the group containing users that can edit the object.</p> <p>The <b>GroupID</b> drop-down list only includes groups which you are a member of.</p>	<a href="#">Assign User To Groups</a> <sup>[326]</sup>
3	<p>The <b>What to Process</b> checkboxes default to selected to also lock or unlock:</p> <ul style="list-style-type: none"> <li>• Elements and/or diagrams in the package</li> <li>• The contents of child packages (that is, the whole branch)</li> </ul> <p>If you want to exclude any type of package content from the change in lock status, deselect each appropriate checkbox.</p>	
4	Click on the <b>OK</b> button to apply the lock.	

### Notes

- To check whether the project security is in **User/Group locking** mode, select **Project | Security**; the **Require User Lock to Edit** option should be **deselected**
- You must have **Lock Elements** permission to lock a package
- If the package is already locked, only the corresponding lock option and the **No lock** option are highlighted; you have to release the lock in order to set a different type of lock
- If you select the **Full Lock , no-one may edit** option, a red exclamation mark displays against the package in the Project Browser
- If you select the **User Lock, locking user may still edit** or **Group lock, locking group may still edit** options, a blue exclamation mark displays against the package in the Project Browser; **other** users see a red exclamation mark

### Learn more

- [Set Security Policy](#) <sup>[319]</sup>
- [Lock Objects Under User/Group Locking](#) <sup>[336]</sup>
- [Lock Objects Under Require User Lock to Edit](#) <sup>[339]</sup>
- [List of Available Permissions](#) <sup>[329]</sup>

- [Locked Element Indicators](#)<sup>[340]</sup>

### 3.6.2.14.3 Lock Objects Under Require User Lock to Edit

In the **Require User Lock to Edit** security mode, if you want to edit one or more of the diagrams or elements in a package, you need to set a User Lock on either the specific objects or the package that contains them. You can set or release the lock from either a diagram or the Project Browser. Once you have set a lock, only you or the Security Administrator can release it again; no other user can release your locks.

**Access** Select one of:

- **Project Browser | right-click on package | Apply/Release User Lock**
- **Project Browser | right-click on diagram | Apply/Release User Lock**
- **Project Browser | right-click on element | Apply/Release User Lock**
- **Diagram | right-click on background | Apply/Release User Lock** or
- **Diagram | right-click on element | Apply/Release User Lock**

#### Set or clear a user lock in Require User Lock to Edit security mode

Step	Action	See also
1	On the Set User Lock dialog, click on either the: <ul style="list-style-type: none"> <li>• <b>Apply User Lock</b> radio button to set the lock or</li> <li>• <b>Release User Lock</b> radio button to clear the lock</li> </ul>	
2	For a package, if you want to also lock all child packages, select the <b>Include Child Packages</b> checkbox.  If any elements in the package tree are locked by other users, a list of elements that couldn't be locked displays.	
3	Click on the <b>OK</b> button.  The system locks or unlocks: <ul style="list-style-type: none"> <li>• The selected element</li> <li>• The selected diagram and any elements in the diagram, or</li> <li>• The package and all elements and diagrams in the top level of the package</li> </ul>	

#### Notes

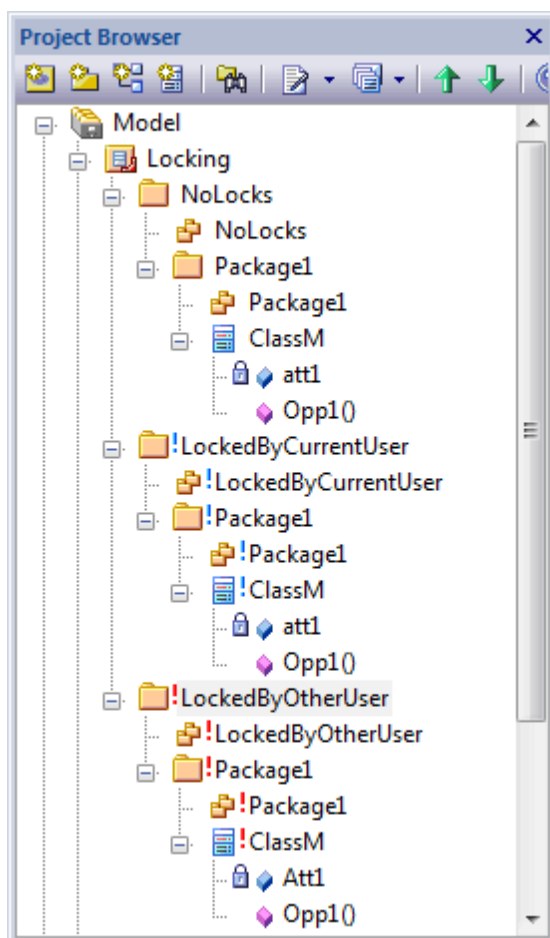
- To check whether the project security is in **Require User Lock to Edit** mode, select **Project | Security**; the **Require User Lock to Edit** option should be **selected**
- You must have **Lock Elements** permission to lock a package
- When you set a user lock, a blue exclamation mark displays against the object or objects that are locked; other users see these as red exclamation marks

Learn more

- [Set Security Policy](#)<sup>[319]</sup>
- [Lock Objects Under User/Group Locking](#)<sup>[336]</sup>
- [Lock Packages Under User/Group Locking](#)<sup>[337]</sup>
- [Locked Element Indicators](#)<sup>[340]</sup>

**3.6.2.15 Locked Element Indicators**

When a user sets a lock on an item through User Security, the lock status of the item is indicated in the Project Browser by a marker against the item - a red or blue exclamation mark - as shown:



The effect of the lock and the meaning of the status marker depend on the security policy applied to your project.

Marker meanings, by security mode

Security Mode	Marker	Meaning	See also
<b>Require User Lock to Edit</b>	No marker	There is no lock, the item <b>is not</b> editable, but any user can now apply	<a href="#">Lock Objects Under Require</a>

Security Mode	Marker	Meaning	See also
		a user lock to edit the item.	<a href="#">User Lock to Edit</a> [339]
	Blue exclamation mark	You have applied a user lock and can edit the item; no other user can release the lock, set their own lock or edit the item.	
	Red exclamation mark	Another user has applied a user lock, and you cannot release the lock, set your own lock or edit the item.  You can find out which user has locked the item.	<a href="#">Identify Who Has Locked An Object</a> [342]
<b>User/Group Lock</b>	No marker	There is no lock, the item is editable, but any user can now apply a user or group lock.	<a href="#">Lock Objects Under User/Group Locking</a> [336]  <a href="#">Lock Packages Under User/Group Locking</a> [337]
	Blue exclamation mark	The item has a lock set by you or a group including your user ID as a member, and you can edit the item.	
	Red exclamation mark	The item has a lock set by another user, or a group of which you are <b>not</b> a member; you cannot edit the item.  You can find out which user has locked the item.  The red exclamation mark also indicates that you or another user has set a <b>full lock</b> on the item. Any user can clear that lock.	<a href="#">Identify Who Has Locked An Object</a> [342]

**Notes**

- If a diagram is locked and you select an object on it, the object border displays in red; this indicates that you cannot change the location or size of the object on the diagram

**Learn more**

- [Set Security Policy](#)  
[319]

### 3.6.2.16 Identify Who Has Locked An Object

When you are working in your model, you might find that you are unable to update an element, diagram or package. If you cannot update any object, this might mean that you do not have the access permissions to update diagrams or elements. However, if the object in the Project Browser has a red exclamation mark next to it this indicates that either another person has placed a user or group lock on the object, or you or another user have put a full lock on the object.

You can quickly establish if the lock is a full lock (which you can remove) or which user has set a user or group lock.

#### Access

- **Project Browser | Package context menu | Lock**
- **Project Browser | Diagram context menu | Lock**
- **Project Browser | Element context menu | Lock**

#### Identify the lock holder

Step	Action	See also
1	<p>In the Project Browser, right-click on the diagram, package or element that has a red lock indicator, and select the <b>Lock</b> option.</p> <p>If the lock is:</p> <ul style="list-style-type: none"> <li>• A <b>Full</b> lock, the Lock &lt;object&gt; dialog displays, and you can delete the lock and set your own</li> <li>• A <b>user</b> lock or <b>group</b> lock, a message displays showing the user ID of the person or group who currently holds the lock on that item; click on the <b>OK</b> button to clear the lock</li> </ul>	<p><a href="#">Lock Objects Under User/Group Locking</a><sup>[336]</sup></p> <p><a href="#">Lock Packages Under User/Group Locking</a><sup>[337]</sup></p>

### 3.6.2.17 Manage Your Own Locks

As you are working in your model, you might set user locks on elements, diagrams and packages so that you can work on them and protect that work while it is progress. Having completed your work, you might then want to remove those locks. You can do this on each object individually, using the same procedure as you used to set the lock. You can also display a list of all the locks you have set, and remove selected locks or all locks at once. This is especially useful when working under the **Require user locks to edit** security policy.

**Access** **Project | Security | Manage My Locks (Ctrl+Shift+L)**

#### Manage your own locks

Step	Action	See also
1	<p>All the locks that <b>you</b> have set in the model are listed on the My Locks dialog.</p> <p>This dialog does not show Full locks or Group locks.</p>	<p><a href="#">Lock Objects Under User/Group Locking</a><sup>[336]</sup></p>

Step	Action	See also
		<a href="#">Lock Packages Under User/Group Locking</a> <a href="#">Lock Objects Under Require User Lock to Edit</a>
2	If you intend to clear locks, either: <ul style="list-style-type: none"> <li>• Select a single lock you intend to clear</li> <li>• Press <b>(Ctrl)</b> or <b>(Shift)</b> as you select a number of locks or a range of locks respectively, to clear</li> <li>• Click on <b>Select All</b> to select all of your locks, or</li> <li>• Click on <b>Select None</b> to clear your selection, if you have made an error</li> </ul>	
3	If you have selected locks to clear, click on the <b>Unlock Selected</b> button.  The objects that were locked are now unlocked.	

#### Notes

- You need to have **Lock Elements** access permission to **set** user locks on modeling elements, but you do not require any access permission to **list** or **clear** them using this procedure

#### Learn more

- [Set Security Policy](#)

### 3.6.3 Team Review Tools

If the project team members require a forum in which to discuss the development and progress of the project, they can make use of the facilities of the Enterprise Architect **Team Review**.













**Access** **View | Team Review (Ctrl+Alt+U)**

#### Features

Feature	Description	See also
<b>Structure</b>	<p>A point of discussion and the responses (<b>Comments</b>) to that point are created as <b>Documents</b>. Documents are held in <b>Topic</b> groups, which are in turn organized into <b>Categories</b>; your organization decides on the grouping and organization of Documents.</p> <p>You can create text to explain Categories and Topics, and you can create and respond to Documents and Comments on</p>	<a href="#">Add a New Category</a> <a href="#">Add a New Topic</a> <a href="#">Add a New Document</a>

Feature	Description	See also
	Documents.	
<b>Displays</b>	<p>The Team Review has two main displays:</p> <ul style="list-style-type: none"> <li>The <b>Team Review window</b> is used to create new Categories and Topics and to delete messages; you operate on it by selecting options from the context menu</li> </ul> <p>Each item in the hierarchy displays a mouse-over <b>tooltip</b>, showing the item title, the author's name and the date and time the item was created</p> <ul style="list-style-type: none"> <li>The <b>Team Review tab</b>, in the main work area, is used to create, view, edit, print, and comment on Documents, and to create and maintain the descriptions of Topics and Categories</li> </ul> <p>This tab has a status bar that shows the item author's name, the date and time the item was created, and the date and time the item was last modified</p>	<p><a href="#">Work on Team Review Items</a> <sup>[346]</sup></p> <p><a href="#">Comment on a Document</a> <sup>[355]</sup></p>
<b>Facilities</b>	<p>Within the Team Review you can:</p> <ul style="list-style-type: none"> <li>Select to display and edit an item in the Team Review tab, by clicking on the item name in the Team Review window</li> <li><b>Search</b> for text strings in the item titles to locate Documents on a specific subject, using the <b>Search</b> icon in the toolbar of the Team Review window</li> <li>Add <b>resources</b> to a Category, Topic or Document, such as diagram images and XML files of Packages; these are held in a <i>Resources</i> folder under the selected Team Review item</li> <li>Link model elements, diagrams, external files and other Documents to a Document</li> <li>Link to a Team Review document from the linked document on an element</li> <li>Change the loading behavior of the Team Review</li> <li>Access Team Reviews from other projects, including those on other servers</li> </ul>	<p><a href="#">Edit an Item</a> <sup>[358]</sup></p> <p><a href="#">Search Team Review</a> <sup>[362]</sup></p> <p><a href="#">Team Review Resources</a> <sup>[361]</sup></p> <p><a href="#">Add Object Links</a> <sup>[359]</sup></p> <p><a href="#">Hyperlink From Linked Document</a> <sup>[1095]</sup></p> <p><a href="#">Team Review Options</a> <sup>[362]</sup></p> <p><a href="#">Team Review Connections</a> <sup>[363]</sup></p>
<b>Specification Manager</b>	<p>The Team Review acts as a repository for the review documents generated in the <b>Specification Manager</b>. These documents are automatically stored in Topics within the <i>Formal Reviews</i> Category, but you can redirect documents to a different Topic either from the:</p> <ul style="list-style-type: none"> <li>Specification Manager, using the <b>Bind Package to Team Review Folder</b> option, or</li> <li>Team Review, using the <b>Bind to Project Browser Package</b> option</li> </ul>	<p><a href="#">Specification Manager</a> <sup>[1728]</sup></p> <p><a href="#">Create Review Document</a> <sup>[1749]</sup></p> <p><a href="#">Work on Team Review Items</a> <sup>[346]</sup></p>



Feature	Description	See also
<b>Icons</b>	<p>Each item in the Team Review window has an icon that indicates the nature or status of the item. The meaning of each of these icons is explained here:</p> <p> Document (name in <b>bold</b> indicates Document is unread)</p> <p> Comment (name in <b>bold</b> indicates Comment is unread)</p> <p> Protected by password (also on Category and Topic icons indicates that a user with the password has opened the item; icon indicates that the item is locked)</p> <p> Status <b>Awaiting Approval</b> (also on Category and Topic icons)</p> <p> Status <b>Approved</b> (also on Category and Topic icons)</p> <p> Status <b>Rejected</b> (also on Category and Topic icons)</p> <p> Category (name in <b>bold</b> indicates Category is unread)</p> <p> Topic (name in <b>bold</b> indicates Topic is unread)</p> <p> <i>Resources</i> folder for a Category, Topic or Document</p> <p> Linked objects folder for a Category, Topic or Document</p> <p> Diagram or clipboard image within <i>Resources</i> folder</p> <p> XML file of Package, within <i>Resources</i> folder</p>	<p><a href="#">Work on Team Review Items</a> <sup>[348]</sup></p> <p><a href="#">Add Object Links</a> <sup>[359]</sup></p>

**Notes**

- You can transport your Team Review set-up between projects, using the **Export Reference Data** and **Import Reference Data** options

**Learn more**

- [Export Reference Data](#)<sup>[376]</sup>
- [Import Reference Data](#)<sup>[380]</sup>

### 3.6.3.1 Work on Team Review Items







Within the Team Review, you can perform a wide range of operations on the Documents and the Topics and Categories that contain them. These operations include creating an item, Commenting on a Document, password-protecting an item so that it is not accidentally changed, accessing protected items, adding resources to an item, and deleting an item. All of these operations are accessed through the context menu in the Team Review window.

**Access**    **View | Team Review > right-click on item**

#### Team Review Options

Option	Action	Shortcut	See also
<b>New Category</b> <b>New Topic</b> <b>New Document</b>	Add a new Category, new Topic or new Document to the Team Review.  Alternatively, for a Topic or Document, click on the <b>New</b> icon in the window toolbar.		<a href="#">Add a New Category</a> <sup>[350]</sup> <a href="#">Add a New Topic</a> <sup>[351]</sup> <a href="#">Add a New Document</a> <sup>[353]</sup>
<b>New Category From Template</b> <b>New Topic From Template</b> <b>New Document from Template</b>	Add a new Category, new Topic or new Document, based on a defined template.		<a href="#">Add a New Category</a> <sup>[350]</sup> <a href="#">Add a New Topic</a> <sup>[351]</sup> <a href="#">Add a New Document</a> <sup>[353]</sup>
<b>New Comment</b>	Create a response to the selected Document.  Alternatively, click on the <b>New Comment</b> icon in the window toolbar.		<a href="#">Comment on a Document</a> <sup>[355]</sup>
<b>Rename</b>	Edit the name, in situ, of the currently-selected item.	<b>F2</b>	
<b>Copy Path to Clipboard</b>	Copy the path of the currently-selected item to the clipboard.  You can then paste the path into a document or file to add the discussion to any text concerning the item.	<b>Ctrl+C</b>	
<b>Show Contents</b>	Display the description or text of the item selected from the Team Review window, if the Team Review tab is not yet open.		
<b>Share Resource</b>	(If anything other than a Resources folder is		

Option	Action	Shortcut	See also
	<p>selected.)</p> <p>Add an item into the <i>Resources</i> folder under the selected Category, Topic or Document. If a <i>Resources</i> folder does not exist, one will be automatically created.</p> <p>Available options:</p> <ul style="list-style-type: none"> <li>• Package from Current Model</li> <li>• Image of Active Diagram</li> <li>• Active Profiler Report</li> <li>• Bookmark Active Website</li> <li>• Image from Clipboard</li> </ul>		<a href="#">Team Review Resources</a> <sup>[36]</sup>
<b>Add Package From Current Model</b>	<p>(If a <i>Resources</i> folder is selected.)</p> <p>Export a Package as an XML file from the current model as a resource of the selected Category, Topic or Document.</p> <p>You browse for and select the required Package using the Select &lt;item&gt; dialog.</p>		<a href="#">Team Review Resources</a> <sup>[36]</sup> <a href="#">Select &lt;item&gt; Dialog</a> <sup>[99]</sup>
<b>Import to Current Model</b>	<p>(If a Package XML file resource is selected.)</p> <p>Import the resource Package to the current model.</p> <p>You browse for and select the required target Package using the Select &lt;item&gt; dialog; the resource is imported as a child of that Package.</p> <p>This is a useful option for transferring relevant Packages from the Team Review of one model into another model.</p>		<a href="#">Select &lt;item&gt; Dialog</a> <sup>[99]</sup>
<b>Image of Active Diagram</b>	<p>(If a <i>Resources</i> folder is selected.)</p> <p>Add an image of the currently-displayed diagram as a resource of the selected Category, Topic or Document.</p> <p>You are prompted to provide a reference name for this image.</p>		<a href="#">Team Review Resources</a> <sup>[36]</sup>
<b>Active Profiler Report</b>	<p>(If a <i>Resources</i> folder is selected.)</p> <p>Add an active Profiler Report as a resource of the selected Category, Topic or Document.</p> <p>You are prompted to browse for and select the appropriate active report.</p>		<a href="#">Team Review Resources</a> <sup>[36]</sup> <a href="#">Generate, Save and Load Reports</a> <sup>[256]</sup>
<b>Image From</b>	(If a <i>Resources</i> folder is selected.)		<a href="#">Team Review</a>

Option	Action	Shortcut	See also
<b>Clipboard</b>	Add an image held on the clipboard as a resource of the selected Category, Topic or Document.		<a href="#">Resources</a> 
<b>View Image</b>	(If a diagram image resource is selected.) Open the View Image window, containing an image of the selected diagram. Alternatively, double-click on the image name.		<a href="#">Team Review Resources</a> 
<b>Copy Image To Clipboard</b>	(If an image resource is selected.) Copy the image or diagram image to the clipboard.		<a href="#">Team Review Resources</a> 
<b>Refresh Category 'xyz'</b> <b>Refresh Topic 'xyz'</b> <b>Refresh Document 'xyz'</b> <b>Refresh Comment 'xyz'</b>	Refresh the currently-open Category, Topic, Document or Comment, getting new Comments, Documents and Topics that other users might have created.  However, if you open another Category, Topic or Document the Team Review always displays the latest information from the database.  Alternatively, click on the <b>Refresh</b> icon in the window toolbar.		
<b>Reload Current Connection</b>	Reload the entire <b>Team Review</b> connection, getting new Categories, Topics and Documents.		
<b>Review Status</b>	Assign or clear a status marker against the selected Category, Topic or Document; you can mark the item as: <ul style="list-style-type: none"> <li>• <b>Awaiting Approval</b></li> <li>• <b>Approved</b></li> <li>• <b>Rejected</b></li> </ul> Or clear the marker ( <b>None</b> ).		<a href="#">Team Review Tools (Icons)</a> 
<b>Mark</b>	Mark the selected item (and, if appropriate, its contents) as read or unread.  The names of unread (such as newly-posted) items are in bold. When you open the item, the name reverts to normal type.		<a href="#">The Mark Submenu</a> 
<b>Bind to Project Browser Package</b>	Bind the selected Topic to a Package in the Project Browser, so that <b>Review Documents</b> created in the <b>Specification Manager</b> for that Package are automatically stored in the Topic.		<a href="#">Specification Manager</a>  <a href="#">Create Review</a>

Option	Action	Shortcut	See also
	When you select this option, the Select Package to Bind dialog displays, on which you browse for and select the Package to bind the Topic to.		<a href="#">Document</a> <sup>[1749]</sup>  <a href="#">Select &lt;Item&gt; Dialog</a> <sup>[994]</sup>
<b>Security Options</b>	<p>Access one of three options:</p> <ul style="list-style-type: none"> <li>• <b>Apply Password Lock</b> - to display a prompt to enter a security password</li> <li>• <b>Unlock for Editing</b> - to display a prompt to enter the set password so that you can edit the item</li> <li>• <b>Remove Password Lock</b> - to display a prompt for the set password, which is then removed</li> </ul> <p>When you set the password, an exclamation mark icon is added to the Category, Topic or Document name.</p>		<a href="#">Protection Against Editing</a> <sup>[357]</sup>
<b>Connections...</b>	<p>Access other Team Reviews from other Enterprise Architect models or models located on servers.</p> <p>Alternatively, click on the drop-down arrow in the <b>Connection Options</b> field in the window toolbar, and select one of the listed models.</p> <p>Use the <b>&lt;Configure Connections&gt;</b> option to add and connect to additional Team Reviews.</p>		<a href="#">Team Review Connections</a> <sup>[363]</sup>
<b>Options...</b>	Change the loading behavior of the Team Review.		<a href="#">Team Review Options</a> <sup>[362]</sup>
<b>Delete Category 'xyz'</b> <b>Delete Topic 'xyz'</b> <b>Delete Document 'xyz'</b> <b>Delete Resource 'xyz'</b> <b>Delete Comment 'xyz'</b>	<p>Delete this Category, Topic, Document or Comment and all sub-topics and sub-documents, or delete the resource attached to the item.</p> <p>Alternatively, click on the item and press ( <b>Delete</b> ).</p> <p>A confirmation dialog displays; click on the <b>Yes</b> button to remove the item and any dependent items from the Team Review.</p>		

### The Mark Submenu

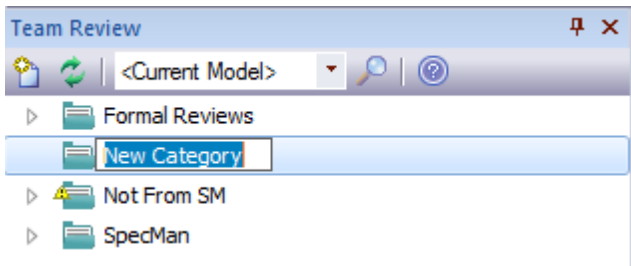
Option	Action	Shortcut	See also
<b>All items as Read</b>	Mark the entire contents of the Team Review as read.		<a href="#">Team Review Tools</a> <sup>[343]</sup>
<b>All items as Unread</b>	Mark the entire contents of the Team Review as unread.		<a href="#">Team Review Tools</a> <sup>[343]</sup>
<b>Branch as Read</b>	Mark this item and all its contents as read.		
<b>Branch as Unread</b>	Mark this item and all its contents as unread.		
<b>'xyz' as Unread</b>	Mark only this item as unread.		

### 3.6.3.2 Add a New Category

You can add new Categories to the Team Review, in which to store related Topics and Documents.

**Access** **View | Team Review** ( Ctrl+Alt+U )

#### Create a Category

Step	Action	See also
1	<p>Right-click on a blank area in the Team Review window and select the <b>New Category</b> context menu option.</p> <p>A new Category icon displays in the hierarchy.</p> 	<a href="#">Work on Team Review Items</a> <sup>[346]</sup>
2	<p>Overtyping the <i>New Category</i> text with the name of the Category, and clicking off the name.</p> <p>The Category description editor displays in the Team Review tab; type the appropriate Category description, if required.</p>	
3	Add new Topics and/or resources to the Category.	<a href="#">Add a New Topic</a> <sup>[351]</sup>

Step	Action	See also
		<a href="#">Team Review Resources</a> <sup>[36]</sup>
<b>Alternatively</b>		
<b>1</b>	Right-click on a blank area in the Team Review window and select the <b>New Category from template</b> context menu option.  A new Category icon displays in the hierarchy.	<a href="#">Work on Team Review Items</a> <sup>[34]</sup>
<b>2</b>	Overtyping the <i>New Category</i> text with the name of the Category, and clicking off the name.  The New Category from Template dialog displays.	
<b>3</b>	Click on the <b>Template</b> drop-down arrow and select a predefined template for the Category description.  Click on the <b>OK</b> button.	
<b>4</b>	The Category description editor displays in the Team Review tab; type the appropriate Category description, if required.	
<b>5</b>	Add new Topics and/or resources to the Category.	<a href="#">Add a New Topic</a> <sup>[35]</sup> <a href="#">Team Review Resources</a> <sup>[36]</sup>

### 3.6.3.3 Add a New Topic

You can add new Topics to a Category in the Team Review, in which to store related Documents and their Comments.

**Access** **View | Team Review** (Ctrl+Alt+U)

#### Create a new Topic

Step	Action	See also
<b>1</b>	Either: <ul style="list-style-type: none"> <li>Right-click on the required Category name in the Team Review window and select the <b>New Topic</b> context menu option</li> <li>Click on the Category name and click on the <b>New</b> icon in the Team</li> </ul>	<a href="#">Work on Team Review Items</a> <sup>[34]</sup>

Step	Action	See also
	<p>Review toolbar, or</p> <ul style="list-style-type: none"> <li>Click on the Category name and press ( <b>Ctrl+N</b> )</li> </ul> <p>A new <b>Topic</b> icon displays under the Category name.</p>	
2	<p>Overtyping the <i>New Topic</i> text with the name of the Topic, and clicking off the name.</p> <p>The Topic description editor displays in the Team Review tab; type the appropriate Topic description, if required.</p>	
3	Add new Documents and/or resources to the Topic.	<a href="#">Add a New Document</a> <small>[353]</small> <a href="#">Team Review Resources</a> <small>[361]</small>
<b>Alternatively</b>		
1	<p>Right-click on the required Category name in the Team Review window and select the <b>New Topic from Template</b> context menu option.</p> <p>A new Topic icon displays under the Category Name.</p>	<a href="#">Work on Team Review Items</a> <small>[346]</small>
2	<p>Overtyping the <i>New Topic</i> text with the name of the Topic, and clicking off the name.</p> <p>The New Topic from Template dialog displays.</p>	
3	<p>Click on the <b>Template</b> drop-down arrow and select a predefined template for the Topic description.</p> <p>Click on the <b>OK</b> button.</p>	
4	The Topic description editor displays in the Team Review tab; type the appropriate Topic description, if required.	
5	Add new Documents and/or resources to the Topic.	<a href="#">Add a New Document</a> <small>[353]</small> <a href="#">Team Review Resources</a> <small>[361]</small>

**Notes**



- If you already have the Team Review tab open, the New Topic from Template dialog displays as soon as you select the **New Topic from Template** menu option; you can click on and overtype the Topic Name after you have created the topic description.

### 3.6.3.4 Add a New Document

Within the Team Review, you can create a new Document on a Topic as a:

- Blank document
- Document based on a predefined template
- Document from a file link

Access **View | Team Review** ( **Ctrl+Alt+U** )

#### Create a blank Document

Step	Action	See also
1	<p>Either:</p> <ul style="list-style-type: none"> <li>• Right-click on the appropriate Topic name in the Team Review window and select the <b>New Document</b> context menu option</li> <li>• Click on the Topic name and on the <b>New Document</b> icon in the Team Review toolbar, or</li> <li>• Click on the Topic name and press ( <b>Ctrl+N</b> )</li> </ul> <p>A new Document icon displays underneath the Topic name.</p>	<a href="#">Work on Team Review Items</a> [346]
2	<p>Overtyping the <i>New Document</i> text with the name of the Document, and clicking off the name.</p> <p>The text editor displays in the Team Review tab.</p>	
3	Type the text of the Document as required.	<a href="#">Edit an Item</a> [358]

#### Create a Document based on a predefined template

Step	Action	See also
1	<p>Right-click on the appropriate Topic name in the Team Review window and select the <b>New Document from template</b> context menu option.</p> <p>A new Document icon displays in the hierarchy.</p>	<a href="#">Work on Team Review Items</a> [346]

Step	Action	See also
2	Overtyping the <i>New Document</i> text with the name of the Document, and clicking off the name.  The New Document from Template dialog displays.	
3	Click on the drop-down arrow in the <b>Template</b> field, and select a predefined template for the Document contents.	
4	Click on the <b>OK</b> button.  The template structure is displayed in the Team Review tab.	
5	Enter the text of the Document.	<a href="#">Edit an Item</a> <sup>358</sup>

#### Create a Document based on a link to an external file

Step	Action	See also
1	Open a file browser (such as Windows Explorer) and locate the file.	
2	Click on the file name and drag it onto the appropriate Topic name in the Team Review window.  A new Document icon is created underneath the selected Topic, and the body of the Document is shown in the Team Review tab.  A link to the source file is created at the start of the message.	
3	Click twice on the <i>New Document</i> text in the Team Review window, and overtype the text with the name of the Document.	
4	In the Team Review tab, create and edit text around the file link, and add further links if required.	<a href="#">Edit an Item</a> <sup>358</sup>

#### When you have created the Document

You can	Detail	See also
<b>Create links from the Document</b>	To elements and diagrams from the: <ul style="list-style-type: none"> <li>• Project Browser</li> <li>• Model Search</li> <li>• Model Views window</li> <li>• Package Browser</li> </ul>	<a href="#">Add Object Links</a> <sup>[359]</sup> <a href="#">The Project Browser</a> <sup>[646]</sup> <a href="#">Model Search</a> <sup>[700]</sup> <a href="#">Model Views</a> <sup>[686]</sup> <a href="#">Package Browser</a> <sup>[673]</sup>
	To related Team Review: <ul style="list-style-type: none"> <li>• Categories</li> <li>• Topics</li> <li>• Documents</li> </ul>	<a href="#">Add Object Links</a> <sup>[359]</sup>
	To external files, either in the text of the Document or on the Category, Topic or Document in the Team Review window (the link is added to the Model Links folder for the Team Review item).	<a href="#">Add Object Links</a> <sup>[359]</sup>
<b>Add resources to the Document</b>	It is simple and convenient to link to the resources that are the subject of discussion or that illustrate points in the discussion.	<a href="#">Team Review Resources</a> <sup>[361]</sup>
<b>Receive Comments on the Document from other users</b>	The objective of the Team Review is to discuss aspects of development, so you can expect to receive Comments on the Documents that you have posted, and to create Comments on the material posted by your colleagues.	<a href="#">Comment on a Document</a> <sup>[355]</sup>

### 3.6.3.5 Comment on a Document

You use the Team Review as a forum for discussion of items and issues within the project, so when you read a Document on a particular point (or a response to that Document) you might want to respond with your own Comment.

**Access** [View | Team Review](#) ( Ctrl+Alt+U )

#### Create a Comment in the Team Review

Step	Action	See also
1	<b>Either:</b> <ul style="list-style-type: none"> <li>• Right-click on the required Document (or Comment) in the Team Review window and select the <b>New Comment</b> context menu option</li> <li>• Click on the Document name and on the <b>New Comment</b> icon in the Team Review toolbar, or</li> </ul>	<a href="#">Work on Team Review Items</a> <sup>[346]</sup>

Step	Action	See also
	<ul style="list-style-type: none"> <li>Click on the Document name and press <b>Ctrl+N</b></li> </ul> <p>A <i>Re:&lt;Documentname&gt;</i> entry displays underneath the Document you are replying to, and the cursor becomes active in the Team Review tab so that you can create and edit your response.</p>	
2	Type in, format and save the contents of the Comment.	<a href="#">Edit an Item</a> <sup>[358]</sup>
<b>Alternatively</b>		
1	Open a file browser (such as Windows Explorer) and locate a file containing the material you want to use in your response.	
2	<p>Click on the file name and drag it into the Team Review window, over the Document to which you are replying.</p> <p>A prompt displays for you to specify whether to:</p> <ul style="list-style-type: none"> <li>Create a new Comment containing the text of the file; the file name displays as the Comment name; or</li> <li>Add the file to the original Document as a Model Link</li> </ul>	<a href="#">Add Object Links</a> <sup>[359]</sup>
3	Edit, format and save the contents of the reply.	<a href="#">Edit an Item</a> <sup>[358]</sup>

**When you have created the Comment**

You can	Detail	See also
<b>Create links from the Comment</b>	<p>To elements and diagrams from the:</p> <ul style="list-style-type: none"> <li>Project Browser</li> <li>Model Search</li> <li>Model Views window</li> <li>Package Browser</li> </ul>	<a href="#">Add Object Links</a> <sup>[359]</sup> <a href="#">The Project Browser</a> <sup>[646]</sup> <a href="#">Model Search</a> <sup>[700]</sup> <a href="#">Model Views</a> <sup>[686]</sup> <a href="#">Package Browser</a> <sup>[673]</sup>
	<p>To related Team Review:</p> <ul style="list-style-type: none"> <li>Categories</li> <li>Topics</li> <li>Documents</li> </ul>	<a href="#">Add Object Links</a> <sup>[359]</sup>

You can	Detail	See also
	To external files, either in the text of the Comment or on the parent Category, Topic or Document in the Team Review window (the link is added to the Model Links folder for the Team Review item).	<a href="#">Add Object Links</a> <sup>[359]</sup>
<b>Add resources to the Comment</b>	It is simple and convenient to link to the resources that are the subject of discussion or that illustrate points in the discussion.	<a href="#">Team Review Resources</a> <sup>[361]</sup>
<b>Receive replies to the Comment from other users</b>	The objective of the Team Review is to discuss aspects of development, so you can expect to receive Comments on the Documents that you have posted, and to create Comments on the material posted by your colleagues.	

#### Learn More

- [Team Review Tools](#) <sup>[343]</sup>
- [Work on Team Review Items](#) <sup>[346]</sup>
- [Add a New Document](#) <sup>[353]</sup>

### 3.6.3.6 Protection Against Editing

When you have created a Team Review Category, Topic or Document containing text, the item text is exposed to change by any user who has access to the Team Review. You can protect the item against unrestricted changes by setting a password on it. The password only refers to that item; to protect the text of Topics, Documents or Comments subordinate to the item, you set a password on each of them as well.

When the password is set, only those users who have the password can edit that item, and only when they specifically enter the password to unlock that item. Any other user of the Team Review can still **view** the protected item.

The password does not protect the item from operations on the item as a whole, including being deleted.

#### Notes

- The use of a password on an item is indicated by an exclamation mark against the item in the Team Review window; a blue icon indicates that a user with the password has opened the item, and a red icon indicates that the item is locked
- When you click on an item to set a password, the item opens; you can edit the item after you have set the password, while it is still open
- Once a password-protected item is closed, any user who wants to edit the item must first select to unlock the item for editing and then enter the password set on the item
- Unlocking is specific to the user; the item remains locked to other users until they enter the password themselves
- If the password becomes unnecessary, you can remove it from the item

#### Learn more

- [Work on Team Review Items](#) <sup>[346]</sup>

### 3.6.3.7 Edit an Item

When you have created a Category, Topic, Document or Comment, you edit the text within it using the Team Review editor. This is based on the Document editor, which is used to edit report documents and linked documents throughout Enterprise Architect.

You can also change the **name** of the item by selecting it, pressing **F2** and typing the new name.

**Access**    **View | Team Review | <item>**

#### Option Descriptions

The editor menu option descriptions are grouped according to the actions you are performing.

Actions	Link
Creating and importing documents	<a href="#">File Control</a> <sup>[1050]</sup>
Configuring the editor page display and formatting tools shown	<a href="#">Editor Tool Display Options</a> <sup>[1052]</sup>
Incorporating stylesheets, special texts and Tables of Contents Managing the base styles in the Normal.rtf style template file	<a href="#">Styles, Special Texts &amp; Table of Contents</a> <sup>[1054]</sup> <a href="#">The Normal.rtf Style Template</a> <sup>[1058]</sup>
Moving through, searching and selecting text	<a href="#">Scroll, Search and Select Text</a> <sup>[1059]</sup>
Formatting characters and text strings	<a href="#">Format Text</a> <sup>[1062]</sup>
Formatting paragraphs and text blocks	<a href="#">Format Paragraphs</a> <sup>[1064]</sup>
Inserting tab points	<a href="#">Set Tabs</a> <sup>[1067]</sup>
Inserting sections, columns and page breaks, and repaginating	<a href="#">Define Document Sections</a> <sup>[1069]</sup>
Inserting and editing page headers and footers, and footnotes and endnotes	<a href="#">Insert Headers, Footers, Footnotes and Endnotes</a> <sup>[1071]</sup>
Inserting tables	<a href="#">Create Tables</a> <sup>[1075]</sup>
Setting up User-Defined Section Numbering (in document report templates)	<a href="#">Apply User-Defined Section Numbering</a> <sup>[1081]</sup>

Actions	Link
Inserting hyperlinks and bookmarks (including using the <b>New</b> , <b>Link to Existing Element</b> and <b>Hyperlink</b> options on the <b>Create</b> context menu)	<a href="#">Insert Reference Links</a> <sup>[1083]</sup> <a href="#">Add Object Links</a> <sup>[359]</sup> <a href="#">Hyperlinks</a> <sup>[2002]</sup> <a href="#">Create Element From Document</a> <sup>[1096]</sup> <a href="#">Hyperlink From Linked Document</a> <sup>[1095]</sup>
Inserting images, OLE objects, frames and drawing objects	<a href="#">Insert Images, Objects and Frames</a> <sup>[1085]</sup>
Printer setup and printing documents	<a href="#">Print Report Documents</a> <sup>[1094]</sup>
Check spelling and use of terms	<a href="#">Checking Text</a> <sup>[1089]</sup>
Tracking, accepting and rejecting changes to the text	<a href="#">Track Changes</a> <sup>[1091]</sup>
Protecting document text from accidental change	<a href="#">Protect Document Contents</a> <sup>[1093]</sup>
Create and refer to definitions of Glossary terms in the Project Glossary (using the <b>Create   Glossary Definition</b> context menu option)	<a href="#">Project Glossary View</a> <sup>[534]</sup>

### Notes

- Throughout your document editing:
  - To undo one or more immediately previous edits, press **(Ctrl+Z)**, or select the **Edit | Undo** menu option; you can still undo a change even after you have saved the change
  - To redo one or more immediately previous undone edits, press **(Ctrl+Y)**, or select the **Edit | Redo** menu option

### 3.6.3.8 Add Object Links

In the Team Review window you can create hyperlinks from a Document to the elements and diagrams that are associated with it. This helps you to:

- Rapidly navigate to the objects in the Project Browser
- Access the element properties
- Open a diagram directly from the Team Review

You can also create links to:

- Other Categories, Topics and Documents in the Team Review window
- External files from a file browser

**Facilities**

Facility	Action	See also
<b>Associate an element, diagram, external file or Team Review item with a message</b>	<p>Drag the object over the required Category, Topic or Document in the Team Review window, from the:</p> <ul style="list-style-type: none"> <li>• Project Browser</li> <li>• Package Browser</li> <li>• Model Views window</li> <li>• Model Search dialog or</li> <li>• External file browser</li> </ul> <p>To create a link to another Team Review item, press <b>Ctrl</b> as you drag that item over the required Category, Topic or Document.</p> <p>If it does not already exist, a <i>Model Links</i> folder is created under the selected Team Review item, and the link to the dragged object is added to the folder.</p>	<p><a href="#">The Project Browser</a><sup>[646]</sup></p> <p><a href="#">Package Browser</a><sup>[673]</sup></p> <p><a href="#">Model Views</a><sup>[686]</sup></p> <p><a href="#">Model Search</a><sup>[700]</sup></p>
<b>Associate an external file with the message text</b>	<p>From any browser, click on and drag the file name into the text of the message.</p> <p>The filename becomes a link to the file; click on it to display the contents of the file.</p> <p>The external file name also becomes a link to the file within the message when you drag the filename onto a Topic to create a Document.</p>	<p><a href="#">Add a New Document</a><sup>[353]</sup></p>

To review an object in the *Model Links* folder, right-click on the object to display the navigation context menu.

Option	Action	See also
<b>Open</b>	<p>Open the linked diagram or external file.</p> <p>Alternatively, double-click on the diagram link.</p>	
<b>Properties</b>	<p>Display the Properties dialog for the selected element or diagram.</p> <p>Alternatively, double click on the element, attribute or operation link to open the Properties dialog of the object.</p>	
<b>Create (Edit) Linked Document</b>	<p>Either:</p> <ul style="list-style-type: none"> <li>• Open the New Linked Document from Template dialog, to begin creating a new Linked Document on the element, or</li> <li>• Open the Linked Document Editor to change an existing Linked Document</li> </ul>	



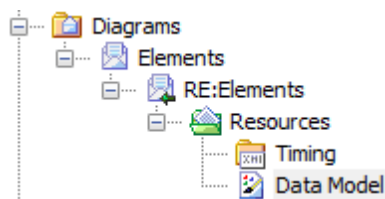
Option	Action	See also
<b>Find in all Diagrams</b>	Open the diagram in which the element or diagram is used, or display a list of several diagrams in which the element or diagram has been used.	
<b>Find in Project Browser</b>	Highlight the element or diagram in the Project Browser.	
<b>Locate Item</b>	Highlight, in the Team Review window, the Team Review item linked to the message.  Alternatively, double-click on the link in the Model Links folder.	
<b>Delete Model Link &lt;name&gt;</b>	Delete the association between the Document and the object.	

### 3.6.3.9 Team Review Resources

Whilst having a discussion through the Team Review, it is convenient to link to the resources that are the subject of discussion or that illustrate points in the discussion. Such resources include:

- XML files of Packages within the current project
- Active Profiler reports
- Images of currently-active diagrams
- Images currently captured on the clipboard
- Bookmarks to currently active websites

You can link to such resources from Category or Topic notes, or from a Document. The resource links are created in a *Resources* folder underneath the selected Category, Topic or Document, as shown:



**Access**    **View | Team Review | Right-click on <object> | Share Resource | <resource type>**

You create the *Resources* folder by creating a link to a resource underneath the selected Category, Topic or Document. Similarly, you delete the *Resources* folder by deleting the last resource within it. Having added a resource, you can right click on it and use a further context menu to:

- Reimport the Package XML files to the model or
- Display the images


Learn more

- [Generate, Save and Load Profile Reports](#) <sup>[2560]</sup>
- [XML Import and Export](#) <sup>[473]</sup>
- [Work on Team Review Items](#) <sup>[346]</sup>

**3.6.3.10 Search Team Review**

The Team Review provides the facility to search the *titles* of all Team Review items, to locate items referring to a specific subject.

Search the Team Review

Step	Action	See also
1	In the Team Review window toolbar, click on the  ( <b>Search</b> ) icon. The search panel displays underneath the toolbar.	
2	In the blank field, type the text string to search for.	
3	If required, select the <b>Match Case</b> checkbox to locate text with the same case as the search string.	
4	If required, select the <b>Match Whole Word</b> checkbox to locate only complete words that match the search string.	
5	Click on the <b>Find</b> button. The search locates the first instance of the search string in the title of a Category, Topic, Document or Comment, and displays the contents of that item in the Team Review tab.	
6	To locate further instances of the text string, click on the <b>Find</b> button again.	
7	To close the search panel, click on the <b>Search</b> icon in the toolbar again.	

**3.6.3.11 Team Review Options**

You can change the loading behavior of the Team Review, using the Team Review Server Options dialog. If you change the settings, the new loading behavior is not applied until you restart the Team Review.

Access    **Right-click Team Review window | Options**

Field/Option/ Button	Action	See also
<b>Preload items less than</b>	Type the threshold item size (in KB) below which the items will be preloaded and above which item data is only loaded when selected.	
<b>Load item data when required</b>	The fastest loading option. Team Review data is only loaded on demand; for example, when you read a Document.	
<b>Preload all data</b>	Caches the entire contents of the Team Review on load; this takes longer to load but, once completed, navigating the Team Review is faster.	
<b>Mark All Unread</b>	Resets all Documents you have read to 'unread' (bolds all items in the Team Review).	

### 3.6.3.12 Team Review Connections

If you are modeling in a different model from the rest of your team, you can switch to Team Reviews from **other** Enterprise Architect models, including models located on servers. This connects only to the **Team Review** for the selected model; it does not change the model open in the **Project Browser** or anywhere else in the system.

**Access** Through the **Connection Options** drop-down field in the **Team Review window Toolbar**  
**Right-click on the Team Review Window | Connections**

#### Switch to another Team Review

Step	Action	See also
<b>1</b>	In the Team Review window toolbar, click on the drop-down arrow in the <b>Connection Options</b> field and select the appropriate model name from the list, to connect to the Team Review for that model.  If the required model is not listed, click on the <b>&lt;Configure Connections&gt;</b> option; the Team Review Server Connections dialog displays.  Go to <b>Step 3</b> .	
<b>2</b>	<i>Alternatively:</i>  Right-click anywhere in the Team Review window and select the <b>Connections</b> context menu option; the Team Review Server Connections dialog displays.  In the list in the Connections panel, select the check box against the appropriate model name to connect to the Team Review for that model, and click on the <b>Open</b> button.  The connection now switches to the Team Review in the selected model.  If the required model is not listed, go to <b>Step 3</b> .	

Step	Action	See also
3	<p>Select the appropriate <b>Connection Type</b> radio button and click on the <b>New</b> button.</p> <ul style="list-style-type: none"> <li>For a project <b>file</b>, a browser dialog displays through which you can search for and select the appropriate model</li> <li>For a model in a <b>DBMS</b> data repository, the Microsoft Data Link dialogs display, through which you can locate and connect to the repository</li> </ul>	
4	When you have selected and opened or connected to the required Enterprise Architect model, and returned to the Team Review Server Connections dialog, the model name displays in the <b>Connection Name</b> field and in the Connections panel.	
5	<p>Select the check box against the model name and click on the <b>Open</b> button to connect to the Team Review for that model.</p> <p>The Team Review now shows the discussions in the selected model.</p>	

#### Fields/Options on the Team Review Server Connections dialog

Field/Option	Action	Shortcut	See also
<b>Connection Name</b>	Verify the name of the selected model.		
<b>Connection Type</b>	Select the appropriate radio button for the type of Enterprise Architect model to locate: a local project file or a model on a remote server.		
<b>Target Model</b>	Verify the path to the selected model.		
<b>New</b>	Click on this button to display the file browser or Data Link Properties dialog to select the model for a new Team Review connection.		
<b>Delete</b>	Click on this button to delete the currently selected connection from the Connections list. There is no confirmation prompt; the connection is immediately removed from the list.		
<b>Connections</b>	View all Team Review connections created; click on the checkbox against the required connection.		

Field/Option	Action	Shortcut	See also
<b>Open</b>	Click on this button to switch the <b>Team Review</b> to the selected connection and to close the dialog.		

### 3.6.4 Element Discussions



Using the dockable **Element Discussions window** you can develop a discussion on, or review of, an individual element that is **directly associated** with that element. This makes it easy to locate, view and contribute to a discussion, or more than one discussion, on the selected element.



You can review and participate in discussions on different elements simply by clicking on each element in turn; the Element Discussions window immediately switches to the discussions of the new selected element.

As a discussion develops, the number of responses is shown next to the original Document, in brackets; therefore you can see when a discussion has a new reply without having to expand and work through the thread. You can also set a status of **Awaiting Review** or **Closed** on a discussion, indicated by an icon against the posting. This helps you to resolve and close off points raised in the discussion, so that they do not continue indefinitely. You can also delete the whole posting or specific replies, to balance retaining discussion history with removing irrelevant information.

**Access**    **select element | Element | Discussions**

#### Operations in Element Discussions

Operation	Action	See also
<b>Create a discussion</b>	<p>The window displays the message <i>Create new Discussion</i>. Double-click on this message and type in your posting as continuous text (no carriage returns).</p> <p>When you have finished typing, click off the text or press <b>Enter</b>.</p> <ul style="list-style-type: none"> <li>• An open expansion box, a  icon, your user ID and the date and time of posting display above your text</li> <li>• The messages <i>Post Reply</i> and <i>Create new Discussion</i> display beneath your text</li> </ul> <p>You can also right-click on the window and select the <b>Create new Discussion</b> menu option. All new discussions are added underneath existing discussion threads.</p>	
<b>Add a response</b>	<p>Double-click on the <i>Post Reply</i> message and type your response in as continuous text.</p> <p>When you have finished typing, click off the text or press <b>Enter</b>.</p> <ul style="list-style-type: none"> <li>• A  icon, your user ID and the date and time of posting display above your text</li> <li>• The message <i>Post Reply</i> displays beneath your text</li> <li>• The number of responses posted displays or is incremented to the left of the original posting</li> </ul>	

Operation	Action	See also
	<p>You can also right-click on a post and select the <b>Post Reply</b> menu option. All new discussions are added underneath existing discussion threads.</p>	
<b>Set discussion status</b>	<p>The icon against a new discussion indicates that the discussion has the status of <b>Open</b>. To change the status, right-click on the discussion item and select either:</p> <ul style="list-style-type: none"> <li>• <b>Discussion Status   Awaiting Review</b> (  ) or</li> <li>• <b>Discussion Status   Closed</b> (  )</li> </ul> <p>If the discussion already has one of these statuses, you can also select the <b>Open</b> option to return the discussion to <b>Open</b> status.</p> <p>The status settings are simply labels - they do not have an impact on posting replies, deleting replies or deleting the discussion.</p>	
<b>Edit postings</b>	Once a discussion or reply has been posted, you cannot edit it.	
<b>Delete a discussion or reply</b>	<p>Right click on the original discussion or a reply, and select the <b>Delete Selected</b> menu option. A prompt displays to confirm the deletion. Click on the <b>Yes</b> button.</p> <p>If you delete a discussion, all replies associated with that discussion will be deleted as well.</p> <p>If you delete a reply in a thread, the count of replies against the original posting is decremented.</p>	
<b>Review Discussion History</b>	<p>Click on the History tab to display a table of <b>all</b> existing discussion items on elements. You can:</p> <ul style="list-style-type: none"> <li>• Add to or reduce the number of columns using the <b>Field Chooser</b>, and display the <b>Filter bar</b> to filter the list for discussions having a specific property such as the user who <b>posted the discussion item</b>, or the person who created the <b>element</b> (element author)</li> <li>• List the discussions posted over a number of days - either <b>Today</b> or <b>7</b>, <b>30</b> or <b>90</b> days prior to today, by right-clicking on the display and selecting the <b>Visible Timeframe</b> option</li> <li>• Locate the element in a discussion in the Project Browser or diagrams</li> <li>• Open a discussion to read the full text and replies in a review window; you cannot operate on or delete the discussions in this window</li> </ul>	<a href="#">Customizing the Search View</a> <sup>708</sup>

[Learn more](#)

- [Team Review Tools](#) <sup>[343]</sup>

### 3.6.5 Workflow Scripts

Workflow scripts validate user work and actions against the policy and procedures within your model, providing a robust approach to applying company policy and strengthening project development guidelines.

Project Administrators can write workflow scripts to manage the way users interact with a model, such as managing security, staff compliance and model access, and monitoring changes made by users. Administrators can also use the scripts to control a user's capacity to change a model element, taking into account factors such as access rights, group membership and even the value of a proposed change.

#### **Access Tools | Scripting**

#### **Application of Workflow scripts**

Topic	Detail	See also
<b>Project Governance</b>	<p>Good corporate governance relies on well written and transparent project development guidelines and company policy.</p> <p>A project might be compromised if the appropriate policies and procedures are poorly understood and not followed correctly - effective governance can be hampered by human error and the costs of recovering from the inadequate compliance of developers.</p>	
<b>Policies, Procedures and Development</b>	<p>Company policy and procedures can be integrated with the development process to manage work flows, determine access rights, extend role based security permissions and respond to property change events.</p> <p>This approach reduces compliance costs, enhances collaborative development and gives you confidence that projects are being developed correctly the first time around.</p> <p>Development teams can adhere to best practice guidelines that govern model validation, change management, access controls and general development principles.</p>	
<b>Script Implementation</b>	<p>When a model is launched, the Workflow Engine is initialized with the current user and group memberships; this information determines who can access and modify parts of a given model.</p> <p>When a selected event occurs, the script engine is initialized with values including the author's name and access rights, and the element name and version details.</p> <p>The workflow script implements rules governing change management, access control and model validation; if a user attempts to make changes in violation of company policy, the script denies the update.</p> <p>The user is notified why the validation failed and the activity is logged.</p> <p>These reminders help to reinforce company policy, reduce human error and provide management with valuable project feedback.</p>	

**Notes**

- Workflow Scripting is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect
- Workflow Scripting requires User Security to be enabled in order to function
- You need **Admin Workflow** permission to develop and manage workflow scripts

**Learn more**

- [Workflow Script Functions](#) <sup>[368]</sup>
- [Scripting](#) <sup>[2791]</sup>
- [List of Available Permissions](#) <sup>[3291]</sup>

**3.6.5.1 Workflow Script Functions**

Workflow scripts are created in the Scripting window, under the Workflow group type as VBScripts. They are executed by the Enterprise Architect workflow engine, to manage user input.

You can make use of a range of functions and data structures to develop your scripts.

**Access** **Tools | Scripting**

**Workflow functions and data structures**


Topic	Detail	See also
<b>Functions for User Input</b>	These are functions that Enterprise Architect calls to validate and control user input.  For each of the functions that Enterprise Architect calls, a set of objects are filled.	<a href="#">Functions - Validate and Control User Input</a> <sup>[369]</sup>  <a href="#">Filled Workflow Data Structures</a> <sup>[371]</sup>
<b>Functions to create a Search</b>	These are functions that Enterprise Architect calls to create a search with user tasks.	<a href="#">Functions - Create a Search With User Tasks</a> <sup>[370]</sup>
<b>Workflow Data Structures Enterprise Architect fills</b>	These are workflow data structure objects that Enterprise Architect fills.	<a href="#">Filled Workflow Data Structures</a> <sup>[371]</sup>
<b>Workflow Data Structures you fill</b>	These are Workflow data structure objects that you can fill.	<a href="#">Workflow Data Structures You Fill</a> <sup>[372]</sup>
<b>Functions you call</b>	These are functions that Enterprise Architect provides for you to call.	<a href="#">Functions You Call</a> <sup>[373]</sup>



**Notes**

- If you make changes to a workflow script listed in the Scripting window, click on the **Refresh Scripts** button in the Scripting window toolbar to reload the script with the changes

**Learn more**

- [Scripting](#) 
- [Script Group Properties](#) 

**3.6.5.1.1 Functions - Validate and Control User Input**

These are functions that Enterprise Architect calls to validate and control user input. For each function a set of objects is filled.

Function	Action	Return Value
<b>AllowPhaseUpdate(OldValue, NewValue)</b>	Validate a change a user has made to a phase	<ul style="list-style-type: none"> <li>• <b>True</b> to allow this user to make this change</li> <li>• <b>False</b> to disallow the change and revert to the previous value</li> </ul>
<b>AllowStatusUpdate(OldValue, NewValue)</b>	Validate a change a user has made to a status	
<b>AllowTagUpdate(TagName, OldValue, NewValue)</b>	Validate a change a user has made to a Tagged Value	
<b>AllowVersionUpdate(OldValue, NewValue)</b>	Validate a change a user has made to a version	
<b>CanEditPhase()</b>	Enable or disable the control for editing a phase	<ul style="list-style-type: none"> <li>• <b>True</b> to allow this user to make changes by enabling the control</li> <li>• <b>False</b> to completely disable edit of this property by disabling the control</li> </ul>
<b>CanEditStatus()</b>	Enable or disable the control for editing a status	
<b>CanEditTag(TagName)</b>	Enable or disable the control for editing a Tagged Value	
<b>CanEditVersion()</b>	Enable or disable the control for editing a version	
<b>OnPreNewElement(</b>	Allow or disallow the creation of the specified	

Function	Action	Return Value
<b>ElementType, ElementStereotype)</b>	element.	<ul style="list-style-type: none"> <li>• <b>True</b> to allow this user to create the element/connector</li> <li>• <b>False</b> to prevent this user from creating the element</li> </ul>
<b>OnPreNewConnector( ConnectorType, ConnectorSubType, ConnectorStereotype)</b>	Allow or disallow the creation of the specified connector.	
<b>PreAllowPhaseUpdate( OldValue, New Value)</b>	Determine what information is required to validate this change	Semi-colon separated list of additional data required in order to validate this change.  <i>Supported Data Type:</i>  <b>Tests</b> - fill the Tests array in the WorkflowContext object.
<b>PreAllowStatusUpdate( OldValue, New Value)</b>		
<b>PreAllowTagUpdate(TagName, OldValue, New Value)</b>		
<b>PreAllowVersionUpdate( OldValue, New Value)</b>		

#### Learn more

- [Filled Workflow Data Structures](#) 

#### 3.6.5.1.2 Functions - Create a Search With User Tasks

These are functions that Enterprise Architect calls to create a search with user tasks.

Function	Action	Return Value
<b>GetWorkflowTasks</b>	Describe the searches that this user must run.	Ignored

### 3.6.5.1.3 Filled Workflow Data Structures

These workflow data structures are objects that Enterprise Architect fills.

Workflow Data Structure	Description	Calls	See also
<b>WorkflowUser</b>	<p>This object provides information about the user currently logged in to the model.</p> <p>It is filled by Enterprise Architect before any function is called by Enterprise Architect; it has the following properties:</p> <ul style="list-style-type: none"> <li>• <b>Username</b> - the username for login to the system (if using Windows Authentication, this matches the Windows username)</li> <li>• <b>Firstname</b> - as found in the Security Users dialog</li> <li>• <b>Surname</b> - as found in the Security Users dialog</li> <li>• <b>Fullname</b> - the combination &lt;<b>Firstname</b>&gt; &lt;<b>Surname</b>&gt; (the form Enterprise Architect uses for <b>Author</b> fields and similar)</li> </ul>	<p>This object calls the <b>IsMemberOf (GroupName)</b> function (see below).</p>	<p><a href="#">Maintain Users</a> [323]</p>
<b>WorkflowContext</b>	<p>This object provides information about the object currently in context.</p> <p>It is filled by Enterprise Architect before any searches except <b>GetWorkflowTasks</b> are run; it has the following properties:</p> <ul style="list-style-type: none"> <li>• <b>MetaType</b> - the type of the current object, either an Enterprise Architect core type or a profile-specified metatype</li> <li>• <b>Name</b> - as found in the object Properties dialog</li> <li>• <b>Status</b> - as found in the object Properties dialog</li> <li>• <b>Phase</b> - as found in the object Properties dialog</li> <li>• <b>Version</b> - as found in the object Properties dialog</li> <li>• <b>Stereotypes</b> - an array of strings for the stereotypes applied to this object</li> <li>• <b>Tags</b> - an array of Tagged Values, providing: <ul style="list-style-type: none"> <li>• <b>Name</b> - the Tagged Value name</li> <li>• <b>Value</b> - the Tagged Value value</li> </ul> </li> <li>• <b>Tests</b> - an array of tests; only filled during an <i>Allow*</i> call after the <i>PreAllow*</i></li> </ul>	<p>This object calls the <b>TagValue (TagName)</b> function (see below).</p>	<p><a href="#">Functions - Create a Search With User Tasks</a> [370]</p> <p><a href="#">Working on Test Records</a> [2605]</p>

Workflow Data Structure	Description	Calls	See also
	<p>call has specified that tests are required; provides the following details, as found in the Testing window:</p> <ul style="list-style-type: none"> <li>• <b>Name</b></li> <li>• <b>Status</b></li> <li>• <b>RunBy</b></li> <li>• <b>CheckedBy</b></li> <li>• <b>TestClass</b></li> <li>• <b>TestType</b></li> </ul>		

Function	Action	Return Value
<b>IsMemberOf (GroupName)</b>	Check the group membership of the current user.	Returns the value <b>True</b> if the current user is a member of the group with the specified name.
<b>TagValue (TagName)</b>	Get the value from a named tag.	Returns the value of the first Tagged Value with that name, or an empty string if no Tagged Value with that name exists.

#### 3.6.5.1.4 Workflow Data Structures You Fill

These two workflow data structures are objects that you can fill.

Workflow Data Structure	Description
<b>WorkflowStatus</b>	<p>Use this data structure to provide information on the status of the object.</p> <ul style="list-style-type: none"> <li>• <b>LogEntry</b> - set to <b>True</b> or <b>False</b> to indicate whether or not a log item should be recorded</li> <li>• <b>Reason</b> - indicate what reason should be recorded in the log</li> <li>• <b>Action</b> - indicate how to display the log message; valid values are: <b>MessageBox</b>, <b>StatusBar</b>, <b>Output</b> (default)</li> </ul>
<b>WorkflowSearches</b>	<p>Use this data structure to provide an array of searches.</p> <p>Use <b>Redim WorkflowSearches(x)</b> to specify the number of searches being provided.</p> <p>Each search has the following attributes:</p> <ul style="list-style-type: none"> <li>• <b>Name</b> - the name of this search</li> <li>• <b>Group</b> - the name of the group that this search should appear under in the <b>Search</b> combo box</li> </ul>

Workflow Data Structure	Description
	<ul style="list-style-type: none"> <li>• <b>ID</b> - the GUID for this search</li> <li>• <b>Tasks</b> - the array of tasks that this search looks for; an entry describes how to find all objects required to meet a particular task: <ul style="list-style-type: none"> <li>• <b>Name</b> - the name of the task, as displayed in the Search view; workflow searches are grouped by this field by default</li> <li>• <b>Conditions</b> - an array of conditions, all of which must be matched for an object to be included in this task; a condition is a comparison of a single field to a value: <ul style="list-style-type: none"> <li>• <b>Column</b> - the name of the field</li> <li>• <b>Operator</b> - operator types, either = (provide matching values only) or &lt;&gt; (provide non-matching values only)</li> <li>• <b>Value</b> - if this contains a comma, the string is treated as a comma separated list of values to compare against; otherwise the string is a single value to compare against</li> </ul> </li> </ul> </li> </ul>

#### 3.6.5.1.5 Functions You Call

These are functions that Enterprise Architect provides for you to call.

Function	Action	Return Value	See also
<b>NewSearch(name, group, guid, taskcount)</b>	Create a new search object to be included in <b>WorkflowSearches</b> . Initialize each member.	The created search	<a href="#">Workflow Data Structures You Fill</a> <sup>[372]</sup>
<b>NewTask(name, conditioncount)</b>	Create a new task object to be included in a search. Initialize each member.	The created task	
<b>NewCondition(column, operator, value)</b>	Create a new condition object to be included in a task. Initialize each member.	The created condition	
<b>SetLastError(message, outputMethod)</b>	Called on user input to the following element properties: <ul style="list-style-type: none"> <li>• Status</li> <li>• Phase</li> <li>• Version, and</li> <li>• Tagged Values</li> </ul> <p>It logs and/or reports the provided message to the user. It can be called within the functions:</p>	The message	

Function	Action	Return Value	See also
	<ul style="list-style-type: none"> <li>• AllowPhaseUpdate</li> <li>• AllowStatusUpdate</li> <li>• AllowTagUpdate</li> <li>• AllowVersionUpdate</li> <li>• preAllowPhaseUpdate</li> <li>• preAllowStatusUpdate</li> <li>• preAllowTagUpdate</li> <li>• preAllowVersionUpdate</li> </ul> <p>For example:</p> <pre> public function AllowPhaseUpdate ( OldValue, NewValue )      AllowPhaseUpdate = false      Set Last Error " No updating to phase allowed", "messagebox"  end function </pre> <p>Parameters:</p> <p>message: Text</p> <p>outputMethod can be "messagebox", "statusbar" or "outputwindow".</p> <p>outputmethod is case sensitive.</p>		

### 3.6.6 Sharing Reference Data

You can conveniently update your models with reference data (including Glossary and Issue information) by exporting the data to and importing the data from .XML files to (for example):

- Copy glossaries from one model to another
- Add additional profiles by merging new stereotypes into the model
- Update reference data from files supplied by Sparx Systems as a maintenance release
- Copy resources, clients and so on from one model to another

You import data into the model automatically or manually from a reference data .XML file, exported from another model or an iteration of the current model.

#### Learn more

- [Reference Data](#) <sup>[1146]</sup>
- [Import Reference Data](#) <sup>[380]</sup>

- [Export Reference Data](#)<sup>[376]</sup>
- [Link Reference Data to a Shared Repository](#)<sup>[375]</sup>

### 3.6.6.1 Link Reference Data to a Shared Repository

If you have a number of projects that use the same sets of reference data, you can use a central project containing that common data as a **shared repository**. The common data includes system resources such as Security Users and Groups, Glossary terms or Data Types. Shared repositories make it possible for teams working on multiple projects to leverage a common and consistent set of system resources. Common standards and definitions can be applied across all projects using this repository. Anytime a new project is started, that project can automatically inherit some or all of these common resources from the repository; there is no need to establish these resources again for each project.

Shared repositories apply to Database Management System (DBMS) based Enterprise Architect projects and can link all projects residing within one DBMS.

A new DBMS based project in Enterprise Architect is initially created with its own stand-alone repository. A single step process then allows that new project to be linked to the shared repository residing on its DBMS. Any older or pre-existing Enterprise Architect projects can also be linked to the shared repository, taking their resource data from the master version. This then allows for centralized management of the contents of the repository, promoting or enforcing common usage and standards.

Any changes to the shared repository are automatically available to every participating project.


**Access**   **Tools | Data Management | Shared Repository**

#### Set up a shared repository

Step	Action	See Also
1	Create the script file to set up the shared repository, as explained in the next table.	
2	Execute the generated SQL script using a third-party tool.	

#### Select the repository tables to share

Field/Button	Action	See also
<b>Script File</b>	Type in or browse ( ... ) to the location into which the script file will be generated by this dialog.	
<b>Main Repository</b>	Type in the name of the current main repository.	
<b>Shared Repository</b>	Type in the name of the database to be used as the shared repository.  This name cannot be the same as the name of the main repository.	

Field/Button	Action	See also
<b>Table Groups</b>	In the list of table groups that can be linked to the shared repository, select the checkbox against each table group to share in the repository.	
<b>Tables in each group</b>	(Read only) Review the list of database tables in the currently-selected table group. All of these tables in the selected group will be available through the shared repository.	
<b>Generate</b>	Click on this button to generate the SQL script to link the selected tables of the current (main) repository to the shared repository. The script is generated into the file location you specified in the <b>Script File</b> field.	
<b>View Script</b>	Click on this button to open the generated script file in the default Code Editor, and to close the dialog.	<a href="#">Editing Source Code</a> 
<b>Close</b>	Click on this button to close the dialog.	

#### Notes



- This feature is not applicable to projects that are based on local project files (.eap or .feap)
- Shared repositories are supported for MySQL, SQL Server and Oracle

### 3.6.6.2 Export Reference Data

When you have a complete project set up, with fully defined project data, it is possible to capture either selected categories or all categories of that data, and export it to a single XML file. You can then review the data, and import it to another project or tool so that you do not have to define it all again. The exported data includes **all instances** of the selected data **type** in the project; for example, all defined cardinality values, or all Document Style Templates.

**Access** [Project](#) | [Model Import/Export](#) | [Export Reference Data](#)

#### Select and Export Reference Data

Step	Action	See also
1	On the Export Reference Data dialog, click on the '+' box against each table group you want to expand.	
2	Select the checkbox against each table or table group to export.	<a href="#">Constraint Status Types</a>  <a href="#">Constraint Types</a> 



Step	Action	See also
	<p>If you select the top-level, group-name checkbox, all tables in that group are selected for export.</p>	<p> <a href="#">Difficulty Types</a> <sup>[1163]</sup>  <a href="#">Priority Types</a> <sup>[1164]</sup>  <a href="#">Requirement Types</a> <sup>[1166]</sup>  <a href="#">Scenario Types</a> <sup>[1167]</sup>  <a href="#">Status Colors</a> <sup>[1159]</sup> (colors defined for status types)  <a href="#">Status Types</a> <sup>[1159]</sup>  <a href="#">Status Types - Application</a> <sup>[1159]</sup> (elements to which status can be applied)  <a href="#">Test Status Types</a> <sup>[1165]</sup>  <a href="#">Type Default</a> <sup>[1163]</sup> (the defined default value for each General Types data type; for example, as set on the Difficulty tab)    <a href="#">Model Authors</a> <sup>[1153]</sup>  <a href="#">Project Clients</a> <sup>[1157]</sup>  <a href="#">Project Resources</a> <sup>[1156]</sup>  <a href="#">Project Roles</a> <sup>[1155]</sup>    <a href="#">General Effort Types</a> <sup>[522]</sup>  <a href="#">Defined Metric Types</a> <sup>[523]</sup>  <a href="#">Risk Types</a> <sup>[524]</sup>    <a href="#">Defined Problem Types</a> <sup>[1169]</sup>  <a href="#">Maintenance Types</a> <sup>[1169]</sup> (Problem Types plus Test Types)  <a href="#">Test Types</a> <sup>[1170]</sup>    <a href="#">Cardinality Types</a> <sup>[1151]</sup>  <a href="#">Tagged Value Types</a> <sup>[1621]</sup>  <a href="#">Stereotypes</a> <sup>[1461]</sup> (as listed on the Stereotypes page of the UML Types dialog)  <a href="#">UML Patterns (Resources)</a> <sup>[1464]</sup>    <a href="#">Estimation - Environment Complexity Factor Values</a> <sup>[586]</sup>  <a href="#">Estimation - Technical Complexity Factor Values</a> <sup>[585]</sup>  Standard Complexity Types <a href="#">a</a> </p>

Step	Action	See also
		<p><a href="#">Predefined Reference Data Tagged Value Type</a> <sup>[1629]</sup></p> <p><a href="#">Security - Groups</a> <sup>[320]</sup></p> <p><a href="#">Security - Group Permission</a> <sup>[321]</sup></p> <p><a href="#">Security - Users</a> <sup>[323]</sup></p> <p><a href="#">Security User Groups</a> <sup>[326]</sup></p> <p><a href="#">Security - User Permissions</a> <sup>[327]</sup></p> <p><a href="#">Calendar of Events</a> <sup>[574]</sup></p> <p><a href="#">Calendar Event Types</a> <sup>[577]</sup></p> <p><a href="#">Model Images</a> <sup>[860]</sup></p> <p><a href="#">Project Issues</a> <sup>[528]</sup></p> <p><a href="#">Project Glossary</a> <sup>[533]</sup></p> <p><a href="#">Project Tasks</a> <sup>[526]</sup></p> <p><a href="#">Diagram Matrix Profiles</a> <sup>[854]</sup> (Model Profiles)</p> <p><a href="#">Gap Matrix Profiles</a> <sup>[745]</sup></p> <p><a href="#">Linked Document Templates</a> <sup>[1098]</sup> (each of the Linked Document templates in the Resources window)</p> <p>RTF Document Templates (each of the <a href="#">Document Report Style templates</a> <sup>[2684]</sup> in the Resources window)</p> <p>Templates - RTF <a href="#">Project Constants</a> <sup>[2664]</sup></p> <p>Templates - RTF Style (the <a href="#">Legacy RTF Style templates</a> <sup>[2726]</sup> in the Resources window)</p> <p>Templates - RTF Style Detail (the content of the Legacy RTF Style templates)</p> <p>Templates - RTF Tag &amp; Language Options (RTF <a href="#">word substitution</a> <sup>[2725]</sup> templates)</p> <p>Templates - HTML Style (the <a href="#">web templates</a> <sup>[2747]</sup> listed in the Resources window)</p>

Step	Action	See also
		<p>Templates - HTML Style Detail (the content of the HTML report templates)</p> <p>Model Data Types - <a href="#">Code</a><sup>[1171]</sup> and <a href="#">DDL</a><sup>[2350]</sup> (for each programming language supported by Enterprise Architect)</p> <p><a href="#">Macro List</a><sup>[2257]</sup> (Preprocessor macros)</p> <p><a href="#">Import Component Types</a><sup>[2249]</sup></p> <p><a href="#">Automation Scripts</a><sup>[2791]</sup> (JavaScript, JScript and VBScript)</p> <p><a href="#">Team Review</a><sup>[343]</sup></p> <p>Code language templates and/or transformation templates, where <a href="#">templates</a><sup>[1632]</sup> for a particular language or transformation exist in the model</p> <p><a href="#">CSV</a><sup>[498]</sup> Specifications</p>
3	Click on the <b>Export</b> button.	
4	When prompted to do so, enter a valid file name with a . XML extension.	
5	Click on the <b>Save</b> and <b>OK</b> buttons.  This exports the data to the file; you can use any text or XML viewer to examine the file.	

### Notes

- You can resize the Export Reference Data dialog; drag the dialog edges to the size you need
- If there are no instances of a selected data type in the project, the export does not generate any output for that data type in the XML file
- Currently, Standard Complexity Types cannot be directly edited and are therefore effectively standard for **all** models; they can be listed using the Predefined Reference Data Tagged Value type *ComplexityTypes*

Learn more

- [Import Reference Data](#)<sup>[380]</sup>

**3.6.6.3 Import Reference Data**

It is possible to import reference data into your model from an .xml file that was exported from another model or from an iteration of the current model, either:

- Manually, whenever you know there is new or changed data to apply, or
- Automatically whenever the model is reloaded into Enterprise Architect (if the file has changed since the previous import)

The automatic import checks if the source file has changed since the last import; if the file has not changed, the import does not proceed. If the file has changed, the changed data is imported; however, you can configure the system to display a prompt for you to allow or cancel the import.

**Access**    **Project | Model Import/Export | Import Reference Data > Import File**  
**Project | Model Import/Export | Import Reference Data > Shared File**

Import reference data manually (Import File)

Step	Action	See also
1	On the Import Reference Data dialog, click on the Import File tab and on the <b>Select File</b> button, then select the filename to import data from.  This would be an XML file produced by the Enterprise Architect Data Exporter.	<a href="#">Export Reference Data</a> <sup>[376]</sup>
2	If you have selected a valid file, a list of available tables to import displays in the Select Datasets to Import panel.	
3	Click on one or more of the tables to import.  Press <b>(Ctrl)</b> or <b>(Shift)</b> to click on multiple tables.	
4	Click on the <b>Import</b> button to start the process.  A message displays when the import is complete; generally the process is quite fast.	

Import reference data automatically (Shared File)

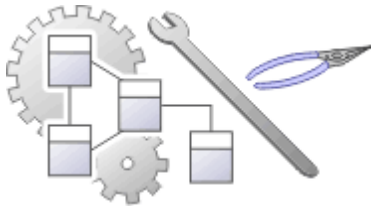
Step	Action	See also
1	On the Import Reference Data dialog, click on the Shared File tab.	

Step	Action	See also
2	If you are changing an existing configuration to import from a different XML file, click on the <b>Clear</b> button to clear the dialog fields.	
3	Click on the <b>Select File</b> button and browse for the filename to import data from. This would be an XML file produced by the Enterprise Architect Data Exporter.	
4	If you have entered the name of a valid file, a list of tables to import displays in the Datasets in File panel.  Click on one or more of the tables to import. Press <b>(Ctrl)</b> or <b>(Shift)</b> to click on multiple tables.	<a href="#">Export Reference Data</a> <sup>[376]</sup>
5	If you prefer to control whether or not the automatic import takes place, select the <b>Always prompt before import</b> checkbox.	
6	Click on the <b>Import</b> button to import the reference data now, and to enable the automatic check and import for subsequent reloads.	

### Notes

- Concerning General Types data:
  - For Statuses, Constraints, Requirements and Scenarios, the imported data is merged with the existing data; if a record already exists it is updated to the new values, and if the record does not exist, a new record is added - records are never deleted
  - For Constraint Status Types, Difficulty, Test Status and Priority, the complete existing list is replaced by the imported list
- Concerning Calendar events and event types, Team Review configurations, Patterns and Gap Matrix Profiles:
  - The imported data is merged with the existing data; if a record already exists it is updated to the new values, and if the record does not exist, a new record is added - records are never deleted

## 3.7 Change Management



As your users develop their models in the project, you need to monitor and control the changes they make to their own data and to data that they might share. Enterprise Architect provides a number of tools and facilities for managing and protecting the data in a project.

### Topics

Topic	Detail	See also
<b>Version Control of Packages</b>	<p>Enterprise Architect Model Version Control enables you to:</p> <ul style="list-style-type: none"> <li>• Coordinate sharing of packages between users, with either read-only access or update access, ensuring that work on different areas of the model is coordinated and synchronous rather than conflicting</li> <li>• Save and retrieve a history of changes to packages</li> </ul> <p>To use version control in Enterprise Architect, you require a third-party source-code control application such as:</p> <ul style="list-style-type: none"> <li>• Subversion</li> <li>• CVS</li> <li>• MS Team Foundation Server (TFS), or</li> <li>• Any other version control product that complies with the Microsoft Common Source Code Control standard</li> </ul>	<a href="#">Version Control</a> [383]
<b>Tracking Changes</b>	<p>Enterprise Architect provides two separate but complementary facilities for tracking changes to data across the project:</p> <ul style="list-style-type: none"> <li>• Auditing of model changes</li> <li>• Baselining and differencing to capture and roll back changes</li> </ul>	<a href="#">Tracking Changes</a> [445]
<b>Project Data Transfer</b>	<p>Enterprise Architect enables you to transfer project data between project data repositories either for:</p> <ul style="list-style-type: none"> <li>• Sections of the project (XML and CSV) or</li> <li>• The whole project, row by row, table by table (in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect)</li> </ul>	<a href="#">Model Transfer</a> [473]

[Learn more](#)

- [Check Project Data Integrity](#)<sup>[597]</sup>
- [Maintenance](#)<sup>[596]</sup>
- [Team Development](#)<sup>[307]</sup>
- [Spell Checking](#)<sup>[549]</sup>
- [Reference Data](#)<sup>[1146]</sup>
- [Upgrade a Project](#)<sup>[599]</sup>

### 3.7.1 Version Control

Within Enterprise Architect, you can manage changes to and revisions of your projects by placing **individual** model packages, view nodes or root nodes under **version control**. Version control provides numerous key facilities, including:

- Saving a **history of changes** to packages
- The ability to retrieve **previous revisions** of packages
- Propagating model updates **between team members**
- Coordinating the **sharing of packages** between team members

You apply version control through a **third-party source-code control application** that manages access to and stores revisions of the controlled packages. Once the version control software has been installed and configured, you must define a **Version Control Configuration** within your project. You can then use version control to manage changes to the packages of your model.

#### Notes

- Sparx Systems strongly urge you not to manipulate version controlled package files outside of Enterprise Architect; it is possible to leave the package files in a state that Enterprise Architect cannot recognize
- Database replication should not be combined with version controlled packages
- If the packages under version control contain any alternative images and those images are subject to frequent change, you can set the **Export alternate images** option on the Options dialog to export the images to the version control repository when you check in the packages; if the images are not subject to frequent change, do not select this option and instead use **Export / Import Reference Data** to manage alternative images

#### Learn more

- [Introduction](#)<sup>[383]</sup>
- [Version Control Product Set-up](#)<sup>[394]</sup>
- [Version Control Setup](#)<sup>[413]</sup>
- [Use Version Control](#)<sup>[422]</sup>

#### 3.7.1.1 Introduction

Enterprise Architect's version control integration provides several key facilities, including:

- Saving a **history of changes** made to your model's packages
- Retrieving **previous revisions** of packages
- **Propagating model updates** between team members
- Coordinating the **sharing of packages** between team members

There are a number of factors to consider when setting up and using version control in your model development.

#### Factors to consider

Factor	Detail	See also
<b>System Requirements and Configuration</b>	<p>You apply version control through a <b>third-party source-code control application</b> that manages access to and stores revisions of the controlled packages.</p> <p>Typically the configuration consists of:</p> <ul style="list-style-type: none"> <li>• A server component that manages a version control repository, and</li> <li>• Client components on the workstations, that manage local working copies of controlled files</li> </ul> <p>Enterprise Architect uses the client component to communicate with the server. A version control client must be installed on every machine where you run Enterprise Architect and want to access your version control system.</p>	<a href="#">System Requirements</a> <sup>[395]</sup>
<b>Version Control Usage</b>	<p>There are two main ways in which projects can be deployed:</p> <ul style="list-style-type: none"> <li>• <b>Centralized Shared Model</b></li> <li>• <b>Distributed Private Models</b></li> </ul> <p>Version control is employed in the same way for both scenarios; however, when using <b>Private Model</b> deployment you have the additional facility of propagating model updates throughout the team.</p> <p>Version Control can also be used to share standard packages between different projects.</p>	<a href="#">Version Control Usage</a> <sup>[385]</sup>
<b>Team Deployment</b>	<p>Consider the process of setting up a version control environment and applying version control to a project to be accessed by <b>a number of users</b>.</p>	<a href="#">Applying Version Control in a Team Environment</a> <sup>[389]</sup>
<b>Version Control Basics</b>	<p>Enterprise Architect enforces <b>serialized editing</b> of version controlled packages, using the <b>lock-modify-unlock</b> mode of operation.</p>	<a href="#">Version Control Basics</a> <sup>[387]</sup>
<b>Applying Version Control to Models</b>	<p>Using version control consists of placing <b>individual model packages</b> under version control, rather than version controlling the project as a whole.</p>	<a href="#">Version Control Of Model Data</a> <sup>[386]</sup> <a href="#">Version Control Nested Packages</a> <sup>[390]</sup>



Factor	Detail	See also
<b>Version Control and Project Reference Data</b>	To share changes in <b>reference data</b> between users in a version-controlled project deployed as multiple <b>private models</b> , you periodically <b>export</b> the reference data from the model where the changes were made, and <b>import</b> it into the other models maintained by the team.	<a href="#">Version Control and Reference Data</a> <sup>[387]</sup>
<b>Offline Version Control</b>	You can prevent the system from attempting to make any version control connections by choosing to <b>Work Offline</b> before loading a model.  If Enterprise Architect is unable to connect a Version Control Configuration for any reason, it displays warning messages to notify you and provides 'offline' version control functionality for all packages associated with the failed connection.	<a href="#">Offline Version Control</a> <sup>[392]</sup>

#### Notes

- Packages under version control are identified in the Project Browser by **icons** that indicate the current **status** of the package

#### Learn more

- [Project Browser Indicators](#) <sup>[391]</sup>
- [Version Control Branching](#) <sup>[394]</sup>

#### 3.7.1.1.1 Version Control Usage

The version control facility can be used in many different ways, although these roughly fall into one of four types of use as discussed here.

#### Usage

Type	Usage	See also
<b>Single Shared model</b>	Users <b>share</b> a model stored in a <b>central</b> project file or DBMS repository. In this configuration you can view changes to other users' Packages without explicitly having to check them out, by simply refreshing your view of the model.  Version control is used to: <ul style="list-style-type: none"> <li>• Archive successive versions of your work to date</li> <li>• Maintain Package revision history</li> <li>• Recover from unwanted changes or accidental deletions, through an 'undo' facility</li> <li>• Regulate access to Packages</li> </ul>	
<b>Multiple Private models</b>	<b>One</b> model is created by a single user who configures it for version control. The model file is then <b>distributed</b> to other users, with each user	

Type	Usage	See also
	<p>storing <b>their own private copy</b> of the model.</p> <p>Version control is used to:</p> <ul style="list-style-type: none"> <li>• Propagate changes to the model, throughout the team</li> <li>• Archive successive versions of your work to date</li> <li>• Maintain Package revision history</li> <li>• Recover from unwanted changes or accidental deletions, through an 'undo' facility</li> <li>• Regulate access to Packages</li> </ul>	
<b>Shared Packages</b>	<p>Individual users create <b>separate</b> models but <b>share</b> one or more <b>Packages</b>:</p> <ul style="list-style-type: none"> <li>• Users share Packages through version control</li> </ul>	
<b>Standard Packages</b>	<p>A company might have a <b>standard set</b> of read-only Packages that are broadly <b>shared</b>:</p> <ul style="list-style-type: none"> <li>• Individual users retrieve Packages with the <b>Get Package</b> menu option</li> </ul>	

#### Learn more

- [Version Control Best Practices for Enterprise Architect](#) (Online Resource)

#### **3.7.1.1.2 Version Control of Model Data**

When applying version control in Enterprise Architect, you place individual model packages under version control, and not the project as a whole.

All Enterprise Architect models are stored in databases - even the .eap file is an MS Jet database. In simple terms, the project file is a **single entity** of **binary data**. Being binary data, the project file would require the use of the **lock-modify-unlock** model of version control, which would mean that only a **single user** at a time could work on any given (version controlled) model. Therefore, it is not practical to apply version control to the database (.eap file) as a whole; this can also create problems for you:

- Most version control systems mark their controlled files as **read only**, unless they are specifically checked-out to you
- The .eap file is an MS Jet database, and Enterprise Architect must be able to open this file for **read/write** access when you load your model; the system displays an error message and fails to load the model if it is read-only

#### **Version Controlling Packages in your Model**

To overcome the limitations described above, Enterprise Architect exports discrete **units** of the model - the **packages** - as **XMI package files**, and it is **these XMI files**, not the project file, that are placed under version control. The XMI file format used by Enterprise Architect dictates that they too be treated as binary files - therefore it is not possible to merge the XMI files either; however, by splitting the model into much smaller parts, many users can work on separate parts of the model simultaneously.

### Version Controlled Nested Packages

Version controlled **nested** packages result in much smaller XML files being exported, as the **parent packages'** XML files do not contain any content for the version controlled **child** packages.

Version control of nested packages, together with a model structure of small individual packages, provides greater scope for multiple users to work concurrently, as individual users are locking much smaller parts of the model.

#### Learn more

- [Version Control Basics](#)<sup>[387]</sup>

#### **3.7.1.1.3 Version Control and Reference Data**

Reference data is data that is used **across** a model or project; it is **not package-specific**. Version control operates at **package level**, and therefore does not capture changes in reference data. Where version control is used in a multiple private model set up, changes in reference data are not brought into the model when packages are updated from version control.

In a Shared Model environment, all users are accessing **the same** project reference data. Changes in reference data can be shared between users in a version-controlled project deployed as **multiple private models**, by periodically **exporting** the reference data from the model where the changes were made, and **importing** it into the other models maintained by the team.

Reference data is exported and imported as an **XML** file, which contains whatever types of reference data you want to transfer. You **can** place your project reference data under version control by exporting the data as an XML file and apply version control **to that file** using your version control software **external** to Enterprise Architect.

#### Learn more

- [Reference Data](#)<sup>[1146]</sup>
- [Sharing Reference Data](#)<sup>[374]</sup>
- [Export Reference Data](#)<sup>[376]</sup>
- [Import Reference Data](#)<sup>[380]</sup>

#### **3.7.1.1.4 Version Control Basics**

Enterprise Architect implements version control of your model by **exporting package data** from the project database to **XML package files**, which are placed under version control in the **source-code control application**. The XML file format cannot be **merged** in the same way as ordinary text files can be merged, which is why Enterprise Architect must enforce **serialized editing** of version controlled packages, as discussed below.

### The Lock-Modify-Unlock Solution

Many version control systems use a **lock-modify-unlock** model to address the problem of different authors in a shared source overwriting each other's work. In this model, the version control repository allows only one person to change a file at a time, and access is managed using locks.

Harry must lock a file before he can begin making changes to it. If Harry has locked a file, Sally cannot also lock it, and therefore cannot make any changes to that file. All she can do is read the file, and wait for Harry to finish his changes and release the lock. After Harry unlocks the file, Sally can take her turn in locking and editing the file.

### The Copy-Modify-Merge Solution

Subversion, CVS and a number of other version control systems use a **copy-modify-merge** model as an alternative to locking. In this model, each user's client contacts the project repository and creates a **personal working copy** - a local reflection of the repository's files and directories. Users then work **simultaneously and independently**, modifying their private copies. In due course, the private copies are **merged** together into a new, final version. The version control system often assists with the merging, but ultimately a person is responsible for making it happen correctly.

### When Locking is Necessary

While the lock-modify-unlock model is generally considered a hindrance to collaboration, there are still times when locking is necessary.

The copy-modify-merge model is based on the assumption that files are contextually merge-able: that is, that the files in the repository are line-based text files (such as program source code). However, for files with **binary** formats, such as artwork or sound, it is often impossible to merge conflicting changes. In these situations, it really is necessary for users to take strict turns in changing the file. Without serialized access, some users end up wasting time on changes that are ultimately overwritten.

### Learn more

- [Add Connectors To Locked Elements](#) <sup>388</sup>

#### **3.7.1.1.4.1 Add Connectors To Locked Elements**

Generally, when working in a diagram containing locked elements, you **cannot** add a connector to a locked element. However, this depends on the lock status of the source and target elements (or more precisely, the lock status of the parent packages of the source and target elements, when the source and target element are held in different packages). There are scenarios in which a connector **can** be created on a locked element.

### Lock Scenarios

Element Status	Add Connectors	See also
Source unlocked, target unlocked	Yes, any kind of connector can be added	
Source unlocked, target locked	Yes, except for Composition connectors	
Source locked, target unlocked	No, except for Composition connectors	
Source locked, target locked	No, prohibited for all connectors	

### Notes

- A connector can be added if its source is unlocked - you are modifying what the source can see
- The exception is Composition connectors, where the target (the parent) must be unlocked - you are

modifying the parent by adding children

#### Learn more

- [Version Control Basics](#)<sup>[387]</sup>

#### **3.7.1.1.5 Applying Version Control in a Team Environment**

The process of **setting up** a version control environment and **applying** version control to a project to be accessed by a **number of users** is summarized below.

#### Version Control - Process Overview

Step	Action	See also
1	Install your version control product.	<a href="#">Version Control Product Setup</a> <sup>[394]</sup>
2	Create a version control repository.	
3	Create a version control project to be used with your Enterprise Architect project.	
4	<p>Check-out a working copy of the version control project (a module, project or folder within the version control system) into a local folder.</p> <p>You must do this for every team member that is accessing the version controlled packages, whether you are using a single shared model or each team member stores his own private copy of the model.</p>	<a href="#">Create a Local Working Copy</a> <sup>[399]</sup> (Subversion)  <a href="#">Prepare a CVS Local Workspace</a> <sup>[405]</sup>
5	<p>Within Enterprise Architect, define a version control configuration to provide access to the working copy files.</p> <p>The name of the version control configuration must be the same across all machines throughout a team. That is, all version control access to a given package must be through version control configurations with the same name, across all models and all users.</p> <p>The easiest way to perform this step, (throughout the team), is to have one user set up version control on the model and then share that model with the rest of the team.</p> <ul style="list-style-type: none"> <li>• In Shared Model deployment, all users connect to a single instance of the model database, so the model is shared automatically</li> <li>• In Private Model deployment, it is easiest to distribute copies of the original model (after version control has been set up) to all other members of the team</li> </ul> <p>Whenever you open a model (Private or Shared) that uses a version control configuration that is not yet defined on your workstation, a prompt displays to complete the definition for that configuration. This typically means specifying the local working copy directory and maybe choosing the version control project associated with this Enterprise Architect project.</p> <p>Once this has been done, the version controlled packages that already exist in</p>	<a href="#">Version Control Setup</a> <sup>[413]</sup>

	the model are ready for use.	
6	Configure packages within the Enterprise Architect model for version control. That is, apply version control to individual packages.	<a href="#">Configure Controlled Package</a> <sup>[424]</sup>
7	Check-out and check-in packages as required.	<a href="#">Check Out a Package</a> <sup>[431]</sup> <a href="#">Check In a Package</a> <sup>[432]</sup>

### Notes

- It is possible to use multiple version control configurations within the same model; different packages can still use different version control configurations within the model, as long as any given package is always accessed via the same version control configuration

### Learn more

Team deployment and the use of version control is discussed in two Sparx Systems white papers, available as on-line resources on the Sparx Systems web site:

- [http://www.sparxsystems.com/WhitePapers/Version\\_Control.pdf](http://www.sparxsystems.com/WhitePapers/Version_Control.pdf)
- [http://www.sparxsystems.com/downloads/whitepapers/EA\\_Deployment.pdf](http://www.sparxsystems.com/downloads/whitepapers/EA_Deployment.pdf)

#### 3.7.1.1.6 Version Control Nested Packages

When you save a package to the version control system, only stub information is exported for any nested packages. This protects information in a nested package from being inadvertently over-written by a top level package.

### Operations on Nested Packages

Operation	Detail	See also
<b>Checking Out</b>	When you check out a package, you do not modify or delete nested packages; only the top level package is modified.  As a consequence of this behavior, if you check out or get a version controlled package with nested packages that are not already in your model, you see stubs in the model for the nested packages only.	
<b>Get All Latest</b>	If you select the <b>Get All Latest</b> option from the version control menu, any new stubs are populated from the version control system.	<a href="#">Get All Latest</a> <sup>[427]</sup>
<b>Importing Models</b>	You can populate a large and complex model, by 'getting' only the root packages, then using <b>Get All Latest</b> to recursively iterate	<a href="#">Get Package</a> <sup>[436]</sup>

Operation	Detail	See also
	<p>through the attached and nested packages.</p> <p>This is a powerful and efficient means of managing your project and simplifies handling very large models, even in a distributed environment.</p> <p>The <b>Import a Model Branch</b> option combines the steps described above into a single operation.</p>	<a href="#">Import Controlled Model Branch</a> <sup>[438]</sup>





### Notes

- It is recommended that, when sharing a version controlled model, you do not mix versions of Enterprise Architect later than version 4.5 with earlier versions; if this is necessary it is best to go to the Version Control Settings dialog and deselect the **Save nested version controlled packages to stubs only** checkbox, setting Enterprise Architect to the pre-version 4.5 behavior (for the current model only)

#### 3.7.1.1.7 Project Browser Indicators

Packages under version control are identified in the Project Browser by icons that indicate the current status of the package.

### Indicators

Icon	Indicates that	See also
	<p>This package is version controlled and not checked out to you.</p> <p>You cannot edit the package (until you check out the package yourself).</p>	<a href="#">Check Out a Package</a> <sup>[437]</sup>
	<p>This package is version controlled and checked out to <b>you</b>.</p> <p>You can edit the package.</p>	
	<p>This package is version controlled, but you checked it out whilst not connected to the version control server.</p> <p>You can edit the package but there could be version conflicts when you check the package in again.</p>	<a href="#">Offline Version Control</a> <sup>[392]</sup>
	<p>This package is controlled and is represented by an XML file on disk, but it is not under version control.</p> <p>You can edit this package.</p>	<a href="#">Controlled Packages</a> <sup>[484]</sup>

### Learn more

- [Using Version Control](#)<sup>[422]</sup>

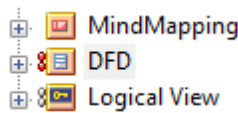
### 3.7.1.1.8 Offline Version Control

When loading a model that uses version control, Enterprise Architect normally initializes a connection to the version control system for each version control configuration defined in the model. If Enterprise Architect is **unable** to connect a version control configuration for any reason, it displays warning messages to notify you and provides **offline** version control functionality for all packages associated with the failed connection.



You can prevent Enterprise Architect from starting to make any version control connections, by selecting to work offline menu **before** loading your model.

Access    **Project | Version Control | Work Offline**

#### Working Offline

Concept	Discussion	See also
<b>Choosing to Work Offline</b>	<p>Selecting to work offline is useful if you know beforehand that Enterprise Architect cannot connect to your version control system. For example: If you are working on a laptop computer that is <b>disconnected</b> from your network, on an Enterprise Architect model that uses a large number of version control configurations, you can select to work offline before you load the model to avoid all the error messages that the system would normally display as each version control connection attempt failed.</p> <p>You can switch between working offline and working online at any time, either before or after a model is loaded, by toggling the <b>Work Offline</b> menu option. Enterprise Architect disconnects or reconnects version control (depending on the connection availability) according to your selection.</p>	
<b>Using Version Control Whilst Disconnected From the Version Control Server</b>	<p>Enterprise Architect 'remembers' the status of a model's version controlled packages. Packages that were checked out to you prior to disconnecting from the server are still shown as checked out to you, even though you are no longer connected to the server. You can still edit these packages as you normally would.</p> <p>Packages that were not checked out to you prior to disconnecting from the server are shown as version controlled and locked. You cannot edit these packages until you check them out.</p>	
<b>Offline Check Out</b>	<p>You can 'check-out' and edit a version controlled package even when your machine is disconnected from the version control server. In the example below, the <b>colored</b> 'figure 8' icon for <i>DFD</i> indicates that you have checked it out whilst <b>offline</b>.</p> <div style="text-align: center;">  <p>MindMapping DFD Logical View</p> </div> <p>(The <b>gray</b> 'figure 8' icon shown against <i>Logical View</i> indicates that you have checked out a version-controlled package <b>online</b>.)</p>	<a href="#">Project Browser Indicators</a> <sup>391</sup>



Concept	Discussion	See also
	<p>You should be aware that the version control system, and therefore other users, have no way of knowing that you have 'checked-out' a package whilst offline. It is not possible to <b>merge</b> changes to an XML file that result from two users editing the same package at the same time. If an offline checkout leads to two people editing the same package at the same time, when the changes are brought back online the <b>first-saved</b> set of changes is lost.</p>	
<b>Checking In a Package That Was Checked Out Offline</b>	<p>Once you reconnect your system to the version control server, if the package you checked out offline is not currently checked out by another user, you can check in that package. However, before Enterprise Architect checks in the package, it compares the local working copy of the package file with the latest revision in the repository. (These package files remain unchanged in your work area until Enterprise Architect exports the package again before checking in.) If the repository version <b>remains unchanged</b> from when you last updated your local copy, Enterprise Architect exports and checks in your package without further prompting.</p> <p>On the other hand, if the repository now contains a file that <b>has changed</b> since you last updated your local copy, checking in your package would <b>overwrite</b> those changes. Enterprise Architect displays a message warning you of the pending data loss and giving you the opportunity to abort the check in. At this point, you must decide whether to discard your own changes, using the <b>Undo Check Out</b> command, or continue with your check in and overwrite the changes that have been committed to the repository since you last updated your local copy from the repository.</p> <p>You can use the <b>File Properties</b> command to determine who checked in the last changes to this package. This might help you to discover what changes have been uploaded and decide whose changes take precedence.</p>	<p><a href="#">Package Version Control Menu</a> </p>
<b>Update Before You Disconnect</b>	<p>Whenever you are connected to the version control server, you are always working with the latest version of a package. This is because you cannot modify a package until you check it out from version control, and checking it out loads the latest revision from the repository into your model.</p> <p>This cannot happen when you are disconnected from the version control server. You are working on whatever versions you have on your machine, dating back to the last time you updated your local copy of each version controlled package. So, if you are planning to work on a model whilst disconnected from version control, it is a very good idea to make sure that you have the latest versions of all packages <b>before you disconnect</b>. The <b>Get All Latest</b> option makes this a simple task.</p>	<p><a href="#">Get All Latest</a> </p>

### 3.7.1.1.9 Version Control Branching

Currently, Enterprise Architect **does not** support Version Control Branching.

Work-arounds to achieve similar results might be possible for certain version-control products; contact **Sparx Support** for advice.

User Type	Contact via
Registered users	<a href="http://www.sparxsystems.com/registered/reg_support.html">http://www.sparxsystems.com/registered/reg_support.html</a>
Trial users	<a href="mailto:support@sparxsystems.com">support@sparxsystems.com</a>

### 3.7.1.2 Version Control Product Setup

To control and maintain the different revisions of your project packages, Enterprise Architect uses third-party version control products. Once your version control product is installed and a suitable environment has been created, Enterprise Architect can use that environment to control your project's packages.

Typically, version control products consist of:

- A server component
- A client component

Enterprise Architect integrates with the version control client components for Subversion, CVS and MS Team Foundation Server command line clients, as well as for products having API clients that comply with the MS SCCI specification.

#### Version Control System Components

Component	Detail	See also
<b>Version Control Server</b>	<p>The server component maintains the controlled files in their many revisions, in a central repository.</p> <p>The server component is usually located on a server machine that is accessible to all team members who are using version control.</p>	
<b>Server Configuration</b>	<p>The steps for configuring a version control server are, broadly:</p> <ul style="list-style-type: none"> <li>• Install the software</li> <li>• Create a repository</li> <li>• Create version control projects (or modules or folders for use with specific projects)</li> <li>• Configure user IDs and passwords</li> </ul> <p>For details on configuring any particular version control server, consult the appropriate documentation provided with the version control product.</p>	

Component	Detail	See also
<b>Version Control Client</b>	<p>The client component manages the working copies of the controlled files, submitting files to or retrieving files from the server as required.</p> <p>A version control client must be installed on every machine on which you run Enterprise Architect and want to access your version control system.</p>	
<b>Client Configuration</b>	<p>The steps for configuring a version control client are, broadly:</p> <ul style="list-style-type: none"> <li>• Install the software</li> <li>• Create a new directory for use as a local working copy folder</li> <li>• Log in to the version control server</li> <li>• Associate the working copy folder with its corresponding server repository folder</li> </ul> <p>For details on setting up a product-specific version control environment for use with Enterprise Architect, click on the appropriate link in the next column.</p>	<p><a href="#">Create a Subversion Environment</a> 397</p> <p><a href="#">Create a CVS Environment</a> 404</p> <p><a href="#">Create a TFS Environment</a> 408</p> <p><a href="#">Create an SCC Environment</a> 411</p>

### 3.7.1.2.1 System Requirements

Enterprise Architect is a Windows-based application and requires a Windows-based version control **client** for integration. It is independent of the version control **server** component and the platform on which that runs.

#### Version Control Product Requirements

Product	Detail	See also
<b>Subversion</b>	<p>Subversion is free, open source software.</p> <p>Subversion server components are available to run on a wide range of different hardware and operating systems.</p> <p>Enterprise Architect is not affected by your choice of <b>server</b> components, but requires Subversion's Windows-based command line <b>client</b> for integration.</p> <p>There are many graphical user interface clients available for use with Subversion, such as TortoiseSVN; this type of client <b>cannot be used directly</b> for integration with Enterprise Architect, but can be very useful in <b>preparing</b> a working Subversion environment for use by Enterprise Architect.</p> <p>Binary packages are available for download from:</p> <ul style="list-style-type: none"> <li>• <a href="http://subversion.apache.org/packages.html">http://subversion.apache.org/packages.html</a></li> </ul> <p>Subversion documentation is available from:</p> <ul style="list-style-type: none"> <li>• <a href="http://svnbook.red-bean.com/nightly/en/index.html">http://svnbook.red-bean.com/nightly/en/index.html</a></li> </ul>	<p><a href="#">Create a Subversion Environment</a> 397</p>

Product	Detail	See also
<b>Concurrent Versions System (CVS)</b>	<p>CVS is free, open source software.</p> <p>CVS server components are available to run on a wide range of different hardware and operating systems.</p> <p>Enterprise Architect is not affected by your choice of <b>server</b> components, but requires CVS's Windows-based command line <b>client</b> for integration.</p> <p>There are many graphical user interface clients available for use with CVS, such as TortoiseCVS; this type of client <b>cannot be used directly</b> for integration with Enterprise Architect, but can be very useful in <b>preparing</b> a working CVS environment for use by Enterprise Architect.</p> <p>The software is available for download from:</p> <ul style="list-style-type: none"> <li>• <a href="http://www.nongnu.org/cvs/">http://www.nongnu.org/cvs/</a></li> </ul> <p>CVS documentation is available from:</p> <ul style="list-style-type: none"> <li>• <a href="http://cvsbook.red-bean.com/cvsbook.html">http://cvsbook.red-bean.com/cvsbook.html</a></li> </ul>	<p><a href="#">Create a CVS Environment</a><sup>[404]</sup></p>
<b>Microsoft Team Foundation Server</b>	<p>Enterprise Architect is able to use either the:</p> <ul style="list-style-type: none"> <li>• Command line client for TFS, or</li> <li>• MS TFS-SCC client</li> </ul> <p>Your choice of client affects how you specify your Version Control Configuration.</p> <p>MS TFS-SCC clients are available for download from Microsoft's web pages:</p> <ul style="list-style-type: none"> <li>• <a href="#">Visual Studio 2005 Team Foundation Server MSSCCI Provider</a></li> <li>• <a href="#">Visual Studio Team System 2008 Team Foundation Server MSSCCI Provider</a></li> </ul>	<p><a href="#">TFS Settings</a><sup>[421]</sup></p> <p><a href="#">SCC Settings</a><sup>[417]</sup></p>
<b>Common Source Code Control (SCC)-compatible products</b>	<p>Any version control product that provides a client that complies with the Microsoft Common Source Code Control standard, version 1.1 or higher, can be integrated with Enterprise Architect.</p> <p>The following products are SCC-compatible and are known to successfully integrate with Enterprise Architect:</p> <ul style="list-style-type: none"> <li>• Accurev</li> <li>• ClearCase</li> <li>• MS Visual Source Safe</li> <li>• MS TFS-SCC</li> <li>• MKS Source Integrity</li> <li>• Perforce</li> <li>• Source Offsite</li> <li>• Snapshot CM</li> </ul>	

Product	Detail	See also
	Products that do not appear in this list should still integrate successfully with Enterprise Architect, if there is a <b>client</b> available for that product that complies with the MS SCC API specification.	

### 3.7.1.2.2 Create a Subversion Environment

You can use **Subversion** as a version control provider for Enterprise Architect. The first step in doing this is for a Subversion administrator to install and configure the appropriate software. A number of basic tasks are performed in creating an **operational Subversion environment**, and useful **tools** are available for performing some of these tasks.

#### Tasks in Creating a Subversion Environment

Task	Detail	See also
<b>Install server components</b>	<p>Executable files for Subversion can be obtained from the Apache Software Foundation.</p> <p>Subversion server components are available to run on a wide range of different hardware and operating systems; Enterprise Architect is not affected by your choice of server components.</p> <p><b>VisualSVN</b> is a package that can greatly simplify the installation, configuration and management of your Subversion server.</p>	<a href="#">Apache Subversion</a>  <a href="#">Official Subversion Documentation</a>
<b>Create a repository</b>	Please consult the official Subversion documentation.	
<b>Create Subversion users</b>	Please consult the official Subversion documentation.	
<b>Create a new repository sub-tree</b>	<p>It is good practice to create a new <b>repository sub-tree</b> in Subversion for <b>each</b> new Enterprise Architect model being added to version control with Subversion. Users should create a new local working copy from the sub-tree to be used with that model</p> <p><b>TortoiseSVN</b> can greatly simplify the process of creating new repository sub-trees.</p>	<a href="#">Create a Repository Sub-tree</a> <sup>[398]</sup> <a href="#">TortoiseSVN</a> <sup>[403]</sup>
<b>Install client components</b>	Executable files for Subversion can be obtained from the Apache Software Foundation.	<a href="#">Apache Subversion</a>
<b>Create a working copy folder</b>	A working copy folder must exist on each users' machine, for Enterprise Architect to use when exporting and importing the version controlled package files. It is this folder that is specified as the Local Project Path, when defining your Version Control Configurations.	<a href="#">Create a Local Working Copy</a> <sup>[399]</sup>

Task	Detail	See also
	<p>The working copy folder is the 'sandbox' where you modify the controlled files. The working copy folder is usually associated with a folder that exists within the version control repository. In Subversion, to create a local working copy you perform an initial check-out of a folder from the Subversion repository; this downloads a copy of the folder and its contents, to create your local working copy.</p> <p>TortoiseSVN can greatly simplify the initial check out of a working copy folder.</p>	<a href="#">TortoiseSVN</a> <sup>[403]</sup>
<b>Setting up Subversion under Wine/CrossOver</b>	The process of setting up and using Subversion with Enterprise Architect under Wine is almost identical to the process when running natively under Windows, apart from minor differences in installing the Subversion client and performing the initial check out of the working copy folder.	<a href="#">Subversion under Wine</a> <sup>[401]</sup>

#### Notes

- Enterprise Architect relies on exclusive file locking when applying version control to its packages; file locking was not introduced into Subversion until version 1.2, therefore Enterprise Architect does not work with Subversion releases earlier than Subversion 1.2
- Enterprise Architect can only communicate with the Subversion server using the Subversion command line client **svn.exe**

#### Learn more

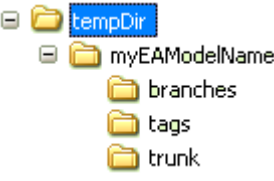
- [TortoiseSVN](#)<sup>[403]</sup>

#### 3.7.1.2.2.1 Create a new Repository Sub-tree

When you set up **Subversion** as your version control tool, it is good practice to create a new **repository sub-tree** in Subversion for **each** new Enterprise Architect model. The sub-tree can be used to control the package files for your model.

#### Create a new sub-tree in the Subversion Repository

Step	Action	See also
1	Use Windows Explorer to create a temporary directory on your PC file system, to be imported into the Subversion repository as a new repository sub-tree. The directory would have this structure:	

		
2	<p>Open a Windows command prompt, navigate to <i>tempDir</i> and issue the Subversion command <b>import</b>.</p> <p>For example;</p> <pre>C:\Documents and Settings\user&gt; cd \tempDir C:\tempDir&gt; svn import . https://host.example.com:8443/repos/ --message "Repository Initialization"</pre>	<a href="#">Subversion repository administration basics</a>
3	On your PC, delete the temporary directory structure ( <i>tempDir</i> ) and all its contents.	

#### Notes

- After the import is finished, the original tree is not converted into a Subversion working copy; you should delete the temporary structure and check out a fresh working copy of the tree
- The process described above can also be performed using TortoiseSVN's Repository Browser, which provides commands for simply creating new folders directly in the repository

#### 3.7.1.2.2.2 Create a Local Working Copy

In order to use Subversion to provide version control of the packages in a model, you need to prepare a functional **SVN working copy folder** that can be accessed through an Enterprise Architect **version control configuration** within that model.

#### Create a Subversion working copy folder

Step	Action	See also
1	Choose or create a suitable directory on your system in which to create your Subversion working copy.	
2	<p>Open a command prompt window and navigate to the directory you have selected. For example:</p> <pre>C:\&gt; mkdir mySVNWorkSpace C:\&gt; mkdir mySVNWorkSpace/myEAModelName C:\&gt; cd mySVNWorkSpace/myEAModelName</pre>	
3	<p>Perform the initial check out from the Subversion repository, specifying the repository folder and local working copy folder, as well as your user name and password. For example:</p> <pre>C:\&gt; svn checkout "https://myserver:8443/svn/repos_folder"</pre>	

	<pre>"C:\mySVNWorkSpace\myEAModelName" --username myUserName --password myPassword</pre> <p>(By specifying your Subversion username and password, you ensure that they are correctly cached by Subversion and available for use by Enterprise Architect.)</p> <p>If you specify the HTTPS protocol when performing the initial Subversion check out, a prompt displays to accept a security certificate; in this instance, press <b>( P )</b> to permanently accept the certificate.</p> <p>The nominated local folder is configured as a Subversion working copy folder.</p> <p>Any files already existing in the repository folder are downloaded to the working copy folder as working copy files.</p>	
4	Verify that your Subversion environment functions correctly.	<a href="#">Verify the SVN Workspace</a> 400

#### Notes

- It is important that Subversion caches your username and password, so that it never has to prompt for user input; a check-out operation might not request authentication, and if it does not you should perform an action that does request authentication, such as adding and committing a dummy test file
- The process described above can also be performed using TortoiseSVN's **Checkout** command (which provides options to browse when specifying both your repository folder and your local folder); when prompted for authentication details by TortoiseSVN, make sure you place a check against the **Save Authentication Data** option

#### 3.7.1.2.2.3 Verify the SVN Workspace

After creating the Subversion local working copy to hold the working copies of your package files, you can verify that it functions correctly before attempting to use it with Enterprise Architect. You need to be able to add files to Subversion, lock the files and commit changes to those files.

#### Verify correct operation of a Subversion working copy folder

Step	Action	See also
1	Use Windows to open a command prompt window.	
2	Select the directory you specified as the working copy, in the Subversion checkout command when preparing the SVN workspace. For example: <pre>C:\&gt; cd mySVNWorkSpace</pre>	<a href="#">Create a Local Working Copy</a> 399
3	Create a test file, such as <i>Test.txt</i> , containing the text <b>Subversion Test</b> . For example:	



	<pre>C:\&gt; echo Subversion Test &gt; Test.txt</pre>	
4	<p>Execute each of the following Subversion commands:</p> <ul style="list-style-type: none"> <li>• <code>svn add Test.txt</code></li> <li>• <code>svn commit -m"Commit comment" Test.txt</code></li> <li>• <code>svn update Test.txt</code></li> <li>• <code>svn lock Test.txt</code></li> <li>• Use your preferred editor to modify the file and save the changes</li> <li>• <code>svn commit -m"Second commit comment" Test.txt</code></li> </ul> <p>The commands should execute without any errors and without prompting the user for any extra input.</p>	

**Notes**

- Your environment must be set up such that you can perform these operations without ever being prompted for user ID or password; for further information, please see the *Caching Client Credentials* topic in the official Subversion documentation

**Learn more**

- [Official Subversion Documentation](#)

**3.7.1.2.2.4 Subversion Under Wine-Crossover**

If you want to set up and use Subversion with Enterprise Architect under **Wine**, the process is almost identical to the process for setting up and using the systems under Windows. When running Enterprise Architect under Wine or CrossOver, you still use a Windows-based Subversion command line client.

There are some minor differences in the way you prepare the Subversion working environment, specifically in the way you install your Subversion client and the way you check out a working copy folder from the Subversion repository.

**System Requirements**

Sparx Systems has tested the use of Enterprise Architect with Subversion under Wine 1.2, on **Mac OS 10.4** and **10.6.2**, and on **Ubuntu 10.04**. All tests were passed.

When using Wine 1.2 on the **Ubuntu 9.10** platform, Sparx Systems was able to use the **svn:** and **file:** protocols to communicate with the SVN server; but not the **https:** protocol.

**Installing a Subversion Client**

Wine is able to install applications from either a Windows **.exe** file, or a **.msi** installer file.


Place the installer for your Windows Subversion client in a convenient location on the native file system, then open a **Wine console window** from within Enterprise Architect and run the installer from within the Wine console. Your Subversion installation can then access the same C: drive and folders that Enterprise Architect is accessing.

Learn more

- [Preparing a Subversion Environment Under Wine.](#)<sup>[402]</sup>

Under Wine, you can install Subversion from either a Windows **.exe** file, or a **.msi** file. By performing your Subversion installation and initial check out from a Wine console window opened from within Enterprise Architect, you have access to the same C: drive and folders that Enterprise Architect is accessing.

Set up Subversion for use with Enterprise Architect, running under Wine

Step	Action	See also
1	Start Enterprise Architect. You do not have to load a project at this point.	
2	Select <b>Tools   Customize &gt; Tools :</b>  (New). A new, blank entry is opened on the Tools tab of the Customize dialog.	<a href="#">Custom Tools</a> <sup>[157]</sup>
3	Define the new menu item entry, as follows: <ul style="list-style-type: none"> <li>• In the newly-opened <b>Menu contents</b> field, type the name <b>Wine Console</b></li> <li>• In the <b>Command</b> field, type <b>wineconsole</b></li> <li>• In the <b>Arguments</b> field, type <b>cmd</b></li> <li>• Leave the <b>Initial directory</b> field blank</li> </ul>	
4	Click on the <b>Close</b> button. The Customize dialog closes.	
5	Select <b>Tools   Wine Console</b> . A Wine console window opens.	
6	Type <b>C:</b> and press ( <b>Enter</b> ). The Wine console switches to the C: drive.	
7	Issue the command to install your Subversion client. For example:  <pre>C:\&gt;/Installers/Subversion-client-1.6.12-1.win32.exe</pre> To install from a .msi file, use Wine's <b>msiexec</b> utility. For example:  <pre>C:\&gt;msiexec "Slik-Subversion-1.6.9-win32.msi" /i</pre>	

	Installation of the Subversion command line client begins.	
8	<p>Create a folder to serve as the working copy folder to be used by Enterprise Architect. For example:</p> <pre>C:\&gt;mkdir C:\VC_workspaces\SVN_workcopy</pre>	
9	<p>Issue the command to perform the initial checkout from the Subversion repository, specifying the repository folder, working copy folder, username and password. For example:</p> <pre>C:\&gt;svn checkout "https://myServer:8443/svn/repos_folder" "C:\VC_workspaces\SVN_workcopy " "--username myUserName" "--password myPassword"</pre> <p>(After specifying your Subversion username and password, they are correctly cached by Subversion and are available for use by Enterprise Architect.)</p> <p>If the HTTPS protocol is specified when performing the initial Subversion check out, you are prompted to accept a security certificate; in this instance, press ( <b>P</b> ) to permanently accept the certificate.</p> <p>The nominated local folder is configured as a Subversion working copy folder. Any files already existing in the repository folder are downloaded to the working copy folder as working copy files.</p>	
10	<p>Type <b>Exit</b> and press ( <b>Enter</b> ).</p> <p>The Wine console window closes.</p> <p>You are now ready to load a project in Enterprise Architect and apply version control to it, following the normal Windows-based procedures.</p>	<a href="#">Version Control Setup</a> 413 <a href="#">Use Version Control</a> 422

**Notes**

- You should copy the installer for your Windows Subversion client to a convenient location on the native file system, so that you can easily refer to it from within the Wine console window in step 7 above

**3.7.1.2.2.5 TortoiseSVN**

**TortoiseSVN** is a Windows shell extension for Subversion; it provides icon overlays in Windows Explorer that are useful as a tool for observing the status of your Subversion controlled files. You can also use it to create your repository sub-trees and check out local working copies from within Windows Explorer, using simple menu commands.

You can download TortoiseSVN from <http://tortoisesvn.net/downloads.html>.

**Notes**

- Enterprise Architect can only communicate with the Subversion server using the Subversion command line client **svn.exe**

### 3.7.1.2.3 Create a CVS Environment

You can use **Concurrent Versions System (CVS)** as a version control provider for Enterprise Architect. The first step in doing this is for a CVS administrator to install and configure the appropriate software. A number of basic tasks are performed in creating an **operational CVS environment**, and useful **tools** are available for performing some of these tasks.

#### Tasks in Creating a CVS Environment

Task	Detail	See also
<b>Install server components</b>	<p>Executable files for CVS can be obtained from the Free Software Foundation, or from March Hare Software.</p> <p>CVS server components are available to run on a wide range of different hardware and operating systems; Enterprise Architect is not affected by your choice of server components.</p>	<a href="#">FSF CVS</a> <a href="#">March Hare CVS</a>  <a href="#">CVS Documentation</a>
<b>Create a repository</b>	Please consult the official CVS documentation.	<a href="#">CVS Repository Administration</a>
<b>Create CVS users</b>	Please consult the official CVS documentation.	
<b>Create a new repository module</b>	<p>It is recommended good practice to create a new <b>repository module</b> in CVS for <b>each</b> new Enterprise Architect model being added to version control with CVS. Users should create a new local working copy folder from the module to be used with that model.</p> <p>A repository module represents a project, or a set of related files in the repository.</p> <p><b>TortoiseCVS</b> can greatly simplify the process of creating new repository sub-trees.</p>	<a href="#">CVS Starting a New Project</a> (Online Resource)  <a href="#">TortoiseCVS</a> <sup>[407]</sup>
<b>Install client components</b>	<p>Executable files for CVS can be obtained from the Free Software Foundation, or from March Hare Software.</p> <p>Enterprise Architect is a Windows based application - it requires a Windows based CVS command line client for integration.</p>	<a href="#">FSF CVS</a> <a href="#">March Hare CVS</a>
<b>Create a working copy folder</b>	<p>A working copy folder must exist on each users' machine, for Enterprise Architect to use when exporting and importing the version controlled package files. It is this folder that is specified as the Local Project Path, when defining your Version Control Configurations.</p> <p>The working copy folder is the 'sandbox' where you modify the controlled files. The working copy folder is usually associated with a folder that exists within the version control repository. In CVS, to create a local working copy you perform an initial check-out of a folder from the CVS repository; this downloads a copy of the</p>	<a href="#">Prepare a CVS Local Workspace</a> <sup>[405]</sup>

Task	Detail	See also
	<p>folder and its contents, to create your local working copy.</p> <p>TortoiseCVS can greatly simplify the initial check out of a working copy folder.</p>	<a href="#">TortoiseCVS</a> <sup>[407]</sup>
<b>Setting up CVS under Wine/ CrossOver</b>	The process of setting up and using CVS with Enterprise Architect under Wine is almost identical to the process when running natively under Windows, apart from minor differences in installing the CVS client and performing the initial checkout of the working copy folder.	

**Notes**

- If you do not already use CVS for version control, you should consider using Subversion instead; Subversion's client-server protocols provide a broader range of possibilities for connecting to remote servers, with easier set up of secure connections
- TortoiseCVS is a Windows shell extension; Enterprise Architect cannot use TortoiseCVS as its client, it must use the CVS command line client

**Learn more**

- [Create a Subversion Environment](#)<sup>[397]</sup>

**3.7.1.2.3.1 Prepare a CVS Local Workspace**

In order to use CVS to provide version control of the packages in a model, you need to prepare a functional **CVS working copy folder** that can be accessed through an Enterprise Architect **version control configuration** within that model.

**Prepare a CVS Working Copy Folder**

Step	Action	See also
<b>1</b>	<p>Ask your System Administrator to install CVS and create a remote repository, with a module that you can use to control your Enterprise Architect package files.</p> <p>Your administrator must create a username and password for you before you can make a connection.</p>	
<b>2</b>	Select or create a suitable directory to use as your CVS working copy directory.	
<b>3</b>	<p>Open a command prompt window and navigate to your CVS working copy directory. For example:</p> <pre>C:\&gt; mkdir myCVSWorkSpace C:\&gt; cd myCVSWorkSpace</pre>	

4	<p>Log in to the remote CVS repository. For example:</p> <pre>C:\myCVSWorkspace&gt; cvs -d:pserver:myUserID@ServerName:/reposPath</pre> <p>Replace <i>myUserID</i> with your CVS username, replace <i>ServerName</i> with the name of your CVS server and replace <i>reposPath</i> with the path to the repository on the server.</p> <p>A prompt for a password displays.</p>	
5	<p>Enter your password.</p> <p>You are logged in to the CVS server.</p>	
6	<p>Perform the initial checkout of the CVS repository module, into the local working copy directory. For example:</p> <pre>C:\myCVSWorkspace&gt; cvs -d:pserver:myUserID@ServerName:/reposPath</pre> <p>(Replace <i>moduleName</i> with the name of the repository module that you want to check out.)</p> <p>A subdirectory is created in your current working directory, with the same name as the module being checked out.</p> <p>Any files already existing in the repository module are downloaded to the working copy folder as working copy files.</p>	
7	<p>Verify that your CVS environment functions correctly.</p>	<a href="#">Verify the CVS Workspace</a> 406

### Notes

- Much of the process described above can also be performed (more simply) using the TortoiseCVS command **Make New Module**

#### 3.7.1.2.3.2 Verify the CVS Workspace

After creating the CVS local working copy to hold the working copies of your package files, you can verify that it functions correctly before attempting to use it with Enterprise Architect. You need to be able to add files to CVS, and commit changes to those files. You also need to be able to register as an editor of the file and retrieve the list of currently registered editors.

#### Verify correct operation of a CVS working copy folder

Step	Action	See also
1	Use Windows to open a command prompt window.	

2	<p>Select the directory you specified as the working copy in the <b>cv</b>s <b>checkout</b> command, when preparing the CVS workspace. For example:</p> <pre>C:\&gt; cd myCVSWorkSpace</pre>	
3	<p>Create a test file, such as <i>Test.txt</i>, containing the text <b>CVS Test</b>. For example:</p> <pre>C:\&gt; echo CVS Test &gt; Test.txt</pre>	
4	<p>Execute the following CVS commands:</p> <ul style="list-style-type: none"> <li>• <code>cv</code>s <code>add</code> <code>Test.txt</code></li> <li>• <code>cv</code>s <code>commit</code> <code>-m</code>"Commit comment" <code>Test.txt</code></li> <li>• <code>cv</code>s <code>update</code> <code>Test.txt</code></li> <li>• <code>cv</code>s <code>edit</code> <code>Test.txt</code></li> <li>• <code>cv</code>s <code>editors</code> <code>Test.txt</code></li> </ul> <p>The commands should execute without any errors and without prompting the user for any extra input.</p> <p>The editors command should produce output that resembles the following:</p> <pre>Test1.txt myUserID Tue Aug 9 10:08:43 2009 GMT myComputer C:\myCVSWorkSpace\moduleName</pre>	
5	<p>Take note of the userID that follows the filename.</p> <p>Enterprise Architect must find and use this user ID when you create your version control configuration.</p>	<a href="#">CVS Settings</a> <sup>419</sup>

#### Notes

- Your environment must be set up such that you can perform these operations without ever being prompted for input, such as user ID or password

#### 3.7.1.2.3.3 TortoiseCVS

**TortoiseCVS** is a Windows shell extension for CVS; it provides icon overlays in Windows Explorer that are useful as a tool for observing the status of your CVS controlled files. You can also use it to create your repository modules and check out local working copies from within Windows Explorer using simple menu commands.

You can download TortoiseCVS from: <http://www.tortoisecvs.org>.

#### Notes

- Enterprise Architect must use the CVS command line client to communicate with the CVS server; it cannot use TortoiseCVS

### 3.7.1.2.4 Create a TFS Environment

You can use **Microsoft Team Foundation Server (TFS)** as a version control provider for Enterprise Architect. The first step in doing this is for a TFS administrator to install and configure the TFS server and client applications. A number of basic tasks are performed in creating an **operational TFS environment**.

#### Tasks in Creating a TFS Environment

Task	Detail	See also
<b>Obtain and install TFS</b>	Enterprise Architect uses the <b>TFS command line client</b> to integrate TFS version control.  The TFS command line client is normally available as part of your Visual Studio installation.	
<b>Choose a TFS project</b>	It is good practice to create a new <b>TFS project</b> , or least a new <b>Source Control Folder</b> within a project, for each Enterprise Architect project being added to version control with TFS.  If you have a single Enterprise Architect project that contains many different models (for example, a DBMS hosted project with multiple model root nodes), you might choose to create a new TFS project for each separate <b>model</b> .  For further information, please consult your TFS product documentation.	
<b>Create a TFS workspace</b>	A working copy folder must exist on each users' machine, for Enterprise Architect to use when exporting and importing the version controlled package files. It is this folder that is specified as the Local Project Path, when defining your Version Control Configurations.  The working copy folder is the 'sandbox' where you modify the controlled files. The working copy folder is usually associated with a folder that exists within the version control repository. In TFS, the TFS workspace is used to map a local working folder on your PC to a Source Control Folder within a TFS project.  A single TFS workspace <b>can</b> map many different local folders, each one to a separate Source Control Folder. In this case, TFS can take a long time to work through and update the files in all of those folders, and the system might appear to 'freeze' whilst it waits for TFS to hand back program control.  You can avoid this if you keep your version controlled <b>package</b> files in a folder that is separate from other artifacts, such as source code files, creating a <b>separate work space</b> to use <b>just</b> for your package files, or creating and mapping a separate folder for package files within an existing workspace.	<a href="#">TFS Workspaces</a> [409]
<b>Configure exclusive check-outs</b>	The XML format files used for the version control of Enterprise Architect's packages can not be merged like ordinary text files. Therefore, Enterprise Architect must enforce serialized editing of its version controlled packages. As a consequence, it is important that TFS is configured to use 'exclusive checkouts' for	<a href="#">TFS Exclusive Check Outs</a> [411]



Task	Detail	See also
	XML files.	

### Notes

- TFS can also be used with an SCC client; the MS TFS-SCC client is available for download from Microsoft's web site
- MDG Integration for Visual Studio 2005 or 2008 enhances TFS support by providing access to, for example, work items and bugs within both Enterprise Architect and the MDG Integration product

### Learn more

- [System Requirements](#)<sup>[395]</sup>

#### 3.7.1.2.4.1 TFS Workspaces

In order to use TFS to provide version control of the packages in a model, you need to prepare a functional **TFS workspace** that can be accessed through an Enterprise Architect **version control configuration** within that model. You use the TFS workspace to map a local working folder on your PC to a Source Control Folder within a TFS project repository.

It is assumed that a Team Project that you can use to control your Enterprise Architect package files already exists. You create the TFS workspace through **MS Visual Studio**.

#### Map a local folder to a TFS Source Control Folder

Step	Action	See also
1	Connect to your TFS server. From the main menu, choose <b>View   Team Explorer</b> .	
2	From the main menu, choose <b>View   Other Windows   Source Control Explorer</b> .	
3	From the main menu, choose <b>Tools   Connect to Team Foundation Server</b> . A prompt displays for you to log in to the Team Foundation server; another prompt then displays to choose an available Team Foundation server.	
4	Choose an appropriate server and click on the <b>OK</b> button. The selected Team Foundation Server is displayed in the Team Explorer pane. The server's project hierarchy is displayed in the Source Control Explorer's Folders pane.	
5	From the main menu, choose <b>File   New   Team Project</b> . The New Team Project wizard opens.	

6	Enter an appropriate name and description for the new Team Project. Choose the option to create an empty source control folder.	
7	Create a new Workspace. In the Source Control Explorer, click on the drop-down arrow in the <b>Workspace</b> field, then choose <b>Workspaces</b> The Manage Workspaces dialog is displayed.	
8	Click on the <b>Add...</b> button. The Add Workspace dialog is displayed.	
9	Type in an appropriate name for the new workspace and, if required, type in a comment.	
10	Click in the <b>Source Control Folder</b> column, then click on the <b>Browse</b> button to select a Source Control Folder. Select the new folder that was created in step 6.	
11	Click on the <b>Browse</b> button in the <b>Local Folder</b> column and create a new local folder. This is the working copy folder into which Enterprise Architect exports the package files.	
12	Click on the <b>OK</b> button. The new workspace is created and saved. The Add Workspace dialog closes.	
13	Click on the <b>OK</b> button. The Manage Workspaces dialog closes.	

### Notes

- The local folder referenced in step 11 is the Working Copy Path that should be specified when defining an Enterprise Architect Version Control Configuration to use this TFS workspace

### Learn more

- [TFS Settings](#) 

### 3.7.1.2.4.2 TFS Exclusive Check Outs

The XML format files used for the version control of Enterprise Architect's packages can not be merged as ordinary text files can. Therefore Enterprise Architect must enforce serialized editing of its version controlled packages, and it is important that Team Foundation Server is configured to use 'exclusive checkouts' for XML files. Otherwise, TFS can return file statuses that make it look as if the package file is not checked-out by another user when indeed it is.

You set exclusive checkouts in the TFS workspace through **MS Visual Studio**.

#### Configure TFS to enforce exclusive check outs for XML files

Step	Action	See also
1	Using Visual Studio, from the main menu select <b>View   Team Explorer</b> .	
2	In the Team Explorer pane, right-click on the TFS Server name that is controlling the Enterprise Architect package files, then from the context menu select <b>Team Foundation Server Settings   Source Control File Types</b> .	
3	Select the entry for XML files (or create an entry if necessary) then click on the <b>Edit</b> button.	
4	Clear the check-mark from the option <b>Enable file merging and multiple check out</b> .	

### 3.7.1.2.5 Create an SCC Environment

You can use a Microsoft Common Source Code Control (SCC) compatible product as a version control provider for Enterprise Architect. The first step in doing this is for an administrator to install and configure the server and client applications. A number of basic tasks are performed in creating an **operational SCC-based environment**.

#### Tasks in Creating an SCC Environment

Task	Detail	See also
<b>Install and configure your chosen version control product</b>	<p>A version control server component is typically installed on a dedicated server machine. All Enterprise Architect users who require access to version control must be able to connect to the server machine.</p> <p>After installing the version control software, the administrator should also create version control user IDs for all users who require access to version control.</p> <p>For further information, consult the product documentation for your particular version control product.</p>	<a href="#">System Requirements</a> (list of SCC compatible version control providers)
<b>Create a new SCC</b>	It is good practice to create a new SCC version control	

Task	Detail	See also
<b>project</b>	<p>project, or least a new folder within a project, for <b>each</b> Enterprise Architect project being added to version control with SCC.</p> <p>If you have a single Enterprise Architect project that contains many different models (for example, a DBMS hosted project with multiple model root nodes), you might choose to create a new SCC version control project for each separate <b>model</b>.</p> <p>For further information, consult the product documentation for your particular version control product.</p>	
<b>Configure your SCC project to support exclusive checkouts for .XML files</b>	<p>The XML-format files used for the version control of Enterprise Architect packages can not be merged like ordinary text files can. Therefore, Enterprise Architect must enforce serialized editing of its version controlled packages. As a consequence, it is important that your version control application is configured to use 'exclusive checkouts' for XML files.</p>	<a href="#">Version Control Basics</a> 387
<b>Create a local working copy folder</b>	<p>A working copy folder must exist on each users' machine, for Enterprise Architect to use when exporting and importing the version controlled package files. It is this folder that is specified as the Local Project Path, when defining your Version Control Configurations.</p> <p>The working copy folder is the 'sandbox' where you modify the controlled files. The working copy folder is usually associated with a folder that exists within the version control repository. Your version control product provides some means by which you associate a working copy folder with a repository folder.</p> <p>For further information, consult the documentation for your particular version control product.</p>	<a href="#">Version Control Setup</a> 413

#### Notes

- When installing the client component software on users' PCs, check that the SCC client is also installed, as it might not be a part of the default installation

#### Learn more

- [Upgrade at Enterprise Architect Version 4.5, Under SCC Version Control](#) 412

#### 3.7.1.2.5.1 Upgrade at Enterprise Architect Version 4.5, Under SCC Version Control

If you are working in Enterprise Architect release **4.5 or later** and you open an SCC version-controlled project that was created under a release **earlier** than 4.5, you must identify the SCC connection with a new unique ID. You can assign a name to the existing SCC configuration or associate the project with another configuration that has previously been assigned a unique ID.

By having a unique ID for each version control configuration, you can assign a configuration quickly and

efficiently using configurations that have been created previously for other version controlled repositories. This enables you to configure the many packages to use an existing version control repository; this can apply to packages created for more than just one model enabling a great deal of flexibility.

**Upgrade an existing SCC version controlled project created before release 4.5, in Enterprise Architect release 4.5 or later**

Step	Action	See also
1	Open the project that has an SCC version control configuration created in Enterprise Architect earlier than version 4.5.  The Select or Create Unique ID for Version Control dialog automatically displays.	
2	On the dialog, either create an ID for an existing configuration or choose a previously created one from the <b>Unique ID</b> drop-down list.	
3	The existing SCC configuration is the initial value, represented by <b>SCC-XXXXX</b> ; this number is not especially meaningful, therefore it is recommended that the configuration be given a meaningful name.	
4	You can associate the version controlled package with a previously-defined configuration by selecting an existing configuration from the <b>Unique ID</b> drop-down list (if one exists).	
5	After you have assigned the unique ID, click on the <b>OK</b> button to load the model.	

### 3.7.1.3 Version Control Setup

Once you or an Administrator have installed and configured the version control **product** software, to start using version control you must first define a **version control configuration** within your project in Enterprise Architect, to be used to control the packages in the project. You can define any number of version control configurations in a single model, but any given package is associated with only **one** configuration.

**Access**    **Project | Version Control | Version Control Settings**

**Define Version Control Configuration**

Step	Action	See also
1	On the Version Control Settings dialog, click on the <b>New</b> button.	
2	In the <b>Unique ID</b> field, type a suitable configuration name.	<a href="#">Re-use an Existing Configuration</a>

Step	Action	See also
		<a href="#">[414]</a>
3	Select the <b>Type</b> of version control product you are connecting to, by clicking on the corresponding radio button.	<a href="#">Version Control Settings</a> <a href="#">[415]</a>
4	At this point, the middle section of the dialog changes to display a collection of fields specific to the type of version control configuration you are defining.  Enter details relating to the version control workspace that this configuration is to use.	
5	Click on the <b>Save</b> button. The new configuration is added to the <b>Defined Configurations</b> list.	
6	If you want to create another version control configuration, return to step 1.  When you have finished defining your version control configurations, click on the <b>Close</b> button.	

### Notes

- Version control configuration details are stored in the user's Windows Registry settings, but each project stores a list of the configurations it uses, so that version control connections can be initialized as the project is being loaded
- If you are using the Corporate or extended editions of Enterprise Architect with **security** enabled, the administrator must also set up access permissions to configure and use version control

### Learn more

- [Version Control Submenu](#) [\[91\]](#)
- [User Security](#) [\[316\]](#)
- [List of Available Permissions](#) [\[329\]](#)

#### 3.7.1.3.1 Re-use an Existing Configuration

Once a version control configuration has been defined for use in one project, it is possible to re-use that configuration in other projects to provide access to:

- An already existing version control environment (a working copy directory and its associated repository that is already in use)
- Version controlled packages that were created (and version controlled) in another project

**Access** **Project | Version Control | Version Control Settings** or  
**Project Browser package context menu | Package Control | Version Control Settings**

**Select existing configuration**

Step	Action	See also
1	On the Version Control Settings dialog, click on the <b>New</b> button. The various fields on the dialog are cleared, ready for data entry.	
2	In the <b>Unique ID</b> field, click on the drop-down arrow and select one of the previously-defined version control configurations. The details of the selected configuration are displayed in the dialog.	<a href="#">Version Control Settings</a> <sup>[415]</sup>
3	Click on the <b>Save</b> button. The configuration details are saved and are ready for use in the current project.	

**3.7.1.3.2 Version Control Settings**

As part of the process of setting up a version control configuration on your model, or updating an existing version control configuration, you define a number of settings that control how the status of your model is communicated to your version control system. You define these settings using the Version Control Settings dialog.

**Access** **Project | Version Control | Version Control Settings** or  
**Right-click on any package node | Package Control | Version Control Settings**

**Configuration Options**

Field/Button	Action	See also
<b>This model is private</b>	Select to specify that this model database is to be accessed by just a single user ( <b>Private Model</b> ).  Leave unselected (the default) or deselect to specify that the database is to be accessed by multiple concurrent users ( <b>Shared Model</b> ).  If in doubt, use the default setting.	<a href="#">Version Control Usage</a> <sup>[385]</sup>
<b>Save nested version controlled packages to stubs only</b>	Select to specify that the exported XML file for a version controlled package will contain package stubs (place holders) for nested version controlled child packages (recommended).  Deselect to specify that the exported XML file will contain the full content of nested version controlled child packages.	
<b>For all packages,</b>	Select to force all XML 1.1 imports across the <b>model</b> to exclude	

Field/Button	Action	See also
<b>create placeholders for external references</b>	incoming relationships and instead create external references. If the <b>Create placeholders for missing External References during XML 1.1/2.1 Import</b> checkbox is not selected in the XML Specifications options for a <b>user</b> , this field overrides that setting.	<a href="#">XML Specifications</a> <sup>[640]</sup>
<b>Unique ID</b>	Specify a name that uniquely identifies the configuration. Either: <ul style="list-style-type: none"> <li>Type a name to identify a new configuration, or</li> <li>Click on the drop-down arrow and select the name of a configuration previously defined in a different project (if any exist)</li> </ul>	
<b>Type</b>	Click on the appropriate radio button for the type of version control system you are associating with this configuration.  The middle section of the dialog changes to display a collection of fields relating to the type of version control configuration you are defining.  Set the type to <b>SCC</b> for: <ul style="list-style-type: none"> <li>MS Visual Source Safe</li> <li>Rational Clear Case</li> <li>Perforce</li> <li>AccuRev</li> <li>Any other SCC-compatible clients</li> </ul> For any other product that you are using, select the type that matches the product - <b>CVS</b> , <b>Subversion</b> or <b>TFS</b> .	<a href="#">SCC Settings</a> <sup>[417]</sup>  <a href="#">CVS Settings</a> <sup>[419]</sup> <a href="#">SVN Settings</a> <sup>[420]</sup> <a href="#">TFS Settings</a> <sup>[421]</sup>
<b>New</b>	Click on this button to clear the fields and create a new version control configuration.	
<b>Save</b>	Click on this button to save the details of a new or updated configuration.	
<b>Delete</b>	Click on an entry in the <b>Defined Configurations</b> list and click this button to remove the definition of the selected configuration from this model.	
<b>Defined Configurations</b>	Review a list of configurations that are in use in the current model.	
<b>In future, do not prompt for incomplete configurations</b>	Select to specify that the user is not prompted to complete the definition of configurations that are not fully specified (the default).	



Field/Button	Action	See also
	Deselect to prompt the user to complete configurations that are not fully defined.	
<b>Close</b>	Close the Version Control Settings dialog.	
<b>Help</b>	Display this Help topic.	

### Notes

- It is important that, for any given version controlled package file, any user accessing that file from any model uses version control configurations having the same **Unique ID**
- When you first open a model that was created by another user and that uses version control, the version control configuration(s) used by that model do not yet exist in your Windows registry settings; you need to complete the definitions of those configurations before you can use version control in that project
- If User Security is enabled, you must have **Configure Version Control** permission to set up version control options for the current model
- It is possible to use multiple version control configurations in the same model

### Learn more

- [Configure Controlled Package](#) <sup>[424]</sup>
- [Apply Version Control to Branches](#) <sup>[426]</sup>
- [List of Available Permissions](#) <sup>[329]</sup>

#### 3.7.1.3.2.1 SCC Settings

When you are setting up your version control configurations on the Version Control Settings dialog, and you set the configuration type to **SCC**, the dialog presents a set of fields specific to SCC-based configurations. You can then define details such as:

- The working copy folder to be used with the configuration
- The details necessary to connect to the SCC version control system

You set the version-control configuration type to **SCC** for version control providers such as:

- MS Visual Source Safe
- Rational Clear Case
- Perforce
- AccuRev
- Any other SCC-compatible clients

**Access** **Project Browser package context menu | Package Control | Version Control Settings: Type:**  
**SCC**

**Settings**

Field/Button	Action	See also
<b>Local Project Path</b>	<p>Displays the full path of the folder that contains the local (working) copies of the XML package files.</p> <p>This field is read-only; its value can only be set through use of the <b>Select Path</b> button (below).</p>	
<b>Select Path</b>	<p>Click on this button to choose the Local Project Path, by opening a file browser dialog and navigating through the file system to the appropriate folder.</p> <ul style="list-style-type: none"> <li>After you choose the appropriate folder path, the Select SCC Provider dialog displays, listing all SCC providers that are installed on the current workstation; choose the SCC provider to use and click on the <b>OK</b> button</li> <li>At this point, the SCC client opens its own dialog to prompt you for information; SCC products implement this functionality in varied ways, but typically you are prompted to log in to the version control system, then prompted to choose the SCC project to use and possibly a (server) folder contained within that project</li> <li>At the conclusion of this process, all of the SCC details should be filled in; you can then save the definition by clicking on the <b>Save</b> button on the Version Control Settings dialog</li> </ul>	<p><a href="#">System Requirements</a> <sup>[395]</sup></p> <p><a href="#">Version Control Settings</a> <sup>[415]</sup></p>
<b>Current User</b>	<p>Displays the user name used to log on to the version control system that is accessed through this configuration.</p> <p>This field is read-only; the value it displays is retrieved from the SCC client.</p>	
<b>SCC Provider</b>	<p>Displays the name of the SCC provider.</p> <p>This field is read-only; the value it displays is retrieved from the SCC client.</p>	
<b>SCC Project</b>	<p>Displays the name of the SCC Project that this configuration attaches to.</p> <p>This field is read-only; the value it displays is retrieved from the SCC client.</p>	

**Notes**

- You define the SCC-specific details as part of the broader process of setting up a version control configuration on the Version Control Settings dialog

### 3.7.1.3.2.2 CVS Settings

When you are setting up your version control configurations on the Version Control Settings dialog, and you set the configuration type to **CVS**, the dialog presents a set of fields specific to CVS-based configurations. You can then define details such as:

- The working copy folder to be used with the configuration
- The path to the CVS command line client

**Access** **Project Browser package context menu | Package Control | Version Control Settings: Type:**  
**CVS**

#### Settings

Field/Button	Action	See also
<b>Working Copy Path</b>	Displays the full path of the folder that contains the local (working) copies of the XML package files.  This field is read-only; its value can only be set through use of the <b>Select Path</b> button (below).	
<b>Select Path</b>	Click on this button to choose the working copy path, by opening a file browser dialog and navigating through the file system to the appropriate folder.	
<b>Current User</b>	This field is read-only; its value is retrieved from a file named <i>CVS\Root</i> , located in the folder selected using the <b>Select Path</b> button, above.	
<b>CVS Exe Path</b>	Displays the full path of the CVS command line client executable file.  This field is read-only; its value can only be set through use of the <b>Select Path</b> button ( <b>below</b> ).	
<b>Select Path</b>	Click on this button to specify the path to the CVS command line client, by opening a file browser dialog and navigating through the file system to locate the appropriate file.	
<b>VC Time-Out Value</b>	Specify the amount of time that Enterprise Architect waits for a CVS command to complete; if the command does not complete within the allowed time, the system displays a <i>Time-out</i> error message, detailing the command that failed to complete.  This single time-out value is applied to all Version Control Configurations (of types SVN, TFS and CVS) that the user accesses from this workstation.	

#### Notes

- When connecting to a remote CVS repository, the **Current User** field should display the user name used to log into that repository; if this does not happen, it indicates that Enterprise Architect cannot extract the user name from the file ...*WorkingCopyPath*\CVS\Root and the configuration does not work correctly
- You define the CVS-specific details as part of the broader process of setting up a version control configuration on the Version Control Settings dialog

### 3.7.1.3.2.3 SVN Settings

When you are setting up your version control configurations on the Version Control Settings dialog, and you set the configuration type to **Subversion**, the dialog presents a set of fields specific to Subversion-based configurations. You can then define details such as:

- The working copy folder to be used with the configuration
- The path to the Subversion command line client

**Access** **Project Browser package context menu | Package Control | Version Control Settings: Type: Subversion**

#### Settings

Field/Button	Action	See also
<b>Working Copy Path</b>	Displays the full path of the folder that contains the local (working) copies of the XML package files.  This field is read-only; its value can only be set through use of the <b>Select Path</b> button (below).	
<b>Select Path</b>	Click on this button to choose the Working Copy Path, by opening a file browser dialog and navigating through the file system to the appropriate folder.	
<b>Subversion Exe Path</b>	Displays the full path of the Subversion command line client executable file.  This field is read-only; its value can only be set through use of the associated <b>Select Path</b> button ( <b>below</b> ).	
<b>Select Path</b>	Click on this button to specify the path to the Subversion command line client, by opening a file browser dialog and navigating through the file system to locate the appropriate file.	
<b>VC Time-Out Value</b>	Specify the amount of time that Enterprise Architect should wait for a Subversion command to complete; if the command does not complete within the allowed time, the system displays a <i>Time-out</i> error message, detailing the command that failed to complete.  This single time-out value is applied to all Version Control Configurations (of types SVN, TFS and CVS) that the user accesses from this workstation.	

Field/Button	Action	See also

### Notes

- You define the Subversion-specific details as part of the broader process of setting up a version control configuration on the Version Control Settings dialog

### 3.7.1.3.2.4 TFS Settings

When you are setting up your version control configurations on the Version Control Settings dialog, and you set the configuration type to **TFS**, the dialog presents a set of fields specific to TFS-based configurations. You can then define details such as:

- The working copy folder to be used with the configuration
- The user name and password to log in to the TFS server
- The path to the TFS command line client

**Access** **Project Browser package context menu | Package Control | Version Control Settings: Type:**  
**TFS**

### Settings

Field/Button	Action	See also
<b>Working Copy Path</b>	Displays the full path of the folder that contains the local (working) copies of the XML package files.  This field is read-only; its value can only be set through use of the associated <b>Select Path</b> button (below).	
<b>Select Path</b>	Click on this button to choose the Working Copy Path, by opening a file browser dialog and navigating through the file system to the appropriate folder.	
<b>Server Name</b>	Displays the name of the TFS Server that is associated with the working copy folder specified in the <b>Working Copy Path</b> field, above.  This field is read-only; Enterprise Architect retrieves the value it displays by querying the TFS client.	
<b>Workspace Name</b>	Displays the name of the TFS Workspace that is associated with the working copy folder specified in the <b>Working Copy Path</b> field, above.  This field is read-only; Enterprise Architect retrieves the value it displays by querying the TFS client.	
<b>User Name</b>	Specify the user name with which to log into the TFS Server.	

Field/Button	Action	See also
<b>Password</b>	Specify the password with which to log into the TFS Server.	
<b>TFS Exe Path</b>	Displays the full path of the TFS command line client executable file.  This field is read-only; its value can only be set through use of the associated <b>Select Path</b> button ( <b>below</b> ).	
<b>Select Path</b>	Click on this button to specify the path to the TFS command line client, by opening a file browser dialog and navigating through the file system to locate the appropriate file.	
<b>VC Time-Out Value</b>	Specify the amount of time that Enterprise Architect waits for a TFS command to complete; if the command does not complete within the allowed time, the system displays a <i>Time-out</i> error message, detailing the command that failed to complete.  This single time-out value is applied to all version control configurations (of types SVN, TFS and CVS) that the user accesses from this workstation.	

### Notes

- If you automatically log in to TFS through means external to Enterprise Architect (for example, through MS Integrated Security), you can leave the **User Name** and **Password** fields blank
- If the **Password** field is blank, Enterprise Architect retrieves the your Windows username and uses that value when determining whether a package file is checked out to you or to another user
- TFS version control can also be accessed using the TFS MSSCCI client; to make use of the TFS MSSCCI client, please define an **SCC** based version control configuration
- You define the TFS-specific details as part of the broader process of setting up a version control configuration on the Version Control Settings dialog

### Learn more

- [SCC Settings](#)<sup>[41]</sup>

### 3.7.1.4 Use Version Control

Once your version control product is installed and you have created a suitable environment, Enterprise Architect can make use of that environment to **control the Packages in your project**. Version control provides a **range of facilities**, as outlined in the following table.

### Facilities

Facility	Detail	See also
<b>Define Version Control Settings</b>	<b>Version control configurations</b> are used by Enterprise Architect to communicate with your version control system.	<a href="#">Version Control Setup</a> <sup>[43]</sup>

Facility	Detail	See also
	You define one or more version control configurations in your project and then use those configurations to control the Packages in your project.	
<b>Configure a Package</b>	To put a Package under version control you mark the Package as a <b>controlled package</b> , specify the version control configuration to control it, and associate an XML file with the Package.	<a href="#">Configure Controlled Package</a> <sup>[424]</sup>
<b>Check In a Model Branch</b>	<b>Checks in</b> all packages involved in a particular unit of work, as a single operation.  Checking-in updates the <b>reference</b> version of a package or group of packages in the model.	<a href="#">Check In a Model Branch</a> <sup>[434]</sup>
<b>Check Out a Model Branch</b>	Checks out all packages within a selected model branch as a single operation, so that you can update modeling objects within them.	<a href="#">Check Out a Model Branch</a> <sup>[433]</sup>
<b>Check In a Package</b>	<b>Checks in</b> the package currently selected in the Project Browser.  Checking-in updates the <b>reference</b> version of a package or group of packages in the model.	<a href="#">Check In a Package</a> <sup>[432]</sup>
<b>Check Out a Package</b>	<b>Checks out</b> the version controlled package currently selected in the Project Browser, so that you can update modeling objects within it.	<a href="#">Check Out a Package</a> <sup>[431]</sup>
<b>Undo Check Out of a Package</b>	<b>Reverses the check-out</b> of a package, discarding any modifications that have been made by restoring the package content to the latest revision held in version control.	<a href="#">Undo Check Out of a Package</a> <sup>[432]</sup>
<b>Import a Package From Version Control</b>	<b>Retrieves</b> packages from version control that have been created by other users, or by you in another model, and <b>imports</b> them into your current model.	<a href="#">Include Other Users Packages</a> <sup>[436]</sup>
<b>Apply Version Control to a Model Branch</b>	Applies version control to all packages within the selected model branch, in a single operation.  In this context, a model branch is a <b>package</b> in the Project Browser, and <b>all of the packages contained within it</b> .	<a href="#">Apply Version Control to Branches</a> <sup>[426]</sup>
<b>Export a Version Controlled Model Branch</b>	<b>Exports</b> version control information about the <b>root package</b> of a model branch, that is used to simplify the process of exporting and importing a hierarchy of packages from one model to another.	<a href="#">Export Controlled Model Branch</a> <sup>[437]</sup>

Facility	Detail	See also
<b>Import a Model Branch From Version Control</b>	Uses Enterprise Architect's <b>Model Branch files</b> , of which there are few, to retrieve information about the root package file and to import the model branch.  Model branch files can simplify the process of exporting and importing a hierarchy of packages from one model to another.	<a href="#">Import Controlled Model Branch</a> <sup>[438]</sup>
<b>View Package Revision History</b>	Displays the <b>change history</b> of version controlled packages.  You can also check out a prior revision of the package for editing, effectively <b>rolling-back</b> to a prior revision of the package.	<a href="#">Review Package History</a> <sup>[440]</sup>
<b>Validate Package Configurations</b>	You can test the <b>validity</b> of the version control settings associated with each version controlled package within your current model.	<a href="#">Validate Package Configurations</a> <sup>[443]</sup>
<b>Resynchronize the Status of Version Controlled Packages</b>	<b>Re-synchronizes</b> the version control status of packages within your project with the status reported by your version control provider.	<a href="#">Resynchronize Package Status</a> <sup>[444]</sup>

### Notes

- **Database replication** should not be combined with version controlled packages
- If the Packages under version control contain any **alternative images**, you can export the images to the version control repository when you check in the Packages, by setting the **Export alternate images** option on the Options dialog

### Learn more

- [Version Control Product Setup](#)<sup>[394]</sup>
- [Sharing Reference Data](#)<sup>[374]</sup>

#### 3.7.1.4.1 Configure Controlled Package

Once your version control application is set up and you have version control configurations in place, you can place the **individual packages** in your model under version control. To **put a package under version control**, you:

- Flag the package as a controlled package
- Specify the version control configuration to control it and
- Associate an XML file with the package

You can then export and import the package data to and from the file and issue commands to the version control system.



**Access** **Project Browser | right-click package | Package Control | Configure... (Ctrl+Alt+P)**

**Apply version control to a single package**

Step	Action	See also
1	Click on the <b>Control Package</b> checkbox to select it, indicating that this package is to be controlled.	
2	Click on the <b>Version Control</b> drop-down arrow and select the version control configuration to be used to control this package.	
3	<p>The <b>XMI Filename</b> field displays a default filename for the package export file, based on the package name.</p> <p>Optionally, modify the filename. Either:</p> <ul style="list-style-type: none"> <li>Type a new name, or</li> <li>Click on the browse button ( ... ) to open a file selection dialog</li> </ul> <p>The target file must be located within the working copy folder of the selected version control configuration, or one of its sub-folders.</p>	
4	<p>The <b>Version ID</b> field defaults to <b>1.0</b>.</p> <p>Optionally, change this to a different version number.</p>	
5	<p>The <b>Owner</b> field defaults to your user name.</p> <p>Optionally, type or select the name of the user who actually owns the package.</p>	
6	<p>Click on the <b>OK</b> button.</p> <p>The Add Package to Version Control dialog displays.</p>	
7	<p>Optionally, clear the <b>Keep checked out</b> checkbox.</p> <p>After applying version control, the package either remains checked-out for editing, or is checked-in and locked against editing, depending on this setting.</p>	
8	<p>Click on the <b>OK</b> button.</p> <p>The Add Comment dialog displays.</p>	
9	<p>Optionally, add any further comments to the default comment.</p> <p>Enterprise Architect provides a default comment that includes the current date &amp; time.</p>	

Step	Action	See also
10	<p>Click on the <b>OK</b> button.</p> <p>The current package is exported to the nominated XML file, which is then committed to version control. The package icon in the Project Browser is updated to reflect the package's version control status.</p>	<a href="#">Project Browser Indicators</a> [391]

### Notes

- If you are using the Corporate or extended editions of Enterprise Architect with security enabled, these features are only available to users who have been granted permission to configure and use version control

### Learn more

- [Use Version Control](#) [422]
- [List of Available Permissions](#) [329]

#### 3.7.1.4.2 Apply Version Control To Branches

It is possible to apply version control to all packages within a selected **model branch**, in a single operation. In this context, a model branch is a package that is currently selected in the Project Browser, and all of the packages contained within it.

**Access**    **Project Browser | right-click node package | Add Branch to Version Control**

#### Apply version control to all packages within a selected model branch

Step	Action	See also
1	On the Apply VC to Branch dialog, click on the drop-down arrow in the <b>Version Control Configuration</b> field and select the configuration to use.	
2	<p>Optionally, tick the checkbox <b>Export as Model Branch</b>.</p> <p>Once the version control operation is complete a Model Branch file (.EAB file) is created for this branch.</p>	<a href="#">Export Model Branch</a> [437] <a href="#">Import Model Branch</a> [438]
3	<p>Click on the <b>OK</b> button.</p> <p>The system creates a number of sub-folders within the version control working copy folder, then exports all of the packages within the selected model branch. The system generates filenames for the XML files, based on the package GUIDs.</p>	

**Notes**

- The version control configuration to be used in this operation must be defined within the model before selecting this command
- When invoked on the **model** root node, this command applies version control to every package within the model

**3.7.1.4.3 Package Version Control Options**

When you have set up a Package for version control, you gain access to a **range of version control operations** that can be performed on that Package, such as:

- Open the dialog for working with **baselines** of the Package
- Check-in and check-out **single** Packages or a selected **hierarchy** of Packages
- Update Packages to the **latest revision** from the version control repository
- Inspect the revision history or properties of the **XMI file** associated with a Package
- **Revert** a Package to a **previous** revision
- **Compare** the current model content of a Package, against the latest revision of the Package in version control
- **Import** and **export hierarchies** of Packages (**model branches**) to and from the model, through the version control system
- Synchronize the status of a Package, with the version control system

**Access**    **Project Browser | right-click version-controlled Package | Package Control**

**Options**

Option	Action	Shortcut	See also
<b>Configure</b>	Apply or remove version control for the selected Package (specify version control settings) or specify a file for use in XMI Package Control.	<b>Ctrl+Alt+P</b>	<a href="#">Configure Controlled Package</a> <sup>[424]</sup>
<b>Package Baselines</b>	Create a Baseline of the current Package, or compare the current Package with a previous Baseline.	<b>Ctrl+Alt+B</b>	<a href="#">Manage Baselines</a> <sup>[461]</sup>
<b>Check In Branch</b>	<p>Check-in Packages contained in the currently selected model <b>branch</b> (that is, the selected Package and all of its child Packages).</p> <p>The Select Packages to Check In dialog lists all version controlled Packages within that branch that are checked out to you; you can then select Packages in the displayed list, to be submitted for check-in.</p> <p>You can also choose to keep the Packages checked-out after committing a new revision to version control.</p>		<a href="#">Check In a Model Branch</a> <sup>[434]</sup>

Option	Action	Shortcut	See also
<b>Check Out Branch</b>	Recursively check out all Packages contained within the currently selected model branch (that is, the selected Package and all of its child Packages) that are version controlled and checked-in.		<a href="#">Check Out a Model Branch</a> <sup>[433]</sup>
<b>Check In</b>	Commit a new revision of the currently selected Package to the version control repository and lock the Package against further editing.  Only available for Packages that you have checked-out yourself.		<a href="#">Check In a Package</a> <sup>[432]</sup>
<b>Check Out</b>	Synchronize the currently selected Package with the latest revision from the version control repository and unlock the Package to allow editing.  Only available for Packages that are not already checked-out (and whose associated Package file is not checked-out).		<a href="#">Check Out a Package</a> <sup>[431]</sup>
<b>Undo Check Out</b>	Restore the selected Package to the latest revision in the version control repository and lock the Package against further editing.		<a href="#">Undo Check Out of a Package</a> <sup>[432]</sup>
<b>Put Latest</b>	Commit a new revision of the currently selected Package to the version control system, while keeping the Package checked-out.  This is equivalent to checking a Package in and immediately checking it back out again.  Only available for Packages that you have checked-out yourself.		
<b>Get Latest</b>	Synchronize the currently selected Package with the latest revision from the version control repository.  Available only for Packages that are checked in.		<a href="#">Update to the Latest Revision of Selected Package</a> <sup>[435]</sup>
<b>Get All Latest</b>	Update all of the version controlled Packages in the project, to the latest revision retrieved from version control.  Only updates Packages that are currently checked in.  Once the latest revisions are retrieved, the system scans all the controlled Packages and		<a href="#">Update to the Latest Revision of All Packages</a> <sup>[435]</sup>  <a href="#">The System Output Window</a> <sup>[169]</sup>

Option	Action	Shortcut	See also
	<p>fixes any missing cross-references by comparing the Package with its XML 1.1 file.</p> <p>If the cross-reference information in the XML does not match the model, the system updates the model with the information from the XML and records this update in the System Output window.</p> <p>You can roll back such updates by selecting the entry in the Output window and using the context menu option <b>Rollback Update</b> (or <b>Rollback Selected Updates</b> if multiple entries are selected).</p> <ul style="list-style-type: none"> <li>• Closing the model clears the entries in the System Output window</li> <li>• An entry in the System Output window is also cleared as and when you roll-back the update for it</li> </ul>		
<b>Scan XML and Reconcile Model</b>	<p>Scan the Package XML files associated with each of the project's controlled Packages and restore any diagram objects or cross-references that are detected as missing from the project.</p> <p>This function is useful in team environments where each user maintains their own private copy of the model database (i.e. multiple private project files) and model updates are propagated through the use of controlled Packages. It provides no benefit when the model is hosted in a single shared database that is accessed by all team members.</p> <p>Each controlled Package is compared with its associated XML file and, if the cross-reference information in the model does not match the XML, the system updates the model with the information from the XML and records the update in the Output window.</p> <p>You can roll back such updates by right-clicking on the entry in the Output window and selecting the context menu option <b>Rollback Update</b> (or <b>Rollback Selected Updates</b> if multiple entries are selected).</p> <p>Closing the model clears the entries in the Output window; an entry in the Output window is also cleared as and when you roll-back the update for it.</p> <p>This functionality is invoked automatically as part of the <b>Get All Latest</b> operation.</p> <p>When working in an environment that uses a Private Model deployment and your model contains a significant number of cross-Package references, it is recommended that you invoke</p>		<p><i>Version Control Best Practices</i> white paper  <a href="http://sparxsystems.com/WhitePapers/Version_Control.pdf">http://sparxsystems.com/WhitePapers/Version_Control.pdf</a></p>

Option	Action	Shortcut	See also
	<p><b>Scan XMI and Reconcile Model</b> from time to time, following the re-importation of controlled Packages - for example, after using <b>Get Latest</b> to update a number of Packages - or after performing a number of Package check-outs.</p> <ul style="list-style-type: none"> <li>As a general rule, avoid running <b>Scan XMI and Reconcile Model</b> while you have uncommitted changes in your model</li> <li>Generally, you should: <ul style="list-style-type: none"> <li>Check-out a number of Packages</li> <li>Invoke <b>Scan XMI and Reconcile Model</b></li> <li>Make your modifications</li> <li>Commit any outstanding changes before you check-out more Packages and run <b>Scan XMI and Reconcile Model</b> again</li> </ul> </li> </ul>		
<b>File Properties</b>	Display version control properties pertaining to the XMI export file associated with the currently selected Package; this also identifies who has checked out the Package.		
<b>File History</b>	<p>Display change history information for the currently selected Package.</p> <p>Revert to or check-out a prior revision of the Package.</p>		<a href="#">Review Package History</a> <sup>[440]</sup>
<b>Compare with Controlled Version</b>	Compare the currently selected Package with the latest revision of its associated XMI file retrieved from version control.		
<b>Add Branch to Version Control</b>	<p>Apply version control to all Packages within a selected <i>model branch</i>, in a single operation.</p> <p>In this context, a model branch is a Package that is currently selected in the Project Browser, and all of the Packages contained within it.</p>		<a href="#">Apply Version Control to Branches</a> <sup>[426]</sup>
<b>Export as Model Branch</b>	Export a newly created model branch from your own private copy of a model.		<a href="#">Export Controlled Model Branch</a> <sup>[437]</sup>
<b>Import a Model Branch</b>	Retrieve a model branch and import it into either the source model or another model.		<a href="#">Importing Controlled Model Branch</a> <sup>[438]</sup>

Option	Action	Shortcut	See also
<b>Get Package</b>	Access Packages in the version control repository that are not currently available in your model.		<a href="#">Include Other Users' Packages</a> <sup>[436]</sup>
<b>Re-synch Status With VC Provider</b>	Update the version control status value recorded for the selected Package in the project to match the value reported by the version control provider, without performing an XML import or export.  Use this function when the Package's version control status recorded in your project is out of synchrony with the version control status reported by your version control provider.		<a href="#">Resynchronize the Status of Version Controlled Packages</a> <sup>[444]</sup>
<b>Version Control Settings</b>	Display the Version Control Settings dialog.		<a href="#">Version Control Settings</a> <sup>[415]</sup>

**Notes**

- You set up Version Control using options from the **project Version Control** submenu
- If the selected Package is **not** under version control, a different set of options is available
- If a version control configuration has not been defined for the model, no options for **using** version control are available, only the options for **configuring** version control

**Learn more**

- [Version Control Submenu](#) <sup>[91]</sup>
- [Controlled Package Menu](#) <sup>[485]</sup>
- [Version Control Nested Packages](#) <sup>[390]</sup>

**3.7.1.4.4 Check Out a Package**

When you need to work on a version controlled package, you check it out. The local XML file associated with the package is then checked-out from version control. No other user can check out the package to make changes to it until it has been checked **in** again.

**Access** **Project Browser | right-click package | Package Control | Check Out**

**Check out a single package**

The package file is imported into your model, and the package icon is updated to reflect the change in the package's version control status.

When working in a **Private Model**, if the system detects that the package content in the model is **already** up to date with the latest revision of the package file retrieved from version control, then the Import Package dialog displays first. This dialog is not displayed for a **Shared Model**.

The following options are available:

- **Force Reload From XMI** - reload the package from XMI regardless of whether it is up to date or not
- **Accept current package** - select to skip the process of re-importing the package from XMI
- **Refresh model view** - select to refresh the Project Browser and diagrams, by reloading the package content from the project database
- **Always use these settings** - When selected, if you subsequently check out a package that is found to be up to date, the same settings are applied again without displaying the dialog

#### Notes

- If you check out a version controlled package whilst offline, the package icon has a red figure 8 in front of it
- If you have selected the **Always use these settings** checkbox and you want to **reconfigure** the Import Package dialog, press the **Ctrl** key whilst you select the **Package Control | Check Out** menu option; the dialog displays and you can change the settings

#### Learn more

- [Offline Version Control](#)<sup>[392]</sup>
- [Check Out Model Branch](#)<sup>[433]</sup>
- [Check In Package](#)<sup>[432]</sup>

#### 3.7.1.4.5 Undo Check Out of a Package

If you check out a package and then decide not to proceed, you can undo the check-out and discard any modifications that have been made, by restoring the package content to the latest revision held in version control. The package returns to a checked-in state and subsequently can be checked out by any user, including yourself if, for example, you need to reverse incorrect changes before checking the package out and starting again.

**Access** **Project Browser | right-click checked-out package | Package Control | Undo Check Out**

#### Undo check out of a selected package

A confirmation dialog displays; click on the **OK** button.

The latest revision of the package is retrieved from version control and re-imported into your model. The icon against the package in the Project Browser is updated to reflect the change in the package's version control status.

#### 3.7.1.4.6 Check In a Package

When you have finished working on the contents of a package under version control, and you want to return it to the model for other users to see, you check it in.

**Access** **Project Browser | right-click package | Package Control | Check In**

#### Check in a single package

Ste	Action	See also



Step		
1	<p>The selected package is exported and the Add Comment dialog displays. A default comment is provided that contains the current date and time.</p> <p>Optionally, modify the default check-in comment</p>	
2	<p>Click <b>OK</b>.</p> <p>The package file is checked-in to version control and the package icon is updated to reflect the change in version control status.</p>	

#### Learn more

- [Check In Model Branch](#)<sup>[434]</sup>
- [Check Out Package](#)<sup>[431]</sup>

#### 3.7.1.4.7 Check Out a Model Branch

If you need to check out a **number** of related packages involved in a particular unit of work, to update the contents, you can do so in a single operation by checking out the whole **model branch** that contains them.

**Access** **Project Browser | right-click root-node package | Package Control | Check Out Branch**

#### Check out a sub-tree of model packages

Step	Action	See also
1	<p>The selected root-node package and all of its contained sub-packages are recursively checked out.</p> <p>Any packages that cannot be checked-out are listed in a message box, with a brief description of the problem; for example: <i>The package is already checked out by user 'Fred'</i>.</p>	
2	<p>When Project Security is enabled in <i>Lock to Edit</i> mode, Enterprise Architect prompts you to apply a User Lock throughout the selected model branch before proceeding.</p>	

#### Learn more

- [Check Out Package](#)<sup>[431]</sup>
- [Check in Model Branch](#)<sup>[434]</sup>
- [Set Security Policy](#)<sup>[319]</sup>
- [Lock Model Elements](#)<sup>[336]</sup>

### 3.7.1.4.8 Check In a Model Branch

If you need to check in a **number** of related packages involved in a particular unit of work, and that you have updated, you can do so in a single operation by checking in the whole **model branch** that contains them. You can also commit new revisions of the affected packages as you complete milestones, whilst **keeping** the packages checked-out for further editing.

**Access**    **Project Browser | right-click root-node package | Package Control | Check In Branch**

#### Check in packages from within a model branch

Step	Action	See also
1	<p>The Select Packages to Check-in dialog lists all version controlled and checked-out packages within the selected model branch. By default, the entire list is selected.</p> <p>Optionally:</p> <ul style="list-style-type: none"> <li>Click an individual package to select just that package</li> <li><b>Ctrl+click</b> on an individual package to add or remove it from the selection</li> <li><b>Shift+click</b> on a range of packages to select them</li> <li>Click on <b>All</b> to select all listed packages</li> <li>Click on <b>None</b> to clear the selection</li> </ul>	
2	<p>Optionally, you can commit into version control a new revision of all selected packages, while keeping those packages checked out for further editing. To do this, select the <b>Keep packages checked-out after committing new revision</b> checkbox.</p>	
3	<p>Click on the <b>OK</b> button.</p> <p>The Add Comment dialog displays. A default comment is provided that contains the current date and time. This comment is applied to all packages that are checked in.</p>	
4	<p>Optionally, modify the default check-in comment.</p>	
5	<p>Click on the <b>OK</b> button.</p> <p>The selected packages are exported and checked-in. The package icons are updated to reflect any change in version control status. If you opted to keep packages checked out, there is no change in status.</p>	

#### Learn more

- [Check In Package](#)<sup>[432]</sup>
- [Check Out Model Branch](#)<sup>[433]</sup>

#### 3.7.1.4.9 Update to the Latest Revision of Selected Package

When you are part of a team working in a Distributed Model environment, you will want to periodically update your model with the changes that other team members have committed into version control. You can transfer the other users' updates from version control into the selected package in the Project Browser.

**Access**    **Project Browser | right-click package to update | Package Control | Get Latest**

##### Update package to latest revision

The local XML file associated with the package is updated to the latest revision from version control. The XML file is imported into your model database, updating the package in your model.

When working in a **Private Model**, if the system detects that the package content in the model is **already** up to date with the latest revision of the package file retrieved from version control, then the Import Package dialog displays first. This dialog is not displayed for a **Shared Model**.

The following options are available:

- **Force Reload From XML** - reload the package from XML regardless of whether it is up to date or not
- **Accept current package** - select to skip the process of re-importing the package from XML
- **Refresh model view** - select to refresh the Project Browser and diagrams, by reloading the package content from the project database
- **Always use these settings** - When selected, if you subsequently check out a package that is found to be up to date, the same settings are applied again without displaying the dialog

##### Notes

- The **Get Latest** command is disabled for any package that is **checked-out** (to anybody) in the currently loaded project
- When using a Shared Model environment, where all users are connected to a single model database, you should reload the package from the **database**, rather than using the **Get Latest** command
- If you have selected the **Always use these settings** checkbox and you want to **reconfigure** the Import Package dialog, press the **Ctrl** key whilst you select the **Package Control | Get Latest** menu option; the dialog displays and you can change the settings

##### Learn more

- [Update to the Latest Revision of All Packages](#) <sup>435</sup>

#### 3.7.1.4.10 Update to the Latest Revision of All Packages

When you are part of a team working in a Distributed Model environment, you will want to periodically update your model with the changes that other team members have committed into version control. You can transfer the other users' updates to all version controlled packages into the currently loaded project.

**Access**    **Project Browser | right-click package to update | Package Control | Get All Latest**

##### Update all packages in project to latest revision retrieved from version control

All the local XML files for all the version control configurations used in the project are updated to the latest

revision from version control. The system then scans the packages in the model, to determine which ones are up to date and which are not, compared to the latest revisions of the associated package files.

A prompt displays, providing the following import options for packages that are up to date:

- **Import changed files only**
- **Always import**
- **Prompt for each file**

Click on the **OK** button. The version controlled packages in your project are updated according to the option you selected; if you chose the **Prompt for each file** option, a prompt displays to confirm import of each file.

#### Notes

- There is no need to re-import packages that are already up to date - re-importing packages first deletes them from the project and then re-imports them from the XML file, which is time consuming as well as unnecessary; we strongly recommend using the default option **Import changed files only**
- The **Get All Latest** command does not update any package that is checked-out (to anybody) in the currently loaded project; otherwise, any changes not yet committed to version control would be discarded
- When using a **Shared Model** environment, where all users are connected to a single model database, the **Get All Latest** option should not be used - the information in the model database is always ahead of what is committed into version control; you should from time to time refresh your view of the model database, by reloading diagrams or reloading package content in the Project Browser

#### Learn more

- [Update to the Latest Revision of Selected Package](#)<sup>435</sup>

#### **3.7.1.4.11 Include Other Users' Packages**

Other users might be developing Packages in their own models that you could use in your model, or you might have other models containing Packages that you want to use in the current model. Unless you are sharing an SQL database or project file, those packages are not **automatically** available to you. However, if the Packages have been placed into **version control**, you can import them into your model as children of one of your model's Packages.

**Access**    **Project Browser | right-click parent Package | Package Control | Get Package**

#### Import Packages from version control into current model

Step	Action	See also
1	On the Get Shared File dialog, click on the drop-down arrow of the <b>Version Control Configuration</b> field and select the version control configuration associated with the Package to retrieve.  The file list is populated with the names of files available through that configuration, for retrieval and import into your model.	
2	Click on the Package file to import into your model and click on the <b>OK</b> button.	

	The Package file is imported as a new child Package, under the parent Package you selected.	
--	---	--

### Notes

- You must have access to the Package files through the version control system and you must define a version control configuration through which to access those files
- The version control configuration must use the same unique ID that was originally used to add the Package to version control
- XML Package files associated with Packages that are already part of your project, are NOT included in the list of files available for import

#### 3.7.1.4.12 Export Controlled Model Branch

Applying version control to a model can result in many XML files placed under version control. It might then be hard to locate and import the file corresponding to the root of a particular model branch. Using Model Branch Files (.eab files) overcomes this problem by making it easier to export and import package hierarchies from one model to another.

You could export a newly created model branch from your own private copy of a model so that, for example:

- Another user can import that branch into their own private copy of the same model
- It can be imported for inclusion as a common branch in a number of different models

**Access** [Project Browser | right-click package to export | Package Control | Export as Model Branch](#)

#### Create a Model Branch File to represent a package hierarchy stored in version control

Step	Action	See also
1	On the Export as Model Branch dialog, in the <b>EAB Filename</b> field, type a name for your Model Branch File.  Alternatively, click on ( ... ) and browse for the file location.  Note that the package name is supplied as a default.	
2	Click on the <b>OK</b> button.  A branch file is created to represent the selected package. The branch file is committed to version control using the same version control configuration that controls the package you selected.	

### Notes

- You can specify any file name, including sub-folder names, as long as the file is contained in or below the working folder of your version control configuration
- The facility is only enabled for packages that are already under version control

### Learn more

- [Import Model Branch](#)<sup>[438]</sup>

#### 3.7.1.4.13 Import Controlled Model Branch

Applying version control to a model can result in many XML files placed under version control. It could then be hard to locate and import the file corresponding to the root of a particular model branch, if you want to:

- Retrieve a model branch created by another user in a private copy of a model, to import it into your own private copy of the same model
- Retrieve a model branch that is common in many models, for inclusion in a new model

**Model Branch Files** overcome this problem by simplifying the retrieval of package hierarchies for use in other models. You use Enterprise Architect's Model Branch Files, of which there are few, to retrieve information about the **root package file** such as the name and type of the version control configuration for the selected package, and the relative filename of the version controlled XML file associated with the package. The system then uses this information to import the branch into your model.

### Prerequisites

Before you begin, you must have:

- An operational version control environment that can be accessed by Enterprise Architect, and
- All of the version controlled package files and the model branch file associated with the model branch to import, in a valid and accessible working copy folder

**Access** **Project Browser | right-click on target package for import | Package Control | Import a Model Branch**

### Import a Model Branch

Step	Action	See also
1	<p>On the Import VC Model Branch dialog, either:</p> <ul style="list-style-type: none"> <li>• Use the lower portion of the Import VC Model Branch dialog to select a model branch file</li> </ul> <p>(This is the simpler option if the associated version control configuration has already been saved in the current model; continue to step 2)</p> <p>OR</p> <ul style="list-style-type: none"> <li>• Click on the <b>Find a Model Branch (.EAB) file</b> button</li> </ul> <p>(This option is useful when you have not yet defined the version control configuration that is associated with the model branch to be imported; see the <i>Manually Locating Model Branch Files</i> topic)</p>	<p><a href="#">Manually Locating Model Branch Files</a><sup>[439]</sup></p>
2	Click on the drop-down arrow in the <b>Select a Version Control Configuration</b> field and select the configuration you need.	

	A list of .eab files controlled by that configuration is displayed in the <b>Select a Model Branch (.EAB) file</b> list.	
3	<p>Select the Model Branch File you need, then click on the <b>OK</b> button.</p> <p>Enterprise Architect imports the root package specified in the Model Branch File and recursively imports and populates all the sub-packages contained in the root package.</p>	

### Notes

- The **Import a Model Branch** command is only enabled for packages that you (the current user) are able to edit, as the imported model branch is inserted into the model under your selected package

### Learn more

- [Version Control Product Setup](#)<sup>[394]</sup>
- [Export a Model Branch](#)<sup>[437]</sup>
- [Version Control Nested Packages](#)<sup>[390]</sup>

#### 3.7.1.4.13.1 Manually Locating Model Branch Files

When importing a Model Branch File from version control, you might not have the associated version control configuration saved in the model that is receiving the import. In this situation, it is simpler to manually browse the file system to locate the Model Branch File (.eab) and let Enterprise Architect derive the details of the configuration from the branch file you select.

### Prerequisites

Before you begin, you must have:

- An operational version control environment that can be accessed by Enterprise Architect, and
- All of the version controlled package files and the model branch file associated with the model branch to import, in a valid and accessible working copy folder

**Access** **Project Browser** | **right-click on target package** | **Package Control** | **Import a Model Branch: Find a Model Branch (.EAB) file**

### Locate the Model Branch File

Step	Action	See also
1	Access the Import VC Model Branch dialog, then browse for and select the Model Branch File that represents the model branch to import.	<a href="#">Import Controlled Model Branch</a> <sup>[438]</sup>
2	<p>Click on the <b>Open</b> button.</p> <p>If the version control configuration referenced by the file is fully defined within</p>	

	the current model, the import commences at this point.  Otherwise, Enterprise Architect displays a dialog prompting you to complete the required configuration.	
3	Click <b>Yes</b> to proceed with completing the definition of the version control configuration.  The Version Control Settings dialog is displayed.	<a href="#">Version Control Settings</a> <sup>[415]</sup>
4	Complete the definition of the configuration.  (Typically this involves simply specifying the working copy folder.)	
5	Click on the <b>Save</b> button.  The configuration details are saved.  The import of the model branch proceeds.	

**Notes**

- The **Import a Model** Branch command is only enabled for packages that you (the current user) are able to edit, as the imported model branch is inserted into the model under your selected package

**Learn more**

- [Export a Model Branch](#) <sup>[437]</sup>
- [Version Control Product Setup](#) <sup>[394]</sup>

**3.7.1.4.14 Review Package History**

It is possible to review the change history of version controlled packages by examining previous revisions. If necessary, you can check out one of these earlier revisions of a package for editing, effectively **rolling-back** to that prior revision of the package.

**Access**    **Project Browser | right-click package | Package Control | File History**

**Review change history of a version-controlled package**

Step	Action	See also
1	For version control environments using Subversion, CVS or TFS command line clients, the File Version History dialog displays.  Click on a revision number in the <b>Revisions</b> field, to select that revision and view its log entry.  For version control environments using SCC based clients, your particular product opens its own File Version History dialog.	<a href="#">Review Package History - SCC Client</a> <sup>[441]</sup>



2	<p>On the File Version History dialog, you can optionally click on either:</p> <ul style="list-style-type: none"> <li>• <b>Check Out</b> - the selected revision of the package file is retrieved from version control and imported into your model as a package that is checked out for editing; you can subsequently check in this revision as a new HEAD revision, effectively allowing you to revert the package to a prior revision</li> </ul> <p>OR</p> <ul style="list-style-type: none"> <li>• <b>Retrieve</b> - the selected revision of the package file is retrieved from version control and imported into your model, but the package remains flagged as checked-in and cannot be modified; subsequently checking out the package updates it to the latest revision before it is unlocked for editing</li> </ul>	
---	---	--

### Notes

- If the selected package was already checked out in the current model, the **Retrieve** and **Check Out** buttons are disabled
- If the selected package contains any sub-package that is already checked-out in the current model, a warning will be displayed and the retrieval or check-out will not go ahead
- If you check out a prior revision of a package, but do not want to commit it as a new revision, right-click on the package and select **Package Control | Undo Check Out**

### Learn more

- [Undo Check Out of a Package](#)<sup>[432]</sup>

#### 3.7.1.4.14.1 Review Package History - SCC Client

It is possible to review the change history of version controlled packages by examining previous revisions. If necessary, you can check out one of these earlier revisions of a package for editing, effectively **rolling-back** to that prior revision of the package. The process for reviewing the change history of packages configured for version control with an SCC client (including products such as Visual Source Safe, TFS-SCC, ClearCase, Perforce, AccuRev and MKS Source Integrity) differs from that for Subversion, CVS or TFS command line clients.

**Access** **Project Browser | right-click package | Package Control | File History**

#### Review change history of a version-controlled package (SCC client)

Step	Action	See also
1	<p>The change history mechanism offered by the third party SCC provider displays.</p> <p>To import a prior revision of the package into your model, use the SCC History dialog to retrieve the revision, then close the dialog.</p>	<a href="#">Retrieve Prior Revision - SCC Client</a> <sup>[442]</sup> (Example)
2	The SCC client notifies Enterprise Architect that a different revision has been retrieved. A prompt then displays, asking whether you want to check-out the	

	prior revision.	
3	<p>Optionally, click on either:</p> <ul style="list-style-type: none"> <li>• <b>Yes</b>, to check out the prior revision - the selected revision of the package file is retrieved from version control and imported into your model as a package that is checked out for editing; you can subsequently check in this revision as a new HEAD revision, effectively reverting the package to the prior revision</li> </ul> <p>OR</p> <ul style="list-style-type: none"> <li>• <b>No</b>, to import the prior revision as read-only - the selected revision of the package file is retrieved from version control and imported into your model, but the package remains flagged as checked-in and cannot be modified; subsequently checking out the package updates it to the latest revision before it is unlocked for editing</li> </ul>	

### Notes

- If the selected package was already checked out in the current model, the system does not proceed with retrieving a prior revision
- If you check out a prior revision of the package, but do not want to commit it as a new revision, right-click on the package and select **Package Control | Undo Check Out**

### Learn more

- [Undo Check Out of a Package](#)<sup>[432]</sup>

Depending on your version control product, retrieving a prior revision of a controlled package can involve a number of prompts regarding overwriting the current local copy.

This example details retrieval of a prior revision from a TFS-SCC version control configuration.

**Access**   **Project Browser | right-click package | Package Control | File History**

### Example Procedure - retrieve prior revision, TFS-SCC client

Step	Action	See also
1	Display the TFS File History dialog.	<a href="#">Review Package History - SCC Client</a> <sup>[441]</sup>
2	<p>Click on the <b>Get</b> button.</p> <p>The TFS-SCC client displays the Resolve Conflicts dialog.</p> <p>This dialog offers an option to <b>Automerge All XML</b> package files. <b>DO NOT select this option.</b></p> <p>It is important to prevent any merging of Enterprise Architect's XML package</p>	

	files.	
3	Click on the <b>Resolve</b> button. The TFS-SCC client displays the Resolve writable file conflict dialog.	
4	Select the <b>Overwrite local file/folder</b> option. The existing working copy of the package file is <b>overwritten</b> by the prior revision retrieved from version control.	
5	Click the <b>OK</b> button. The TFS-SCC client redisplay the Resolve writable file conflict dialog; it should now show no conflicts.	
6	Click on the <b>Close</b> button. The TFS-SCC client redisplay the File History dialog.	
7	Click on the <b>Close</b> button. Enterprise Architect displays a prompt, asking whether to check-out the prior revision.	
8	Click on the: <ul style="list-style-type: none"> <li>• <b>Yes</b> button to check-out the prior revision</li> <li>• <b>No</b> button to retrieve a read-only version of the package, that is NOT checked-out and is NOT editable</li> </ul>	

#### 3.7.1.4.15 Validate Package Configurations

Having defined the version control settings for your current model, you can test the validity of those settings associated with each version controlled package within the model.

**Access** **Project | Version Control | Validate Package Configurations**

##### Validate version control settings

Step	Action	See also
1	The validation process scans the model database and verifies that the version control configuration associated with each version controlled package is fully specified in the current model. It also queries the corresponding version control provider to find the status of the package file associated with each version controlled package.  The results of the validation process are sent to the System Output window.	<a href="#">The System Output Window</a>

Step	Action	See also
		<a href="#">[169]</a>
2	Open the Version Control Settings dialog to complete the definition of any invalid or missing version control configurations.	<a href="#">Version Control Setup</a> <a href="#">[413]</a>
3	Click on an error message in the System Output window to highlight the corresponding package in the Project Browser.	
4	Right-click a package node and choose <b>Package Control   Configure Package</b> to open the Package Control Options dialog.  Correct any problems with the version control details for the package.  Correct any problems with the package's associated XML file.	<a href="#">Configure Controlled Package</a> <a href="#">[424]</a>

#### 3.7.1.4.16 Resynchronize the Status of Version Controlled Packages

It is possible to update the version control status of version controlled packages within your project to re-synchronize with the status reported by your version control provider. This can be useful if you are creating copies of your project, where checking in a package from one copy of the model leaves the package in the second copy of the model with an out-of-date version control status.

For a given package, the re-synchronization process queries the corresponding version control provider to find the status of the package file associated with the version-controlled package. If necessary, the process then updates the package flags within the model database, to synchronize the package status recorded in the model with the value reported by the version control provider.

**Access** **Project Browser | right-click package | Package Control | Re-synch Status With VC Provider**  
or  
**Project | Version Control | Re-synch Statuses of All Packages**

#### Resynchronize version control status

Step	Action	See also
1	The results of the re-synchronization process are sent to the System Output window.	<a href="#">The System Output Window</a> <a href="#">[169]</a>
2	Double-click on any result message to select, in the Project Browser, the corresponding package.	

#### Notes

- This process does **not** cause any package data to be either exported from your model to the

associated package file, or imported from a package file into your model's package data

- If a package has been checked-out and modified with Enterprise Architect, but your version control provider reports the package file as checked-in, running this process marks the package within Enterprise Architect as being checked-in, without exporting and committing the pending changes; subsequently checking-out the package imports the latest revision of the package file from version control, effectively discarding the uncommitted modifications from the model
- Similarly, if a package file is checked-out to you in your local working copy folder, but not in the Enterprise Architect model, running this process marks the package within the model as checked-out, but it does not import the associated package file from the version control system; consequently, it is possible to check-in a package from Enterprise Architect that is potentially out of date, compared to the latest revision of the package file within the version control system

### 3.7.2 Tracking Changes

If you want to track changes to data across your project, you can use two separate but complementary facilities - **Auditing** and **Baselines**.

#### Topics

Topic	Detail	See also
<b>Auditing of model changes</b>	<p><i>Auditing</i> is a project-level feature, available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions, that enables you to record model changes in Enterprise Architect.</p> <p>By enabling this feature, model administrators can view a range of information regarding changes, such as:</p> <ul style="list-style-type: none"> <li>• Who changed an element</li> <li>• How many elements they changed</li> <li>• When they changed the data</li> <li>• What the previous values were, and</li> <li>• What type of elements they changed</li> </ul>	<a href="#">Auditing</a> <sup>[446]</sup>
<b>Baselining and differencing to capture and roll back changes</b>	<p>The Enterprise Architect Corporate, Business and Software Engineering, System Engineering and Ultimate editions provide a facility to 'baseline' or snapshot a model branch in XML format at a particular point in time, and store it within the model in compressed format.</p> <p>More than one baseline can be stored against a single Enterprise Architect package; using baselines, you can compare packages at the current and earlier stages of development, using the <i>Compare</i> (Diff) utility.</p> <p>The Compare utility is available in the Professional, Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect; it enables you to compare the current model with:</p> <ul style="list-style-type: none"> <li>• A Baseline</li> <li>• An exported Enterprise Architect XML file on disk</li> <li>• A version-controlled Enterprise Architect XML file on disk</li> </ul>	<a href="#">Package Baselines</a> <sup>[457]</sup> <a href="#">The Compare Utility (Diff)</a> <sup>[463]</sup>

Topic	Detail	See also

### 3.7.2.1 Auditing

**Auditing** is a project-level feature that model administrators can use to record model changes in Enterprise Architect. After switching Auditing on, you can view information on changes such as:

- Who changed an element
- How many elements they changed
- When they changed the data
- What the previous values were, and
- What type of elements they changed

Auditing does **not** record changes to:

- Document Templates
- Model Documents
- Baselines or
- Profiles

We provide a Quickstart procedure to help you enable Auditing and see it in action, from which point you can explore a number of set-up options and display features.

#### Features

Feature	Detail	See also
<b>The Audit View</b>	To view what has been recorded by the audit, use the <b>Audit View</b> , which provides an interface to everything recorded by Auditing.  If security is enabled, you must have <b>Audit View</b> permission to display data in the Audit View.	<a href="#">Audit View</a> <sup>[451]</sup> <a href="#">List of Available Permissions</a> <sup>[329]</sup>
<b>Model Views</b>	You can also obtain a snapshot of selected items in the model, using the <b>Model View</b> facility.  In the Corporate, Business and Software Engineering, Systems Engineering or Ultimate editions of Enterprise Architect, you can use this facility to automatically generate a snapshot at intervals and, if there are changes in the items collected by the defined search, to trigger a notification to you of such changes.  This helps you to monitor work flow and other events of concern to you.	<a href="#">Model View</a> <sup>[686]</sup>  <a href="#">Monitor Events</a> <sup>[593]</sup>
<b>Document Report</b>	You can generate a document report that includes the audit history information for the selected element or Package, by choosing the <b>basic + audit</b> template.	<a href="#">Generate Documentation</a> <sup>[2644]</sup>

Feature	Detail	See also
<b>Audit History</b>	Using Auditing, you can track changes to an element or connector over time.  However, enabling Auditing also enables an Audit History tab in the System Output window, which summarizes all changes made to the selected element or connector.	<a href="#">Audit History Tab</a> <sup>[455]</sup>
<b>Performance Issues</b>	By enabling Auditing on a project, you increase the time taken for most actions.  For most modeling tasks, this increase is insignificant; however, there are some instances where the difference is more substantial.	<a href="#">Performance Issues</a> <sup>[456]</sup>

### Notes

- The Auditing facility is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions
- Warning** - If your site runs separate editions of Enterprise Architect, when Auditing is turned on in a project any Desktop or Professional edition users are locked out of the project; to restore access, turn Auditing off in the project from the Corporate, Business and Software Engineering, Systems Engineering or Ultimate edition instance of Enterprise Architect

### Learn more

- [Auditing Quickstart](#) <sup>[447]</sup>
- [Auditing Settings](#) <sup>[448]</sup>

#### 3.7.2.1.1 Auditing Quickstart

It is very simple to enable Auditing and see it in action.

**Access** **Project | Auditing: Audit Settings**

### Enable Auditing

Step	Action	See also
<b>1</b>	In the Audit View, click on the <b>Audit Settings</b> button.  The Audit Settings dialog displays.	<a href="#">The Audit View</a> <sup>[451]</sup>  <a href="#">Auditing Settings</a> <sup>[448]</sup>
<b>2</b>	Select the <b>Enable Auditing</b> checkbox.	
<b>3</b>	Click on the <b>OK</b> button to close the Audit Settings dialog.	

Step	Action	See also
4	Close the Audit View dialog.	
5	Change and save your project. <ul style="list-style-type: none"> <li>• Add a new Package</li> <li>• Add a new Class</li> <li>• Add a new connector</li> <li>• Change the name of an element</li> <li>• Delete an element</li> </ul>	
6	Open the Audit View again and click on the <b>Refresh</b> (or <b>Load</b> ) button to display a record of the changes you have made.	

### 3.7.2.1.2 Auditing Settings

Using the Audit Settings dialog, you can change what is recorded by the Auditing facility and:

- Define the areas of processing in Enterprise Architect to audit
- Administer your audit records
- Indicate the kind of model objects for which changes are to be recorded
- Configure Auditing to record changes to only certain types of elements

**Access**    **Project | Auditing: Audit Settings**

#### Configure Settings

Field/Option/ Button	Action	See also
<b>Enable Auditing</b>	Select this checkbox to turn the Auditing facility on.	
<b>Audit XMI Import</b>	Select this checkbox to record changes arising from <b>XMI importing</b> in the audit.  As version control uses XMI, you must select this option to record changes from checking out Packages.	<a href="#">Version Control</a> <sup>383</sup>
<b>Audit XMI Export</b>	Select this checkbox to record changes arising from <b>XMI exporting</b> in the audit.  As version control uses XMI, you must select this option to record changes from checking out Packages.	



Field/Option/ Button	Action	See also
<b>Audit Reverse Engineering</b>	Select this checkbox to record changes arising from <b>reverse engineering</b> in the audit.	
<b>Use Database Timestamp</b>	Select this checkbox to use the <b>database server's</b> timestamp instead of each user's <b>local</b> timestamp; this improves security.  This option is not available for project <b>files</b> .	
<b>Save Logs</b>	Click on this button to save a copy of the logged audit items currently held in the project. These items remain in the project; you can use the <b>Clear Logs</b> button to remove them.  The system prompts you to specify whether to save items covering a specific period of time. <ul style="list-style-type: none"> <li>Click on the <b>No</b> button to save all log items currently held in the database</li> <li>Click on the <b>Yes</b> button to display the Time Filter dialog, on which you select a standard time period or define your own</li> </ul> This function can be accessed through the Automation Interface.	<a href="#">Audit View Controls</a> <sup>[453]</sup>
<b>Clear Logs</b>	Click on this button to permanently delete all log data from the current project; use the <b>Save Logs</b> button first, to save the audit records outside the project.  The system prompts you to specify whether to clear items covering a specific period of time. <ul style="list-style-type: none"> <li>Click on the <b>No</b> button to clear all log items currently held in the database</li> <li>Click on the <b>Yes</b> button to display the Time Filter dialog, on which you select a standard time period or define your own</li> </ul> This function can be accessed through the Automation Interface.	<a href="#">Repository</a> <sup>[2850]</sup>
<b>Load Logs</b>	Click on this button to load a previously saved set of logs back into the project. A browser displays through which you select the log file to reload.  If the same record exists in both project and log file, that record is not reloaded.	
<b>Core</b>	Select this radio button to record changes to elements (including attributes and operations), Packages, connectors and some model-level information.	
<b>Standard</b>	Select this radio button to record the same changes as for the <b>Core</b>	

Field/Option/ Button	Action	See also
	option, plus changes to diagrams.	
<b>Extended</b>	Select this radio button to record the same changes as for the <b>Standard</b> option, plus changes to project security.	
<b>Maintenance</b>	Select this radio button to audit changes to maintenance elements only; that is: <ul style="list-style-type: none"> <li>• Package (element)</li> <li>• Requirement</li> <li>• Feature</li> <li>• Use Case</li> <li>• Actor</li> <li>• Note</li> <li>• Issue and</li> <li>• Change</li> </ul>	
<b>Core Structural</b>	Select this radio button to audit changes to maintenance elements (as above) plus certain structural elements; that is: <ul style="list-style-type: none"> <li>• Package (structure)</li> <li>• Class</li> <li>• Interface</li> <li>• Signal</li> <li>• Node</li> <li>• Component</li> <li>• Artifact</li> <li>• Part</li> <li>• Port and</li> <li>• Device</li> </ul>	
<b>All</b>	Select this radio button to audit changes to all types of element.	
<b>Custom</b>	Select this radio button to audit changes to element types that you specify.  The <b>Customize</b> button is made available; click on this button to display a list of element types, and select the checkbox against each element type to include in the audit (or click on the <b>Select All</b> button to select every element type).  Click on the <b>OK</b> button to save the selection.	

**Notes**

- As the number of records increases, the performance of the Audit View reduces; it is recommended that audit records that are not regularly required are saved to file, then cleared from the project, to help ensure high performance
- Connectors are audited when they are connected to an element that is included in the **Audit Options**
- If security is enabled, you must have **Audit View** permission to turn Auditing on, and **Audit Settings** permission to change the audit settings

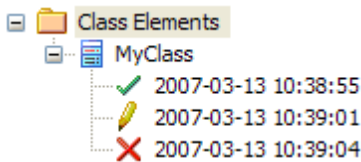
**3.7.2.1.3 The Audit View**

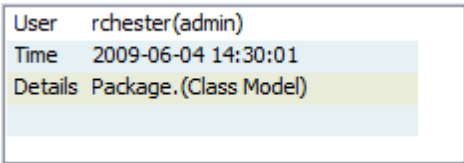
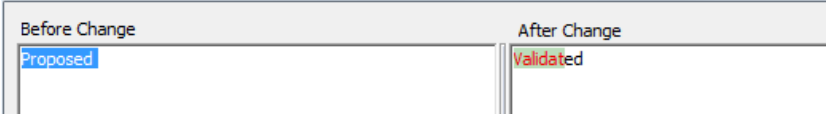
The Audit View provides an interface to the information that has been recorded by auditing.

**Access** **Project | Auditing**

**Sections of the Audit View**

The Audit View is divided into three main areas:

Section	Description	See also
<b>View controls</b>	The view controls provide a variety of settings for controlling auditing and the display of audit records.	<a href="#">Audit View Controls</a> <sup>[453]</sup>
<b>Audit tree</b>	<p>The audit tree displays the log items that have been recorded by auditing. What is displayed in the tree is affected by the view controls, such as:</p> <ul style="list-style-type: none"> <li>• Sorting</li> <li>• Filter (by time)</li> <li>• Mode</li> <li>• Auditing settings (what was actually recorded)</li> </ul> <p>If synchronizing with the Project Browser, it is also affected by the Package, diagram or element you have selected.</p>  <p>In the audit tree:</p> <ul style="list-style-type: none"> <li>• The green tick indicates a creation</li> <li>• The yellow pencil indicates an edit</li> <li>• The red cross indicates a deletion</li> </ul>	<a href="#">Audit View Controls</a> <sup>[453]</sup> <a href="#">Auditing settings</a> <sup>[448]</sup> <a href="#">The Project Browser</a> <sup>[646]</sup>

Section	Description	See also
	<p>Right-clicking an element in the audit tree (such as <i>MyClass</i>) displays a context menu that enables you to locate the selected element in:</p> <ul style="list-style-type: none"> <li>• The Project Browser</li> <li>• Any diagrams in which it exists</li> </ul>	
<b>Record display</b>	<p>The record display is in two parts: the identity of the selected change, and the actual change made.</p> <p>The data in the record display is determined by the view controls and mode and, if synchronizing with the Project Browser, by the package, diagram or element you have selected.</p> <p><b>Identity</b></p>  <p>The identity of a change consists of:</p> <ul style="list-style-type: none"> <li>• The Windows username of the user that made the change; if security is enabled, the name is of the format <i>WindowsUsername (SecurityUser)</i></li> <li>• When the change was made</li> <li>• The path of the change; for example: Class <i>Class1</i> - Attribute <i>Att1</i> - Attribute Constraint <i>Constraint</i></li> </ul> <p><b>Changes</b></p> <p>Changes are displayed in a table format, showing the:</p> <ul style="list-style-type: none"> <li>• Property (or data item) name</li> <li>• Its original value before the change and</li> <li>• Its value after the change</li> </ul> <p>If you double-click on an item in the list of changes (or right-click and select the <b>Show Differences</b> context menu option) the Difference window displays:</p>  <p>This shows the specific changes that have been made, highlighting the edited, created, deleted or formatted characters.</p>	<p><a href="#">Audit View Controls</a> <sup>[453]</sup></p> <p><a href="#">The Project Browser</a> <sup>[646]</sup></p>

Section	Description	See also
	Changes to the <i>format</i> of text in the element, made through the element Properties dialog, are not apparent in the initial table; they are visible in the Difference window, identified by HTML formatting tags.	

### Notes

- If security is enabled, you must have **Audit View** permission to display data in the Audit View

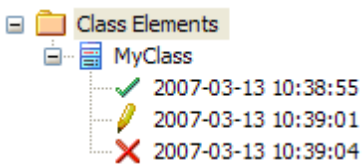
### Learn more

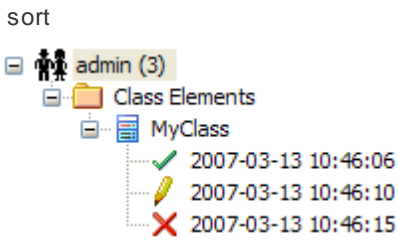
- [List of Permissions](#) <sup>[329]</sup>

#### 3.7.2.1.3.1 Audit View Controls

The **Audit View** controls provide a variety of settings for controlling auditing and the display of audit records.

### Reference

Action	Usage	See also
<b>Load or Refresh</b>	Click to reload the Audit Tree, updated with any new audit results.	
<b>Search</b>	Click to search through log items for a particular area - you can search by Name, Type or GUID.  The items are loaded and filtered with the current Sort By, Time Filter and Mode settings.  If you refresh the Audit View, you must run the search again.	
<b>Audit Settings</b>	Click to open the Audit Settings dialog.	<a href="#">Audit Settings</a> <sup>[448]</sup>
<b>Sort-by</b>	Select the appropriate radio button for the required display setting: <ul style="list-style-type: none"> <li><b>Type</b> - changes are grouped under element type (such as Class or Requirement), and then grouped under the changed element    </li> <li><b>User</b> - changes are grouped under user name, each with the number of changes for that user  Under each user name, changes are grouped as for the Type</li> </ul>	

Action	Usage	See also
		
<b>Filter By Date/Time</b>	<p>Select to enable the <b>Filter Settings</b> button, to filter the audit results by time period.</p> <p>Changes that occur <i>outside</i> the selected filter period are not shown in the Audit View.</p>	
<b>Filter Settings</b>	<p>Click to display the Time Filter dialog, to set the filter time period:</p> <ul style="list-style-type: none"> <li>• <b>Today</b> - to display changes occurring today</li> <li>• <b>Previous Hour</b> - to display changes occurring in the last 60 minutes</li> <li>• <b>Previous 24 Hours</b> - to display changes occurring in the last 24 hours</li> <li>• <b>Previous Week</b> - to display changes occurring in the last 7 days</li> <li>• <b>Previous 30 Days</b> - to display changes occurring in the last 30 days</li> <li>• <b>Previous Year</b> - to display changes occurring in the last 365 days</li> <li>• <b>Custom</b> - to define your own time period, using the <b>From</b> and <b>To</b> fields</li> </ul> <p>The six pre-configured time periods automatically update when you click on the <b>Refresh</b> button; custom periods are static and do not automatically update.</p> <p>If you have set a filter period, and you deselect the <b>Filter By Date/Time</b> checkbox, the period remains set; the custom time period, too, is retained so that you can re-use it or modify it later if required.</p>	
<b>Status Text</b>	<p>Read to see which:</p> <ul style="list-style-type: none"> <li>• Mode has been selected and</li> <li>• Time filter is being applied to the data</li> </ul>	
<b>Mode</b>	<p>Click to display a menu of options to change the mode of the Audit View.</p> <p>Select:</p> <ul style="list-style-type: none"> <li>• <b>Standard</b> - to automatically synchronize with the Project Browser; where changes have been made, the Audit View reflects your selection from the Project Browser - if you click on:</li> </ul>	<a href="#">The Audit View</a> <sup>#45↑</sup>


Action	Usage	See also
	<ul style="list-style-type: none"> <li>• An element, the Audit View displays the history for that element</li> <li>• A package, the Audit View displays the history for that package and its immediate children (but not the contents of nested packages)</li> <li>• A diagram, the Audit View displays the history for that diagram and its contents (which could be drawn from a wide area of the Project Browser)</li> <li>• <b>Advanced</b> - to load large sets of log items independent of the Project Browser; in this mode, a special Audit Settings group can be displayed in the Audit Tree, which records: <ul style="list-style-type: none"> <li>• When Auditing has been enabled and disabled</li> <li>• Who made the change</li> <li>• The date and time of the change</li> <li>• Changes to the Audit Settings</li> <li>• When Audit Operations are executed</li> <li>• Security changes (which can be browsed in the same way as other changes)</li> </ul> </li> <li>• <b>Deleted</b> - to display only deleted records, but otherwise data is shown as in <b>Advanced</b> mode; records can be sorted by element type or by user as required</li> <li>• <b>Raw</b> - to display all audit records in chronological order without sorting, enabling you to see a progression of changes; this can be especially useful in determining date-time inconsistencies</li> </ul> <p>Any search and filtering you define still apply, enabling you to view all of today's changes in order, or all changes for a particular element in order, or both</p> <p>Additional database information is displayed; this additional information might be insignificant or only in machine-readable format</p>	

### 3.7.2.1.4 Audit History Tab

When Auditing is turned on, an Audit History tab is enabled in the System Output window.

**Access** [View | System Output > Audit History](#)

#### Guide

Topic	Detail	See also
<b>Prerequisite</b>	To see this tab, you must have the Audit View open.	<a href="#">The Audit View</a> 

Topic	Detail	See also
	The tab continues to display if you subsequently close the Audit View.	
<b>Information Shown</b>	<p>The tab shows a history of changes to whichever element or connector you have selected in the:</p> <ul style="list-style-type: none"> <li>• Current diagram</li> <li>• Project Browser</li> <li>• Audit View, or</li> <li>• Package Browser</li> </ul> <p>As you select different elements or connectors, the Audit History tab automatically updates to reflect your current selection.</p> <p>For each change made to the element or connector, the tab shows:</p> <ul style="list-style-type: none"> <li>• Who made the change</li> <li>• When the change was made</li> <li>• Where the change was made</li> <li>• The value of the characteristic before the change</li> <li>• The value of the characteristic after the change</li> </ul>	

#### Notes

- If security is enabled, you must have **Audit View** permission to display data on the Audit History tab

#### Learn more

- [System Output Window](#)<sup>[169]</sup>
- [List of Permissions](#)<sup>[329]</sup>

#### 3.7.2.1.5 Auditing Performance Issues

Enabling auditing on a project increases the time taken for most actions.

For most modeling tasks, this increase is insignificant; however, there are some situations where the difference is more substantial.

Operation	Detail	See also
<b>Large Deletions</b>	<p>Deleting large Packages or Package structures, or large numbers of elements, takes significantly longer with auditing on.</p> <p>You might disable auditing before performing such a deletion.</p>	<a href="#">Auditing Settings</a> <sup>[448]</sup>
<b>XML Imports</b>	Importing XML takes longer with auditing enabled.	<a href="#">Import From XML</a>



Operation	Detail	See also
	A project-level option is provided for disabling auditing of XML Imports.	<a href="#">Auditing Settings</a>
<b>Reverse Engineering</b>	Reverse engineering code takes longer with auditing enabled. A project-level option is provided for disabling auditing of reverse engineering.	<a href="#">Import Source Code</a> <a href="#">Auditing Settings</a>
<b>Replication</b>	You cannot remove replication from a model with Auditing enabled. If you have to remove replication, disable Auditing and - if prompted to do so - allow Enterprise Architect to roll back the database version; you can then remove replication.	<a href="#">Remove Replication</a> <a href="#">Auditing Settings</a>

### 3.7.2.1.6 Audit View Performance Issues

Most operations in the Audit View are affected by the volume of use of the database - both by other facilities and by auditing itself.

Some potential considerations and responses are outlined below:

Consideration	Detail	See also
<b>Navigating the Project Browser Within Auditing is Slow</b>	Try setting the time filter to a period in the immediate past, such as <b>Today</b> , <b>Previous 24 Hours</b> or <b>Previous Week</b> ; this time period updates each time you open or refresh the Audit View.	<a href="#">Audit View Controls</a> <a href="#">Auditing Settings</a>
<b>The Audit View is Slow in Loading and Changing Modes</b>	Save log items outside the project with the <b>Save Logs</b> button; if you then clear the logs you have just saved, the load time of the Audit View is reduced.  You can reload logs into the project at any time, using the <b>Load Logs</b> button.	
<b>Navigating the Audit Tree is Slow</b>	Close the Audit History tab in the Output window	<a href="#">Audit History Tab</a>

### 3.7.2.2 Package Baselines

Enterprise Architect includes tools to help you manage and review changes to your models over time. These tools apply the concepts of *Baselines*, *Differencing* and *Merges*.

You use Baselines, Differencing and Merges essentially to compare two snapshots of a specific part of your project, to capture the differences between them and either roll back or incorporate selected changes or all changes.

Topic	Detail	See also
<b>Baselines</b>	Enterprise Architect provides a facility to create a Baseline or	<a href="#">Baselines</a>

Topic	Detail	See also
	<p>'snapshot' of the contents of a selected package and its child packages at a particular point in time; this enables you to later compare that branch of the model at that time with the current state of the branch.</p> <p>Baselines are stored in the same XML format as is used for version control, but are stored within the project in compressed format.</p> <p>You can also have parallel copies of parts of your model for team development, and create Baselines within each copy to merge changes into the project master.</p>	
<b>Differencing</b>	<p>Differencing (<i>Diff</i>, or <i>Compare</i>) enables you to explore the differences between:</p> <ul style="list-style-type: none"> <li>• The current state of a specific part of your project, and</li> <li>• Previous or parallel versions captured in a Baseline or an XMI 1.1 file on disk</li> </ul> <p>The section includes an example of such a comparison.</p>	<p><a href="#">The Compare Utility (Diff)</a> <sup>[463]</sup></p> <p><a href="#">Example Comparison</a> <sup>[468]</sup></p>
<b>Merges</b>	<p>Once Differencing is complete, you can merge information from the Baseline into the current project; it is not possible to go the other way.</p> <p>You can:</p> <ul style="list-style-type: none"> <li>• Merge information manually, change by change</li> <li>• Merge information automatically by electing to merge in all changes in one batch procedure</li> <li>• Revert completely to the original Baseline by importing the stored XMI directly</li> <li>• Merge information and elements from a Baseline in a different project, making it possible to keep multiple versions of a single model in synch</li> </ul> <p>The merge options are available through the toolbar, context menus and the keyboard on the Compare Utility tab, which shows the results of a comparison.</p>	<p><a href="#">Compare Utility Tab Options</a> <sup>[470]</sup></p>
<b>Visual Differences in Diagrams</b>	<p>Changes to a model might include:</p> <ul style="list-style-type: none"> <li>• Adding or removing elements and connectors on a diagram, or</li> <li>• Changing the position of elements or the overall layout of a diagram</li> </ul> <p>You might believe that a diagram has changed, and select to compare it with a baseline using a context menu option from the Project Browser. Alternatively, you might perform a</p>	<p><a href="#">Check Visual Changes to Diagrams</a> <sup>[466]</sup></p>

Topic	Detail	See also
	baseline comparison on a package or a model and select from the comparison output any diagrams that are flagged as changed.	

### Notes

- Package Baseline facilities are available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect
- The Enterprise Architect Corporate, Business and Software Engineering, System Engineering and Ultimate editions provide another facility, Auditing, which you can switch on to perform continuous monitoring of changes across the project; you can dovetail your use of each facility to meet the range of your change management requirements
- If a package under version control forms part of a Baseline, and that package is checked in to the model, you cannot merge the original data from the Baseline into that package
- You can also obtain a snapshot of selected items in the model, using the Model Views facility; this facility enables you to automatically generate the snapshot at intervals and, if there are changes in the items collected by the defined search, to trigger a notification to you of such changes, which enables you to monitor work flow and other events of concern to you
- If security is enabled you must have **Baselines - Manage** permission to create, import and delete Baselines, and **Baselines - Restore** permission to merge data from a Baseline; security permissions are not required to select an existing Baseline and perform a comparison with the model

### Learn more

- [Auditing](#) <sup>[446]</sup>
- [List of Permissions](#) <sup>[329]</sup>
- [Model Views](#) <sup>[686]</sup>
- [Monitor Change Events](#) <sup>[593]</sup>

#### 3.7.2.2.1 Baselines

Enterprise Architect provides a facility to 'Baseline' (snapshot) a model branch at a particular point in time for later comparison with the current package state.

Topic	Detail	See also
<b>Baselines</b>	<p>Baseline comparison is most useful for determining the changes made to the model during development compared to some Baseline saved at a crucial point - for example the completion of a phase or version iteration.</p> <p>More than one Baseline can be stored against a single Enterprise Architect package.</p> <p>Baselines are particularly useful during requirements management to check for changes, additions and deletions that have occurred since the start of the current work phase; knowing how a model has changed is an important part of managing change and the overall development process.</p> <p>Baselines are stored within the model in compressed XML format; you can save a Baseline to an external XML file for</p>	<a href="#">The Compare Utility (Diff)</a> <sup>[463]</sup>

Topic	Detail	See also
	<p>storage or archive, or for distributing to other users working on models derived from a master project.</p> <p>Baselines are generally used in conjunction with the <i>Compare</i> utility.</p>	
<b>Scenario</b>	<p>A typical scenario for using Baselines would be:</p> <ul style="list-style-type: none"> <li>• Create the base model branch to a sufficient point to create a Baseline (checkpoint); create and store the Baseline as Version 0.1a</li> <li>• As work continues on development, managers and developers can check the current model branch against the Baseline for important modifications, additions and deletions; the Compare utility can be invoked from the Baseline dialog to check the current model branch against the stored version</li> <li>• As required, minor Baselines can be created to check recent progress; these 'temporary Baselines' are useful for managing change when a lot of work is being done and it is important to only see what has changed in, for example, the last 24 hours</li> <li>• At sign-off or the move to a new version/phase, a major Baseline can be created to capture the new state of the model</li> </ul> <p>Minor Baselines created earlier can be deleted if required to save space</p>	<p><a href="#">Manage Baselines</a> <sup>461</sup></p> <p><a href="#">Create Baselines</a> <sup>462</sup></p>
<b>Considerations</b>	<ul style="list-style-type: none"> <li>• Baselines are based on the GUID or unique ID of a particular package: <ul style="list-style-type: none"> <li>• Enterprise Architect checks for that ID as the root element within the XML document being used as a Baseline</li> <li>• When you export a package to XML, the package you export is the root element; likewise when you create a Baseline, the current package is the root package of the XML Baseline</li> <li>• When you save information in a version control system, the current version-controlled package is again the root package of the document</li> </ul> </li> <li>• It is not useful to create a Baseline by importing an XML package file created by version controlling a package that itself contains version-controlled child packages; that type of XML package file contains stubs for the child packages, not full information on the child packages and elements</li> <li>• If a package under version control forms part of a Baseline, and that package is checked in to the model, you cannot merge the original data from the Baseline into that package</li> <li>• XML files must be in the same format used by the</li> </ul>	

Topic	Detail	See also
	Baseline engine - currently the UML 1.3 XMI 1.1 format (plus Enterprise Architect extensions), which contains all the information necessary to reconstruct a UML model, even a UML 2.x model	

### Notes

- The Baseline facility is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect
- The Compare utility is available in the *Professional* edition of Enterprise Architect, as well as in the Corporate and extended editions above

### 3.7.2.2.1.1 Manage Baselines



Enterprise Architect provides a range of facilities for working with and managing Baselines.

**Access** Right-click model branch package | Package Control | Package Baselines

### Reference

Create, select and process Baselines using the Package Baselines dialog, as indicated below:

Field	Usage	See also
<b>Current Baselines For Package: &lt;Name&gt;</b>	Review the Baselines for the current model branch, listed by version reference with the highest alphabetical/numerical value at the top.  If an entry is longer than the display area, a horizontal scroll bar displays at the bottom of the panel; use this to scroll to the text that is not shown.	<a href="#">Create Baselines</a> <sup>[462]</sup>
<b>Show Differences</b>	Run the Compare <sup>[463]</sup> utility on the selected Baseline and the current model branch or diagram, to display the differences <sup>[468]</sup> between the two.	<a href="#">The Compare Utility</a> <sup>[463]</sup> <a href="#">Check Visual Changes to Diagrams</a> <sup>[466]</sup> <a href="#">Example Comparison</a> <sup>[468]</sup>
<b>Restore to Baseline</b>	Completely restore the model branch from the selected Baseline.	
<b>New Baseline</b>	Create a new Baseline.	<a href="#">Create Baselines</a> <sup>[462]</sup>
<b>Delete Selected</b>	Delete the selected Baseline.	

Field	Usage	See also
<b>Load Other Baselines</b>	<p>Display a drop-down menu that enables you to load Baselines from another model, in either a project file or a DBMS repository.</p> <ul style="list-style-type: none"> <li>For project files, a browser displays; locate the required project file</li> <li>For DBMS repositories, the Windows Data Link Properties dialog displays; select the data provider and click on the <b>OK</b> button to display the Select Data Source dialog, from which you select the required project</li> </ul> <p>In either case, the <i>Connected To:</i> message at the bottom of the Package Baselines dialog changes to the name of the alternative model.</p> <p>To return the dialog to the original project, select the third option on the drop down list: <b>Load From Selected Package</b>.</p>	
<b>Import File</b>	Import an XML 1.1 file from the file system as a new Baseline for this current model branch.	
<b>Export File</b>	Export the selected Baseline to an XML file on disk.	
<b>Compare Model to File</b>	Compare the selected model branch with an XML 1.1 file on disk; a browser displays, which you use to locate the file.	
<b>Options</b>	Set filters  to make the comparison more specific.	<a href="#">Compare Options</a> 

#### Notes


- Package Baseline facilities are available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect

#### 3.7.2.2.1.2 Create Baselines

**Access** Project Browser package context menu | Package Control | Package Baselines: New Baseline

#### Reference

Create the new baseline using the New Baseline dialog, as indicated below:

Field	Usage	See also
<b>Name</b>	Display the package name of the currently selected model branch.	<a href="#">Manage Baselines</a> 

Field	Usage	See also
<b>Version</b>	Type a unique version reference for this Baseline, which can consist of any alphanumeric characters.  The Package Baselines dialog sorts the Baselines according to the value of this field.	
<b>Include Sub-packages</b>	Include the entire sub-package hierarchy of this branch in the Baseline; this option defaults to selected.  If you deselect the checkbox, only the immediate contents (XML stubs) of the package are included in the Baseline.	
<b>Note</b>	Edit the default current time and date to any other value.  The field is a single-line entry, for display on the Package Baselines dialog (a one-line-per-entry list).	
<b>OK</b>	Click to create a new Baseline and return to the Package Baselines dialog.	

### Notes

- Package Baseline facilities are available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect

#### **3.7.2.2.1.3 The Compare Utility (Diff)**

Enterprise Architect has a comprehensive and powerful built in *Compare (diff)* utility, which enables you to:

- Explore what has changed within a model over time
- Explore how previous versions of a model branch differ from what is currently in the model
- Perform a full model comparison by exporting all of Model A to XML, then using **Compare Model to File** from within the current model (Model B)

Comparing and checking model development at various points in the process is an important aspect of managing change and development, monitoring what is being modified and ensuring the development and design process is on track.

### Access

- Project Browser package context menu | Package Control | Package Baselines: Show Differences**  
- use the Baseline dialog
- Project Browser package context menu | Package Control | Compare with XMI File** (package not under version control)
- Project Browser package context menu | Package Control | Compare with Controlled Version** (package under version control)

**Use to**

- Compare a model branch in Enterprise Architect with a Baseline created using the Baseline functionality (Corporate, Business and Software Engineering, System Engineering and Ultimate editions)
- Compare a model branch in Enterprise Architect with a Baseline stored in a different model
- Compare a model branch in Enterprise Architect with an XML 1.1 file on disk created previously using the Enterprise Architect XML export facility (user selects file)
- Compare a model branch in Enterprise Architect with the current version-controlled XML 1.1 file on disk as created when using Version Control in Enterprise Architect (file automatically selected)

**Guide:**

Topic	Detail	See also
<b>Differencing With Baselines</b>	<p>As a Baseline is stored within a model and contains all the information, elements and connections for a package at a point in time, it can be used within Enterprise Architect to track changes to model elements over time.</p> <p>The Differencing engine first builds a representation of the current package in memory, based on what is currently in the model.</p> <p>It then compares this with the stored Baseline, highlighting changes, new elements, missing elements and elements that have been moved to other packages.</p> <p>It is possible to filter the resultant output to display only one particular kind of change: for example, additions to the model.</p> <p>If a Baseline has been created to ignore child package content, a comparison between that Baseline and the model does not include any child package content in the model.</p> <p>See the example model comparison.</p>	<p><a href="#">Compare Options</a> <sup>[464]</sup></p> <p><a href="#">Create Baselines</a> <sup>[462]</sup></p> <p><a href="#">Example Comparison</a> <sup>[468]</sup></p>

**Notes**

- This utility is available in the Professional, Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect
- You cannot compare the current model with an XML 2.1 file; the utility can only compare with an XML 1.1 file

**3.7.2.2.1.4 Compare Options**

The Compare Options dialog enables you to refine the output of the Compare utility when it compares the current model with a Baseline.

To display the dialog, either:

- Click on the **Options** button on the Package Baselines dialog
- Click on the **Compare Options** icon on the Compare Utility tab toolbar



Field	Usage	See also
<b>Always Expand to Differences</b>	<p>Always display the list of elements fully expanded to show changes.</p> <p>If you deselect the checkbox, when the Compare Utility tab is first opened it lists the package contents to element level, and you expand each element as required to show the changed items.</p> <p>For large branches of the model, it is better to leave the checkbox unselected.</p>	
<b>Show Elements that are</b>	<p>List elements that:</p> <ul style="list-style-type: none"> <li>• Have been changed since the Baseline was created</li> <li>• Are in the Baseline only (that is, have been deleted from the model since the Baseline was created)</li> <li>• Are in the model only (that is, have been created since the Baseline was created)</li> <li>• Have not changed since the Baseline was created (you might generally leave this checkbox unselected)</li> </ul>	
<b>Suppress these Changes</b>	<p>Exclude:</p> <ul style="list-style-type: none"> <li>• Changes to diagrams</li> <li>• Changes to the <b>Date Modified</b> field for an item</li> <li>• Changes to the <b>Date Created</b> field for an item</li> <li>• Child items of a deleted item</li> <li>• Changes to advanced properties (defaults to selected)</li> </ul>	
<b>Baseline Diagram Compare Options</b>	<p>Select the checkbox to always open the first parent package for which there is a Baseline, when you select the diagram for comparison from the Project Browser.</p>	

If the Compare Utility tab shows the results of a Baseline comparison, when you click on the **OK** button the display refreshes to refine the information according to the options you have selected.

### Notes

- Package Baseline facilities are available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect

### Learn more

- [Compare Utility Tab Options](#) <sup>[470]</sup>

### 3.7.2.2.1.5 Check Visual Changes to Diagrams

The Baseline Diagram Compare feature is a quick and easy way to **visually compare** a current diagram with an earlier version from a saved Baseline, and highlight any elements in the diagram that have been Added, Deleted, Resized or Moved.

You can then review these changes and optionally roll back each change if needed to their previous state from the Baseline.

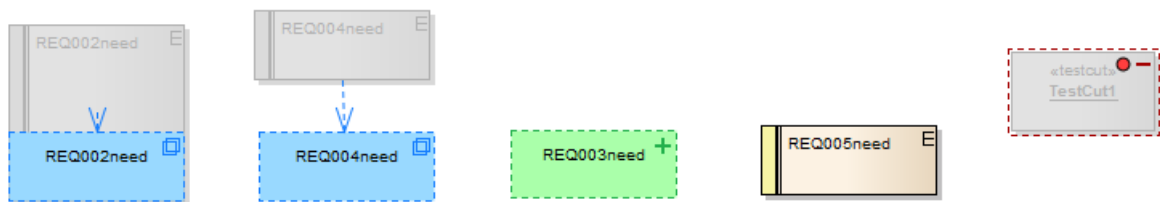
The changes are identified on the **Baseline Diagram Compare** dialog and on the diagram itself. If the diagram is not already open, the compare feature also opens the diagram.





**Access** Any of the following:

- **Project Browser diagram context menu | Compare Diagram to Baseline | <select baseline>: Show Differences**
- **Project Browser Package context menu | Package Control | Package Baselines | <select baseline>: Show Differences | Selected diagram context menu | Compare to Baseline**
- **Diagram context menu | Compare to Diagram Baseline | <select baseline>: Show Differences, or**
- **Select Package | Project | Package Baselines | <select baseline>: Show Differences | Selected diagram context menu | Compare to Baseline**

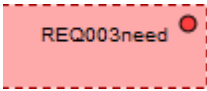
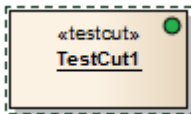
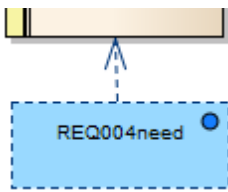
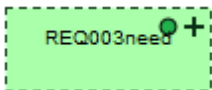
#### Guide


As you work on the dialog, the diagram changes as illustrated and as explained here.



	Name	Status	Action
<input checked="" type="checkbox"/>	 REQ003need	Baseline only	Restore Element & add to Diagram
<input checked="" type="checkbox"/>	 REQ004need	Changed	Restore Element position
<input checked="" type="checkbox"/>	 REQ002need	Changed	Restore Element position
<input checked="" type="checkbox"/>	 TestCut1	Model only	Delete from Diagram

Topic	Detail	See also
<b>Select element for review</b>	<p>The <b>Status</b> column indicates whether the element has been:</p> <ul style="list-style-type: none"> <li>• Moved or re-sized (<i>Changed</i>)</li> <li>• Deleted from the diagram (<i>Baseline only</i>), or</li> <li>• Added to the diagram since the Baseline was captured (<i>Model only</i>)</li> </ul> <p>Click on the item name.</p>	

Topic	Detail	See also
	<p>When an item is selected, the corresponding element on the diagram is highlighted as follows:</p> <ul style="list-style-type: none"> <li>Deleted from the diagram  </li> <li>Added to the diagram  </li> <li>Resized or moved to a new position  </li> </ul> <p>The currently-selected element on the diagram is marked with a dot, as shown, to indicate that it is in focus.</p>	
<b>Position the diagram to show the selected element</b>	<p>To scroll the diagram so that you can see the original (Baseline) position of an element, double-click on the item in the list.</p> <p>To scroll the diagram so that you can see the current (model) position of the element, press and hold (<b>Ctrl</b>) while you double-click on the item.</p>	
<b>Leave the changes in the item as they are</b>	<p>Ensure that the checkbox against the item is not selected.</p> <p>Click on the <b>OK</b> button.</p>	
<b>Roll the changes back to the Baseline position</b>	<p>Click on the checkbox against each required item (or click on the <b>Check All</b> button to select all items).</p> <p>The <b>Action</b> column displays the action required to roll each element's relationship to the diagram back to the Baseline relationship.</p> <p>The corresponding element on the diagram is highlighted as follows:</p> <ul style="list-style-type: none"> <li>Deleted, now being restored  </li> <li>Added, now being removed</li> </ul>	

Topic	Detail	See also
	 <ul style="list-style-type: none"> <li>Resized/repositioned, now being put back to original position</li> </ul> <p>The comparison shows the blue direction arrow for all repositions or resizes that have been checked. For a heavily edited diagram this can be confusing. However, you can hide the arrow for all elements except the one currently in focus; to do this:</p> <ul style="list-style-type: none"> <li>Deselect the <b>Visually link new and old position for checked items</b> checkbox</li> </ul> <p>To roll back the changes for all items for which a checkbox is selected:</p> <ul style="list-style-type: none"> <li>Click on the <b>Apply</b> button</li> </ul>	

### Notes

- Diagram Baseline facilities are available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect

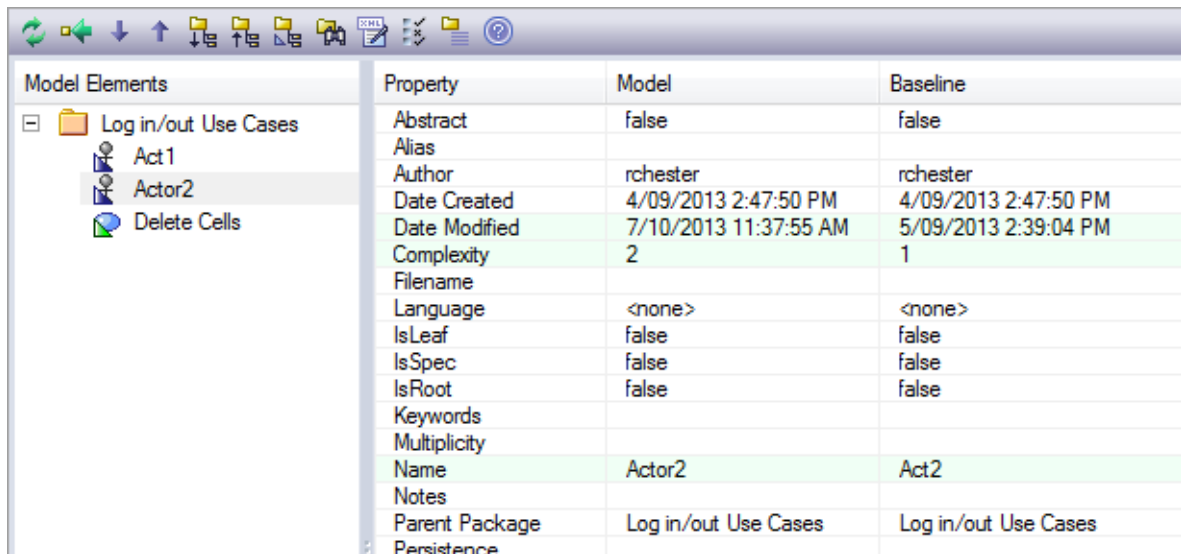
### Learn more

- [Baselines](#)<sup>[459]</sup>
- [Manage Baselines](#)<sup>[461]</sup>
- [Compare Options](#)<sup>[464]</sup>
- [Compare Utility Tab Options](#)<sup>[470]</sup>

#### 3.7.2.2.2 Example Comparison

The diagram below shows the result of a comparison between a package (Log in/out Use Cases) in the current project and that package in a Baseline captured at an earlier date.

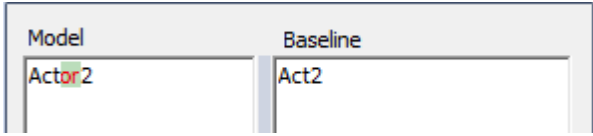
The results of the comparison are displayed on the Baseline Comparison tab.



Property	Model	Baseline
Abstract	false	false
Alias		
Author	rchester	rchester
Date Created	4/09/2013 2:47:50 PM	4/09/2013 2:47:50 PM
Date Modified	7/10/2013 11:37:55 AM	5/09/2013 2:39:04 PM
Complexity	2	1
Filename		
Language	<none>	<none>
IsLeaf	false	false
IsSpec	false	false
IsRoot	false	false
Keywords		
Multiplicity		
Name	Actor2	Act2
Notes		
Parent Package	Log in/out Use Cases	Log in/out Use Cases
Persistence		

### Review Changes

Topic	Detail	See also
<b>Interpretation</b>	<p>A hierarchy of model elements is displayed in the left-hand pane. It is clearly visible, from the triangle-based icons and the highlighted lines on the report, which items in the hierarchy have, since the Baseline was captured, been:</p> <ul style="list-style-type: none"> <li>• Changed</li> <li>• Deleted from the model (in the Baseline only)</li> <li>• Added to the model (in the Model only) or</li> <li>• Switched to different packages (changes in the <b>Parent Package</b> property)</li> </ul> <p>If you click on an item in the left hand pane, the right-hand pane displays a table of properties showing the values of those properties in the current model and in the Baseline.</p> <p>For each property where there is a difference between the model and the Baseline, the row is highlighted.</p> <p>The Compare Utility tab enables you to perform operations (such as merging or rolling back changes) on the reported information, using the toolbar, context menu and keyboard.</p>	<a href="#">Compare Utility Tab Options</a> <sup>[470]</sup>
<b>Increase Level of Detail</b>	<p>The right panel of the Compare Utility tab might, for some fields, display only part of the value.</p> <p>It might also not be immediately obvious what a change is.</p> <p>In either case, you can double-click on the property to display full details and to highlight the exact differences; the following example shows the highlighted changes to the <b>Name</b> property.</p>	

Topic	Detail	See also
		

### Notes

- The Compare utility is available in the Professional, Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect

#### 3.7.2.2.1 Baseline Comparison Tab Options

The Baseline Comparison tab enables you to perform operations on the reported information, using the toolbar, context menu, Merge dialog and certain keyboard keys.



- The toolbar is at the top of the left-hand panel; the icons operate either on the comparison as a whole or on the currently-selected item in the left hand panel of the Baseline Comparison tab
- Each item in the hierarchy has a context menu, which you display by right-clicking on the item; the options displayed depend on the level of the item in the hierarchy
- The Merge dialog enables you to specify which changes to roll back in the model from the baseline
- You can use a selection of keyboard keys to move up and down the hierarchy, or to roll back changes

### Toolbar Options

Option	Action	Shortcut	See also
<b>Refresh</b>	Re-run the comparison to refresh the current display.		
<b>Merge To Model</b>	Merge the values of the currently-selected item in the Baseline back into the model.		
<b>Next Change</b>	Highlight the next changed item (this skips <i>Moved</i> items).		
<b>Previous Change</b>	Highlight the previously-changed item.		
<b>Expand All</b>	Fully expand the selected item.		
<b>Collapse All</b>	Collapse the changed items in the selected item.		

Option	Action	Shortcut	See also
<b>Expand To Changed Items</b>	Expand the selected item to show changed items only (in the event that you have selected to also show unchanged items in the comparison).		<a href="#">Compare options</a> <sup>[464]</sup>
<b>Find in Project Browser</b>	Highlight the item in the Project Browser.		
<b>Log To XML</b>	Log the changes to an XML file. A browser displays, on which you specify the file name and location.		
<b>Compare Options</b>	Display the Compare Options dialog.		<a href="#">Compare Options</a> <sup>[464]</sup>
<b>Manage Package Baselines</b>	Display the Package Baselines dialog.		<a href="#">Manage Baselines</a> <sup>[464]</sup>
<b>Help</b>	Display the Help topic <i>Package Baselines</i> .		<a href="#">Package Baselines</a> <sup>[457]</sup>

#### Context Menu Options

Option	Action	Shortcut	See also
<b>Merge from Baseline</b> <b>Add from Baseline</b>	Restore the item in the model to the Baseline state, or restore a deleted item from the Baseline.		
<b>Delete from Model</b>	Remove a recently-created item from the model.		
<b>Merge From Baseline (with Options)</b>	(For the root node of the hierarchy on the Compare Utility tab.) Display the Merge dialog (see below) which enables you to specify options for rolling back the whole model branch to the Baseline state.		
<b>Refresh</b>	(Object-level items.) Re-run the comparison to refresh the current display.		
<b>Find in Project Browser</b>	Locate and highlight the item in the Project Browser.		
<b>Open Baseline</b>	(For a diagram listed in the comparison.)		<a href="#">Check Visual</a>

Option	Action	Shortcut	See also
<b>Diagram Compare</b>	Displays the Baseline Diagram Compare dialog, showing differences in diagram content and layout.		<a href="#">Changes to Diagrams</a> <sup>[466]</sup>
<b>Expand All</b>	Fully expand the selected item.		
<b>Expand To Changed Items</b>	Expand the selected item to show changed items only.		
<b>Collapse All</b>	Collapse the changed items in the selected item.		
<b>Log To XML</b>	Log the changes to an XML file. A browser displays, on which you specify the file name and location.		
<b>Compare Options</b>	Display the Compare Options dialog.		<a href="#">Compare Options</a> <sup>[464]</sup>

#### Merge Dialog Options

Field/Button/Option	Action	Shortcut	See also
<b>Changed</b>	Restore all changed items in the model branch to the Baseline state.		
<b>In Baseline Only</b>	Restore all deleted items to the model branch from the Baseline.		
<b>In Model Only</b>	Remove all recently-created items from the model branch.		
<b>Moved</b>	Restore all moved items to their original locations, as identified in the Baseline.		
<b>Full Restore from XML</b>	Completely restore the model branch to the version held in the Baseline XML 1.1 file, (using the XML Import function).  (This option automatically selects all the other options.)		

#### Keyboard Keys



- ( **Ctrl+↓** ) - expand and highlight the next changed item
- ( **Ctrl+↑** ) - expand and highlight the previous changed item
- ( **Ctrl+←** ) - undo the changes for a selected item (roll back to the Baseline values)

### 3.7.3 Model Transfer

Enterprise Architect enables you to transfer data between projects, using three different mechanisms.

#### Topics

Topic	Detail	See also
<b>XMI Import and Export</b>	<p>Packages can be exported from and imported into Enterprise Architect models.</p> <p>This greatly improves the flexibility and robustness of Enterprise Architect models by enabling Analysts and Modelers to externalize model elements in XMI for version control, distributed development, post processing and transferring packages between models.</p>	<a href="#">XMI Import and Export</a> <sup>[473]</sup>
<b>CSV Import and Export</b>	<p>You can import and export information about Enterprise Architect elements in CSV format.</p> <p>This enables you to store and retrieve information from spreadsheet tools such as Microsoft Excel.</p>	<a href="#">CSV Import and Export</a> <sup>[497]</sup>
<b>Data Transfer</b>	Enterprise Architect enables you to move complete models between files, and between files and repositories, row by row, table by table.	<a href="#">Perform a Project Data Transfer</a> <sup>[504]</sup>

#### 3.7.3.1 XMI Import and Export

XML Metadata Interchange (XMI), defined by the OMG and based on XML, is an open standard file format that enables the interchange of model information between models and tools.

#### Topics

Topic	Detail	See also
<b>XMI and Enterprise Architect</b>	<p>Enterprise Architect uses XMI as a method of importing and exporting model specifications between different UML packages, Enterprise Architect projects and other tools that support XMI.</p> <p>Enterprise Architect supports the XMI 1.1, 1.2, 2.1 and 2.4.1 specifications. XMI 1.1 has support for UML 1.3, whereas XMI 2.1 is used for UML 2.0 - 2.3.</p> <p>Enterprise Architect does not fully support the older 1.0 specification. When importing or exporting to XMI 1.0, some loss of data occurs due to the limitations of XMI 1.0.</p>	

Topic	Detail	See also
	<p>With XML, model details can be exchanged between different UML tools and other tools that are capable of using XML.</p> <p>Limited support for export to Rational Rose is provided using the Rose version of the XML 1.1 specification, as implemented by Unisys for Rational products.</p> <p>Packages can be exported from and imported into Enterprise Architect models, which greatly improves the flexibility and robustness of the models by enabling Analysts and Modelers to externalize model elements in XML for:</p> <ul style="list-style-type: none"> <li>• Version control</li> <li>• Distributed development</li> <li>• Post processing and</li> <li>• Transferring packages between models</li> </ul> <p>When performing Enterprise Architect-to-Enterprise Architect transfers, ensure that either XML version 1.1 or 2.1 is selected.</p> <p>When importing an XML file over an existing package, all information in the current package is deleted first, before importing from the XML file. Please make sure you do not have important changes that you do not want to lose before importing from XML.</p>	
<b>XMI Tasks</b>	<p>Tasks you might perform in importing and exporting XMI include:</p> <ul style="list-style-type: none"> <li>• Setting XML Options - XMI import, export and package control all rely on saving and loading XML files; you can set a number of options to streamline this process</li> <li>• Exporting a package to XMI in XMI 2.1 (and earlier)</li> <li>• Importing from XMI with support for XMI 2.1 (and earlier)</li> <li>• Setting up controlled packages</li> <li>• Manually controlling a package by linking it to an XMI file</li> <li>• Batch exporting controlled packages</li> <li>• Batch importing controlled packages</li> <li>• Factoring in the limitations of XMI</li> <li>• Applying a UML Data Type Definition (DTD)</li> <li>• Importing a model from IBM Rational Rhapsody</li> </ul>	<p><a href="#">XML Specifications</a> <sup>[639]</sup></p> <p><a href="#">Export to XMI</a> <sup>[475]</sup></p> <p><a href="#">Publish Model Package</a> <sup>[476]</sup></p> <p><a href="#">Import from XMI</a> <sup>[478]</sup></p> <p><a href="#">Controlled Packages</a> <sup>[484]</sup></p> <p><a href="#">Manual Version Control with XMI</a> <sup>[494]</sup></p> <p><a href="#">Batch XMI Export</a> <sup>[492]</sup></p> <p><a href="#">Batch XMI Import</a> <sup>[493]</sup></p> <p><a href="#">Limitations of XMI</a> <sup>[482]</sup></p> <p><a href="#">The UML DTD</a> <sup>[483]</sup></p> <p><a href="#">Import a Rhapsody Model</a> <sup>[482]</sup></p>

**Notes**

- XML 2.1 exported by Enterprise Architect 7.0 (or later) might not be correctly imported into earlier versions of Enterprise Architect
- When performing Enterprise Architect-to-Enterprise Architect transfers, ensure that either XML version 1.1 or 2.1 is selected

#### Learn more

- [XML/XML Technology](#)

### 3.7.3.1.1 Export to XMI

The Export Package to XMI dialog is used to export a package to various XMI (XML-based) formats. This dialog, by default, allows exporting to either **XMI 1.1** or **XMI 2.1**. The exported XMI file contains all the information required for completely round-tripping the package within Enterprise Architect; that is, performing Enterprise Architect-to-Enterprise Architect transfers.

**Access** Click on a package in Project Browser, then:

**Project | Model Import/Export | Export Package to XMI (Ctrl+Alt+E)**  
**Right-click | Import/Export | Export Package to XMI file**

#### Use to

- Move Enterprise Architect model elements between models, for distributed development, manual version control and other benefits

#### Reference

Field/Option/Button	Usage	See also								
Root Package	Indicate the name of the selected package.									
Filename	Specify the file path where the XMI file is to be generated.									
Export Type	<table><tr><th>Option/Button</th><th>Usage</th></tr><tr><td>XMI 1.1</td><td>Export the selected package to XMI 1.1.</td></tr><tr><td>XMI 2.1</td><td>Export the selected package to XMI 2.1.</td></tr><tr><td>Publish</td><td>Open the Publish Model Package dialog for exporting to other versions of XML.</td></tr></table>	Option/Button	Usage	XMI 1.1	Export the selected package to XMI 1.1.	XMI 2.1	Export the selected package to XMI 2.1.	Publish	Open the Publish Model Package dialog for exporting to other versions of XML.	<a href="#">Publish Model Package</a> <sup>[476]</sup>
Option/Button	Usage									
XMI 1.1	Export the selected package to XMI 1.1.									
XMI 2.1	Export the selected package to XMI 2.1.									
Publish	Open the Publish Model Package dialog for exporting to other versions of XML.									
View XMI	View the exported XMI file.									
Export	Initiate an XMI Export.									

<b>Close</b>	Close this dialog.	
<b>Help</b>	Display this Help topic.	
<b>Progress</b>	List the progress of the XMI Export.	

### Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Export XMI** permission to export to XMI
- Alternate Images used in the package being exported are not included in the exported XMI file unless the **Export Alternate Images** option is selected in the XML Specifications section of the Options dialog
- The Log file for the package being exported is not generated unless the **Write Log** option is selected in the XML Specifications section of the Options dialog
- XMI 2.1 exported by Enterprise Architect 7.0 (or later) might not be correctly imported into earlier versions of Enterprise Architect

### Learn more

- [Limitations of XMI](#)<sup>[482]</sup>
- [List of Available Permissions](#)<sup>[329]</sup>
- [Manual Version Control with XMI](#)<sup>[494]</sup>
- [XML Specifications](#)<sup>[639]</sup>

#### 3.7.3.1.2 Publish Model Package

The Publish Model Package facility is used to export a package to various XML formats, including XMI 1.1, XMI 2.1, ArcGIS and BPMN 2.0 XML; for example you might:



- Export BPMN 2.0 models to BPMN 2.0 XML, or
- Export Enterprise Architect model elements to Rational Rose and other tools that implement the:
  - UML 1.3 XMI 1.1 / XMI 1.0 standard
  - UML1.4 XMI 1.2 standard, or
  - UML 2.x XMI 2.x standard

**Access** Click on a package in Project Browser, then select one of the following:

**Project | Publish Model**  
**Project | Model Import/Export | Export Package to XMI: Publish** or **(Ctrl+Alt+E): Publish**  
**Extensions | Publish | <XML Format Type>**, or  
**Right-click | Import/Export | Export Package to XMI File: Publish**

### Publish Model Package options

Field/Option/ Button	Usage	See also
-------------------------	-------	----------

<b>Root Package</b>	Displays the name of the selected package.	
<b>Filename</b>	Type in or browse for the file path into which the XML file is to be generated.	
<b>XML Type</b>	<p>Select the XML/XMI version to which to export the package.</p> <ul style="list-style-type: none"> <li>• Select <b>UML 2.1</b> if you want to use XMI 2.1 in round-tripping a model</li> <li>• Select <b>UML 2.0, 2.1.1, 2.1.2, 2.2, 2.3</b> or <b>2.4.1</b> if exporting to a tool that requires that specific version of the XMI</li> </ul> <p>The selected version might render some of the following eight options unavailable.</p>	
<b>Export Diagrams</b>	Export all the diagrams in the selected package.	
<b>Export Alternate Images</b>	Export the alternate images used in the diagrams in the selected package.	<a href="#">Image Manager</a> 
<b>Format XML Output</b>	Format the output into readable XML (this takes a few more seconds at the end of the run).	
<b>Write Log File</b>	<p>Write a log of the export activity (recommended).</p> <p>The log file is saved to the directory into which the XML file is exported.</p>	
<b>Use DTD</b>	<p>Use the XMI DTD when exporting to XMI 1.0 or XMI 1.1.</p> <p>Setting this option validates the correctness of the model and checks that no syntactical errors have occurred.</p>	
<b>Exclude EA Extensions</b>	Exclude tool-specific information from the export.	
<b>Unisys/Rose Format</b>	Export in Unisys/Rose compatible XMI format.	
<b>Generate Diagram Images</b>	<p>Generate the exported diagrams in the format you select from the <b>Format</b> drop-down list.</p> <p>The generated diagrams are saved in a package called <i>Images</i> in the directory into which the XML file is exported.</p>	
<b>Stylesheet</b>	Select an XSL Stylesheet to post-process the XML content before saving to file ( <i>optional</i> ).	<a href="#">Resource Window</a> 

	You can post-process the XML content before saving the package to file, using an XSLT to convert the output to HTML, XSL, code or other versions of XML; to do this, you must import the required style sheet into the project through the Resources window.	
<b>View XML</b>	View the exported XML file.	
<b>Export</b>	Initiate XML export.	
<b>Close</b>	Close this dialog.	
<b>Help</b>	Display this Help topic.	
<b>Progress</b>	Indicate the progress of the XML export.	

#### Notes

- When exporting and importing with XML 1.0 with Enterprise Architect, some loss of data occurs due to the limitations of XML 1.0
- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Export XML** permission to export to XML
- When you select to apply a Data Type Definition (DTD) during an XML 1.1 export, the UML\_EA.DTD file is written to the output directory into which the XML files are written (unless the UML\_EA.DTD file is already present in the directory); no error is generated if the UML\_EA.DTD file is not present in this directory during the XML export

#### Learn more

- [Limitations of XML](#)<sup>[482]</sup>
- [List of Available Permissions](#)<sup>[329]</sup>
- [The Resources Window](#)<sup>[1173]</sup>
- [Manual Version Control with XML](#)<sup>[494]</sup>
- [Export to BPMN 2.0 XML](#)<sup>[1869]</sup>
- [Export to ArcGIS XML](#)<sup>[1961]</sup>

#### 3.7.3.1.3 Import from XML

You can import a package from an XML (XML-based) file in any of the following formats:

- UML 1.3 (XML 1.0)
- UML 1.3 (XML 1.1)
- UML 1.4 (XML 1.2)
- UML 2.x (XML 2.1)
- MOF 1.3 (XML 1.1)
- MOF 1.4 (XML 1.2)

You can also import the \*.emx and \*.uml2 files generated by tools such as Rational Software Architect (RSA) and Rational Software Modeler (RSM) as well as XML files exported by ArcGIS.

**Access** Click on target package in Project Browser, then:

**Extensions | Import | <XML Format>  
Project | Model Import/Export | Import Package From XMI File** or  
**Right-click | Import/Export | Import Package From XMI**

#### Use to

- Move Enterprise Architect Model elements between models, for distributed development, manual version control and other benefits

#### Import a Package from an XMI file

Field/Option/ Button	Action	See also
<b>Filename</b>	Type the directory path and filename from which to import the XMI file.	
<b>Import diagrams</b>	Select the checkbox to import diagrams.	
<b>Strip GUIDs</b>	Select the checkbox to remove Universal Identifier information from the file on import.  This enables the import of a package twice into the same model; the second import requires new GUIDs to avoid element collisions.	
<b>Write log file</b>	Select the checkbox to write a log of import activity (recommended).  The log file is saved in the directory from which the file is being imported, with the same name as that of the imported file plus the suffix <i>_import.log</i> .	
<b>Import using single transaction</b>	This option defaults to the setting of the <b>Import using single transaction</b> checkbox in the XML Specifications page of the Options dialog. If you wish, you can change the option from the default setting.  If the checkbox is selected, the XMI file is imported in one transaction. This is not recommended for large imports.  If the checkbox is not selected the data items are imported separately; you can identify problem items without the whole import being blocked. If the import is likely to encounter locking issues, or if you are importing a large XMI file, deselect the checkbox.	<a href="#">XML Specifications</a> [639]
<b>Treat Imported Datatypes</b>	If you are importing from Rose XMI 1.1, click on the drop-down arrow and select the datatypes to add to the model.	

Field/Option/ Button	Action	See also
<b>Import</b>	Click to start the import.	
<b>Close</b>	Click to close the dialog.	
<b>Help</b>	Click to display this Help page.	
<b>Import Progress</b>	Indicates the progress of the import.	

### Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Import XMI** permission to import packages from XMI
- When you import an XML file over an existing package, ALL information in the current package is deleted first; before you import the XML file, please make sure you do not have important changes that you do not want to lose
- If you are importing an XMI 1.1 file that was previously exported with a UML\_EA.DTD file, the UML\_EA.DTD file must be present in the directory into which the XML file is being written; an error occurs if the UML\_EA.DTD file is absent

### Learn more

- [List of Available Permissions](#) <sup>[329]</sup>
- [The Resources Window](#) <sup>[1173]</sup>
- [Manual Version Control with XMI](#) <sup>[494]</sup>
- [The UML DTD](#) <sup>[483]</sup>
- [Import EMX/UML2 Files](#) <sup>[480]</sup>
- [Import ArcGIS File](#) <sup>[1968]</sup>

#### 3.7.3.1.4 Import EMX/UML2 Files

Rational Software Architect (RSA) enables you to add many UML models under a single root. These models can have cross references between them.

However, RSA cannot save the entire root as one file; it saves each UML model as a separate EMX file. This means that an EMX file with cross-references is not self-contained as it references elements in another EMX file.

In releases earlier than release 7.0, Enterprise Architect treats each EMX file as a separate model and hence does not allow for cross-references between them. From release 7.0, Enterprise Architect enables these cross-references.

You therefore have the options of:

- Importing a single EMX/UML2 file or



- Importing a group of related EMX/UML2 files together, thereby retaining the cross-references between the different files

**Access** Click on target package in Project Browser, then:

**Extensions | Import | EMX/UML2**  
**Project | Model Import/Export | Import Package From XMI File: Other XML File Formats | EMX/UML2**, or  
**Right-click | Import/Export | Import Package From XM File: Other XML File Formats | EMX/UML2**

**Use to**

- Import Enterprise Architect Model elements from \*.emx/\*.uml2 files

**Reference**

On the Import Package from EMX/UML2 dialog:

Field/Option/Button	Action	See also
<b>Directory</b>	Click on the ( ... ) (Browse) button.  The Select Import EMX / UML2 File(s) dialog displays, which enables you to select multiple files.  Select the file or files (use ( <b>Ctrl</b> ) + <b>click</b> or ( <b>Shift</b> ) + <b>click</b> to select several files) and click on the <b>Open</b> button to return to the Import Package from EMX/UML2 dialog.	
<b>Selected Files</b>	Lists the selected *.emx/*.uml2 files.	
<b>Write Log File</b>	Select to write a log of import activity (recommended).  The log file is saved in the directory from which the files are being imported, with the name <i>import.log</i> .	
<b>Import</b>	Click to import the *.emx/*.uml2 files.	
<b>Close</b>	Click to close this dialog.	
<b>Help</b>	Click to display this Help topic.	
<b>Import Progress</b>	Indicate the progress of the import.	

### 3.7.3.1.5 Import a Rhapsody Model

If you have developed a UML model in **IBM Rational Rhapsody**, you can import it into an Enterprise Architect project.

Each Rhapsody project contains one Project file (with the file extension *.rpy*), which is associated with a folder with the same name, ending in the prefix *\_rpy*. For example, if the Rhapsody project file is called *CDPlayer.rpy*, it is associated with a folder called *CDPlayer\_rpy*.

The associated folder contains all the project-related files. It could also contain a file named *filesTable.dat*, which lists all the files related to the Rhapsody project.

**Access** [Project Browser | Click on package | Extensions | Import | Rhapsody](#), or [Project Browser | Right-click on package | Import/Export | Import Package From XMI File: Other XML File Formats | Rhapsody](#)

#### Import UML Model from Rhapsody Project Files

Field/Option/ Button	Action	See also
<b>Rhapsody Project</b>	Click on the (...) (Browse) button.  The Select Import Rhapsody Project dialog displays.  Select a Rhapsody Project file, then click the <b>Open</b> button to return to the Import Model From Rhapsody dialog.	
<b>Rhapsody Files</b>	Read the name of the file located in the associated project folder.	
<b>Import</b>	Click on this button to import the Rhapsody model into Enterprise Architect.	
<b>Close</b>	Click on this button to close this dialog.	
<b>Help</b>	Click on this button to display this Help topic.	
<b>Import Progress</b>	Indicates the progress of the import.	

### 3.7.3.1.6 Limitations of XMI

Whilst XMI is a valuable means of specifying a UML model in a common format, it is relatively limited in the amount of additional information it can tolerate using the standard syntax.

#### Topics

Topic	Detail	See also
<b>Notes on Enterprise Architect</b>	<p>A lot of information from an Enterprise Architect model must be converted to Tagged Values, which import into other modeling systems as additional information or are ignored completely.</p> <p>Enterprise Architect can both generate and read:</p> <ul style="list-style-type: none"> <li>• XMI 1.0 and 1.1 using UML 1.3 format</li> <li>• XMI 1.2 using UML 1.4 format, and</li> <li>• XMI 2.1 using UML 2.x format</li> </ul> <p>Round-tripping model elements using XMI in Enterprise Architect is possible using XMI 1.1 or XMI 2.1; XMI 1.1 uses the additional Tagged Values to store the UML 2.x information - this ensures model completeness when round-tripping UML 2.x model elements using XMI 1.1.</p> <p>Please note that Enterprise Architect exclusively uses XMI 1.1 for round-tripping models using Version Control.</p>	
<b>Notes for Exporting to Rose and Other Tools</b>	<p>There are discrepancies in the Unisys/Rose implementation with regard to spelling mistakes and slightly different syntax to the official XMI 1.1 specification, so problems might occur.</p> <p>The way packages are arranged in different models can impact successful import into other systems; experimentation is the only work around for this problem.</p> <p>Some parts of the XMI import/export process do not work as expected in products like Rational Rose; for example, Note Links are not supported, and State Operations import but do not appear in diagrams.</p> <p>Rational Rose only supports import of a full project, not a single package.</p> <p>For best results, it is recommended that you keep the model elements to export to Rose simple and conforming as closely as possible to the UML 1.3 specification.</p>	

### 3.7.3.1.7 The UML DTD

When you import or export Enterprise Architect packages to XML, the import or export process can be validated using a Data Type Definition (DTD).

#### Guide:

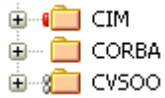
Topic	Detail	See also
<b>Notes on DTDs</b>	<p>The XML parser uses the DTD to validate the correctness of the model and to check that no syntactical errors have occurred.</p> <p>It is always best to use a DTD when moving packages between Enterprise Architect models as it ensures correctness of the XML output, and prevents attempted imports of incorrect XML.</p>	

Topic	Detail	See also
	<p>Several DTDs for XML/UML exist; the OMG defines a standard UML1.3 DTD for use in XML 1.1, and Enterprise Architect uses an extension of this with some additional element extensions for non-standard UML types, such as testing details.</p> <p>Whenever you read an XML file, the XML parser looks in the current directory for the DTD - if specified - using the DOCTYPE element in the XML file.</p> <p>If the parser cannot find the DTD, it records an error and aborts processing; you must ensure the UML_EA.DTD file is in the current XML output path (generated by default).</p>	

### 3.7.3.1.8 Controlled Packages

Controlled packages are a powerful means of 'externalizing' parts of an Enterprise Architect model.

#### Guide:

Topic	Detail	See also
<b>Using Controlled Packages</b>	<p>A controlled package is a package configured to save and load in XML format to a named file; the XML output is UML1.3 compliant XML, with Enterprise Architect extensions to support diagrams and additional model elements.</p> <p>Package XML is standard XML-compliant output that can be loaded into any XML viewer, or used by any XML-based tool to perform manipulations and extracts, such as document or code generators.</p> <p>Controlled packages appear in the Project Browser with a small colored rectangle to the left of the package icon, as shown below for the <i>CIM</i> package:</p>  <p>Using controlled packages you can support:</p> <ul style="list-style-type: none"> <li>• Widely distributed development by having team members submit packages in the form of XML for import into a central Enterprise Architect repository</li> <li>• Version control, by writing model elements in XML text files suitable for version control using standard version control software</li> </ul> <p>Using XML this way enables you to manually connect to third-party version control software outside the Enterprise Architect environment</p> <p>Enterprise Architect internally supports the configuration of</p>	<a href="#">Version Control</a> <sup>[383]</sup>

Topic	Detail	See also
	<p>version control through SCC and CVS</p> <ul style="list-style-type: none"> <li>Import and export of model elements between different models</li> </ul> <p>For example, a Class library can be re-used in many models and kept up to date in target models using controlled packages, reloading packages as required when new versions of the Class model become available</p>	

### Notes

- If you are importing an XML 1.1 file that was previously exported with a UML\_EA.DTD (Data Type Definition) file, the UML\_EA.DTD file must be present in the directory into which the XML file is being written; an error occurs if the UML\_EA.DTD file is absent
- When you select to apply a DTD during an XML 1.1 export, the UML\_EA.DTD file is written to the output directory into which the XML files are written (unless the UML\_EA.DTD file is already present in the directory); no error is generated if the UML\_EA.DTD file is not present in this directory during the XML export

#### 3.7.3.1.8.1 Controlled Package Menu

##### Access **Project Browser package context menu | Package Control**

This menu is for a Package that is not itself under version control (but that might contain child Packages that are under version control).

Option	Action	Shortcut	See also
<b>Lock Package</b>	<p>Select this option to apply a lock on the selected Package.</p> <p>(This option is only displayed when user security is enabled in User/Group locking mode.)</p>		<a href="#">Lock Package</a> <sup>[33]</sup>
<b>Apply/Release User Lock</b>	<p>Select this option to apply or release a user lock on the selected Package.</p> <p>(This option is only displayed when user security is enabled in Require User Lock to Edit locking mode.)</p>		<a href="#">Apply a User Lock</a> <sup>[33]</sup>
<b>Configure (various settings for the Package)</b>	Select this option to display the Package Control Options dialog, and specify whether this Package (and its child Packages) is controlled and which file it is controlled through.	<b>Ctrl+Alt+P</b>	<a href="#">Configure Packages</a> <sup>[48]</sup>
<b>Package Baselines</b>	Select this option to create a Baseline of the current Package, or compare the current Package with a previous Baseline.	<b>Ctrl+Alt+B</b>	<a href="#">Managing Baselines</a> <sup>[46]</sup>

Option	Action	Shortcut	See also
<b>Check In Branch</b>	<p>Select this option for the selected branch of the model (that is, the selected Package and all of its child Packages), to display the Select Packages to Check In dialog, listing all version controlled Packages within that branch that are checked out to you.</p> <p>You can then select Packages in the displayed list, to be submitted for check-in.</p>		<a href="#">Check in a Model Branch</a> <sup>[434]</sup>
<b>Check Out Branch</b>	<p>Select this option for the selected Package, to check out the Package and recursively check out all of its contained sub-Packages.</p> <p>The system retrieves the latest version of the Packages from the central repository, overwriting the current Packages.</p> <p>After check out, the Packages are available for editing.</p>		<a href="#">Check Out a Model Branch</a> <sup>[433]</sup>
<b>Save Package to File</b>	Select this option to save a controlled Package to an XML file.	<b>Ctrl+Alt+S</b>	<a href="#">Save a Package</a> <sup>[491]</sup>
<b>Load Package from File</b>	Select this option to load a previously-saved XML file.	<b>Ctrl+Alt+L</b>	<a href="#">Load a Package</a> <sup>[492]</sup>
<b>View Package XMI</b>	Select this option to display the Package XMI after the Package has been exported to XMI.	<b>Ctrl+Alt+X</b>	
<b>Compare Package with XMI file</b>	(Package not under version control) Select this option to compare the current Package with a previously-saved XMI file of the Package.		<a href="#">The Compare Utility (Diff)</a> <sup>[463]</sup>
<b>Add Branch to Version Control</b>	Select this option to apply version control to all Packages within the selected model branch, in a single operation.		<a href="#">Apply Version Control to Branches</a> <sup>[426]</sup>
<b>Export as Model Branch</b>	Select this option to export a newly created model branch from your own private copy of a model.		<a href="#">Export Controlled Model Branch</a> <sup>[437]</sup>
<b>Import a Model Branch</b>	Select this option to retrieve a model branch and import it into either the source model or another model.		<a href="#">Import Controlled Model Branch</a> <sup>[438]</sup>

Option	Action	Shortcut	See also
<b>Get Package (for version control)</b>	Select this option to gain access from Packages in the version-controlled repository that is currently available in your model.		<a href="#">Version Control</a> <sup>[383]</sup>
<b>Get All Latest (for version control)</b>	<p>Select this option to retrieve the latest version of the Package from the repository. The option is available only for Packages that are checked in.</p> <p>The alternative option <b>Get Latest</b> - if displayed - is not intended for sharing project files and should only be used when users have their own individual databases.</p>		<a href="#">Version Control</a> <sup>[383]</sup>
<b>Scan XMI and Reconcile Model</b>	<p>Select this option to scan the Package XMI files associated with each of the project's controlled Packages and restore any diagram objects or cross-references that are detected as missing from the project.</p> <p>This function is useful in team environments where each user maintains their own private copy of the model database (i.e. multiple private project files) and model updates are propagated through the use of controlled Packages. It provides no benefit when the model is hosted in a single shared database that is accessed by all team members.</p> <p>Each controlled Package is compared with its associated XMI file and, if the cross-reference information in the model does not match the XMI, the system updates the model with the information from the XMI and records the update in the Output window.</p> <p>You can roll back such updates by right-clicking on the entry in the Output window and selecting the context menu option <b>Rollback Update</b> (or <b>Rollback Selected Updates</b> if multiple entries are selected).</p> <p>Closing the model clears the entries in the Output window; an entry in the Output window is also cleared as and when you roll-back the update for it.</p> <p>This functionality is invoked automatically as part of the <b>Get All Latest</b> operation.</p> <p>When working in an environment that uses a Private Model deployment and your model contains a significant number of cross-Package references, it is recommended that you invoke <b>Scan XMI and Reconcile Model</b> from time to time, following the re-importation of controlled Packages - for example, after using <b>Get Latest</b> to update a number of Packages - or after performing a number of Package check-outs.</p> <ul style="list-style-type: none"> <li>As a general rule, avoid running <b>Scan XMI and Reconcile Model</b> while you have</li> </ul>		<p><i>Version Control Best Practices</i> white paper</p> <p><a href="http://sparxsystems.com/WhitePapers/Version_Control.pdf">http://sparxsystems.com/WhitePapers/Version_Control.pdf</a></p>

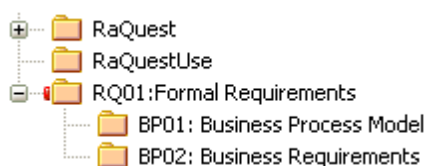
Option	Action	Shortcut	See also
	uncommitted changes in your model <ul style="list-style-type: none"> <li>Generally, you should:               <ul style="list-style-type: none"> <li>Check-out a number of Packages</li> <li>Invoke <b>Scan XML and Reconcile Model</b></li> <li>Make your modifications</li> <li>Commit any outstanding changes before you check-out more Packages and run <b>Scan XML and Reconcile Model</b> again</li> </ul> </li> </ul>		
<b>Re-synch Status With VC Provider</b>	(Version controlled Package.) Select this option to update the version control status value recorded for the selected Package in the Enterprise Architect project to match the value reported by the version control provider, without performing an XML import or export.		
<b>Version Control Settings</b>	Select this option to display the Version Control Settings dialog.		<a href="#">Version Control Settings</a> <sup>[415]</sup>

#### Learn more

- For a Package that is directly under version control, see [Package Version Control Menu](#) <sup>[427]</sup>

#### 3.7.3.1.8.2 Configure Packages



In order to configure a Package to save and load in XML format to a named file, you use the Package Control Options dialog to flag the Package as being controlled and set a number of options. These options include setting the XML filename to save to/load from, the type of XML to import and export, and whether to export and/or import the XML in a batch process. When the Package is configured, this is indicated by a small colored rectangle on the left of the Package icon in the Project Browser, as shown for the *RQ01: Formal Requirements* Package.



**Access** **Right-click Package in Project Browser | Package Control | Configure** (Ctrl+Alt+P)

#### Configure a controlled Package



Field/Button	Description	See also
<b>Control Package</b>	Select the checkbox to indicate that this is a controlled Package.	
<b>For all packages, create placeholders for external references</b>	<p>Select to force all XML 1.1 imports of this Package to exclude incoming relationships and instead create external references.</p> <p>This overrides the <b>Create placeholders for missing External References during XMI 1.1/2.1 Import</b> checkbox in the XML Specifications options for a user, if that checkbox is <b>not</b> selected.</p>	<a href="#">XML Specifications</a> 
<b>Version Control</b>	<p>Leave this field set to <b>(None)</b>, to manage the Package under XML control.</p> <p>The Package Control Options dialog is also used in adding a Package to Version Control and, in this context, is described under the <i>Version Control</i> section.</p>	<a href="#">Configure Controlled Package</a> 
<b>XMI Filename</b>	<p>Type or browse for the path and XML file name into which to export the Package data and from which to import it.</p> <p>You can also enter <b>Local Path Substitution strings</b>; for example, an XML local path definition where:</p> <pre>myLocal Path=" C: \ Documents and Settings\ John\ Desk top\ EA Model s"</pre> <p>So that %myLocal Pat h% CI M. x ml is equivalent to:</p> <pre>C: \ Documents and Settings\ John\ Desk top\ EA Model s\ CI M. x ml</pre>	
<b>UML/XMI Type</b>	<p>Click on the drop-down arrow and select the type of XML to be generated; options include:</p> <ul style="list-style-type: none"> <li>• <b>Enterprise Architect XMI/UML 1.3</b></li> <li>• <b>Rational Rose/Unisys UML 1.3 and</b></li> <li>• <b>Generic XMI 1.0/UML 1.3</b></li> <li>• <b>Enterprise Architect XMI 2.1/UML 2.1</b></li> </ul> <p>For complete import/export round tripping of Packages, currently only <b>Enterprise Architect XMI/UML 1.3</b> is supported.</p>	
<b>Version ID</b>	<p>Type the version number you want to apply to the Package.</p> <p>This field is the same as the <b>Version</b> field on the Package Properties dialog; any change to either field updates the other.</p>	
<b>Owner</b>	Defaults to the name of the Package owner; if necessary, type or select a different name.	

Field/Button	Description	See also
<b>Use DTD</b>	(Optional) Select the checkbox to use a Data Type Definition (DTD) for the imports and exports.  (This option is not available for the <b>Enterprise Architect XMI/UML 1.3</b> or <b>Rational Rose/Unisys UML 1.3</b> types.)	<a href="#">The UML DTD</a> [483]
<b>Log Import/Export</b>	(Optional) Select the checkbox to log import and export activity to a log file.	
<b>Batch Import</b>	(Optional) Select the checkbox to mark the Package as a Batch Import Package.	
<b>Batch Export</b>	(Optional) Select the checkbox to mark the Package as a Batch Export Package.	
<b>Include sub-packages</b>	(For an <b>Enterprise Architect XMI/UML 1.3</b> export) Select the checkbox to include the entire sub-Package hierarchy of this branch in the exported XMI (default).  Deselect the checkbox to include only the immediate contents of the Package in an XML export (XMI stubs).	
<b>OK</b>	Click on this button to set the Package Control options and close the dialog.	

### Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Configure Packages** permission to configure controlled Packages and Package properties
- For batch import, the file date of the XMI file is stored; you can bypass the batch import if there is no change - that is, the file date of the last import matches that of the current file

### Learn more

- [XMI Import and Export](#) [473]
- [Batch XMI Export](#) [492]
- [Batch XMI Import](#) [493]
- [List of Available Permissions](#) [329]

#### 3.7.3.1.8.3 Remove Package from Control

If required, you can remove the control from a Package. Before removing the Package control, and if it is being used for version control, you must check in the Package.

**Access** **Project Browser | Package Context Menu | Package Control | Configure** (Ctrl+Alt+P)

### Remove control from a Package

Step	Action	See also
1	Select the Package and click on the <b>Configure</b> menu option. The Package Control Options dialog displays.	
2	Deselect the <b>Control Package</b> checkbox. If the Package is under version control, this sets the <b>Version Control</b> field to <b>(None)</b> .	
3	Click on the <b>OK</b> button to remove Package control. Package control for the selected Package has now been removed.	

### Notes

- When disconnecting a Package from version control, the association between the Package and the exported XML file is removed from your model; however, the XML file itself is not removed from version control, nor is it deleted from your local version control working copy folder

This is because it is possible for another model to be using the version controlled Package and still referencing the associated version controlled XML file

### Learn more

- [Check In a Package](#)<sup>[432]</sup>

#### **3.7.3.1.8.4 Save a Package**

Once you have correctly configured a Package, you can save the controlled Package to the designated XML file.

**Access** **Project Browser | Package Context Menu | Package Control | Save Package to File**

### Save a controlled Package to file

Step	Action	See also
1	Select the Package and the <b>Save package to File</b> menu option. The export process executes automatically according to your configured preferences, overwriting any existing file.	

### Notes

- If you are using a version control Package in conjunction with the exported Package files, you must check out the XML file first to enable Enterprise Architect to overwrite the existing version

#### Learn more

- [Configure Packages](#) 

#### 3.7.3.1.8.5 Load a Package

If a Package has been marked for control it is displayed in the Project Browser with a red rectangle to the left of the Package icon. If you have previously saved the controlled Package, you can reload it.

**Access** **Project Browser | Package Context Menu | Package Control | Load package from file**

#### Load a controlled Package from an XML file

Step	Action	See also
1	Select the Package and the <b>Load package from file</b> menu option. If you have configured the Package control details, Enterprise Architect prompts you to confirm the import.	
2	Click on the <b>Yes</b> button to confirm the import. The current Package is deleted and the saved Package is imported.	

#### Notes

- Importing **deletes the current Package entirely from the model, and the action cannot be undone;** you must be careful not to lose any current changes or information

#### Learn more

- [Configure Packages](#) 

#### 3.7.3.1.8.6 Batch XML Export

You can export a group of controlled packages to XML in one step.

**Access** **Project | Model Import/Export | Batch XML Export**

#### How to

To export a group of controlled packages to XML

Step	Action	See also
1	Select the <b>Batch XMI Export</b> menu option. The Batch XMI Export dialog displays.	
2	Select the checkbox against each package to include in this export run. Select the <b>Select All</b> checkbox to select all packages in the list.	
3	Select the <b>Restore Cross References Without Prompting</b> checkbox to automatically restore any missing cross package references without displaying a prompt.	<a href="#">Report Deletion of Cross-Package References</a> <sup>[496]</sup>
3	To save this configuration as the default, click on the <b>Save Settings</b> button.	
4	Click on the <b>Run Export</b> button. Enterprise Architect cycles through each checked package and exports it using the options specified in the Package Control Options dialog. As long as a valid filename exists, Enterprise Architect exports the package to XMI and proceeds to the next package.	<a href="#">Configure Packages</a> <sup>[488]</sup>

**Notes**

- The **Restore Cross References Without Prompting** checkbox is enabled only if the option **Report Cross Package Reference Deletions (XMI 1.1)** is set on the XML Specifications page of the Options dialog
- The option is applicable only to XMI 1.1 controlled packages

**Learn more**

- [XML Specifications](#) <sup>[639]</sup>

**3.7.3.1.8.7 Batch XMI Import**

You can import a group of controlled packages from XMI into Enterprise Architect in one step.

As Enterprise Architect processes each package, it updates the **Status** column against each package name on the Batch XMI Import dialog.

- If the import is successful, Enterprise Architect updates the package status to **Imported**
- If the import is unsuccessful, Enterprise Architect updates the package status to **Not Imported**

Common reasons for an import to fail include:

- The package not being correctly configured
- The last import file date matches the import date of the file currently on disk

**Access** **Project | Model Import/Export | Batch XMI Import**

### **How to**

To import a group of controlled packages from XMI

Step	Action	See also
1	Select the <b>Batch XMI Import</b> menu option. The Batch XMI Import dialog displays.	
2	Select the checkbox against each package to include in the import. Select the <b>Select All</b> checkbox to select all packages in the list.	
3	To save this configuration as the default, click on the <b>Save Settings</b> button.	
4	Click on the <b>Run Import</b> button. Enterprise Architect cycles through each checked package and imports it.	<a href="#">Configure Packages</a> <sup>488</sup>

### **Notes**

- To avoid re-importing the same module multiple times, select the **Check file date before import** checkbox; Enterprise Architect then does not import a file if the last import file date matches that of the one currently on disk

#### **3.7.3.1.8.8 Manual Version Control with XMI**

You can use XMI to support version control by writing model elements in XML text files suitable for use with standard version control software.

Using XMI in this manner enables you to manually connect to third-party version control software outside the Enterprise Architect environment.

Enterprise Architect internally supports the configuration of version control through SCC and CVS configurations.

### **How to**

Before using XMI for version control

Step	Action	See also
1	Select suitable packages in the Project Browser, to be marked as controlled packages.	

Step	Action	See also
2	Configure these with filenames that are visible to a version control system of your choice.	<a href="#">Configure Packages</a> <sup>[488]</sup>
3	Save the controlled packages to establish a model base and check these into the version control system.	<a href="#">Save a Package</a> <sup>[491]</sup>

To apply version control

Step	Action	See also
1	Continue working on a package until versioning is required.	
2	Check out the package XML file from the version control system.	
3	Save the relevant package using the controlled package support.	
4	Check the package back into the version control system.	

To recover an earlier version

Step	Action	See also
1	Save the current version first. This is important, because the package is completely deleted during the import process. If necessary, manually update the version control system.	
2	Get the required package version from the version control system.	
3	Select the package to reload.	
4	Select the <b>Package Control   Load package from file</b> menu option to import the previous version.	<a href="#">Load a Package</a> <sup>[492]</sup>

Step	Action	See also
	Enterprise Architect deletes the controlled package and restores the previous version.	

#### Learn more

- [Version Control](#) <sup>[383]</sup>

#### 3.7.3.1.8.9 Report Deletion of Cross Package References

Your model might contain at least one XML 1.1 controlled Package that is also used in other models, but not necessarily in the same context. For example:

- You have a model of **Power Generation** and a model of **Power Consumption**
- Both models contain the Package called *Power Storage*
- In the Generation model, the *Power Storage* Package has links to a Package called *Turbine Control* (which does not exist in the Consumption model)
- In the Consumption model, the *Power Storage* Package has links to a Package called *Consumer Demand* (which does not exist in the Generation model)
- All Packages are maintained in a master project, **Power Control**, where all the cross-Package relationships exist; compared with Power Control, both Power Generation and Power Consumption are **incomplete**

#### Problem

Consider the process at the end of a work session on *Power Storage* in, say, the **Power Generation** model. When saving the Package to XML, Enterprise Architect has no information on relationships between *Power Storage* and the *Consumer Demand* Package from the **Power Consumption** model, and so these are not represented in the XML file. Consequently, when the XML file is imported into **Power Control**, Enterprise Architect deletes the relationships in the master project because they don't match the file.

#### Solution

You can easily check for and preserve the cross-Package relationships by selecting the **Report Cross Package Reference Deletions (XML 1.1)** checkbox in the Options dialog (**Tools | Options | XML Specifications**). When you save a controlled Package, Enterprise Architect compares the XML for that save with the most recent XML 1.1 revision of the Package. Any items that are missing from the Package are listed on the **Missing Cross References** dialog.

#### Actions

Action	Detail	See also
<b>Merge missing items</b>	Items missing from the exported Package are identified in the Missing Cross References dialog. To regenerate them into the XML file that is being saved: <ul style="list-style-type: none"> <li>• Select the checkbox against each item <b>type</b></li> </ul>	



Action	Detail	See also
	<ul style="list-style-type: none"> <li>Click on the <b>OK</b> button</li> </ul> <p>Items of any type that you do not select are omitted from the saved XML file.</p>	
<b>Abort or cancel this XML Export</b>	<p>If you do not want to proceed with the export (or <b>Check-in</b> or <b>Put Latest</b>) operation:</p> <ul style="list-style-type: none"> <li>Click on the <b>Cancel</b> button</li> </ul>	
<b>Review the details of an entry</b>	<p>To display the details of an entry on the Missing Cross References dialog:</p> <ul style="list-style-type: none"> <li>Double-click on the type name</li> </ul> <p>The details of each item are shown on the Missing &lt;item type&gt; dialog. For any items where <b>information</b> is missing from the XML file, the missing details are grayed-out.</p> <p>To find the item with the missing cross-reference in the Project Browser, either:</p> <ul style="list-style-type: none"> <li>Right-click on the entry in the list and select the <b>Find in Project Browser</b> option, or</li> <li>Click on the item and press <b>(Alt+G)</b></li> </ul>	
<b>Copy a value from a column on the dialog to the clipboard</b>	<p>To preserve a copy of an item from the dialog on the clipboard:</p> <ul style="list-style-type: none"> <li>Click on the column value to highlight it; the selected value is also shown in the bottom left corner of the dialog</li> <li>Press <b>(Ctrl+C)</b> to copy that value to the clipboard</li> </ul>	

#### Learn more

- [XML Specifications](#) <sup>639</sup>

### 3.7.3.2 CSV Import and Export

Enterprise Architect enables you to import and export information about Enterprise Architect elements to and from CSV text files.

#### Topics

Topic	Detail	See also
<b>Specifications</b>	<p>To import and export element data from Enterprise Architect using CSV files, you must first set up one or more file specifications.</p> <p>You do this because the specification defines what types of value from the spreadsheet are to be imported, and how the information is translated between the spreadsheet and Enterprise Architect.</p>	<a href="#">CSV Specifications</a> <sup>498</sup>

Topic	Detail	See also
<b>Import From CSV</b>	Once you have defined a CSV import specification, you can read in major element characteristics from a CSV text file.	<a href="#">CSV Import</a> <sup>[503]</sup>
<b>Export To CSV</b>	Once you have defined a CSV export specification it is possible to write out major element characteristics to a CSV text file.	<a href="#">CSV Export</a> <sup>[504]</sup>

### 3.7.3.2.1 CSV Specifications

To import and export element data to and from Enterprise Architect using CSV files, you must first set up one or more file specifications.

A file specification defines the:

- Fields from the spreadsheet in the order they are imported or exported
- Filename (optional) and
- Delimiter between columns

Once you have defined one or more specifications, one can be selected in the CSV Import/Export dialog as the current specification to apply during an import or export action.

CSV Import/Exports only imports and exports *elements* (within packages) and their properties; items such as Class attributes cannot be imported or exported through this mechanism.

XML import/export provides a solution to this limitation, as does use of the Automation Interface (Object Model).

**Access**    **Project | Model Import/Export | CSV Import/Export Specifications**

#### Reference

On the CSV Import/Export Specification dialog, complete the fields as required, as indicated in the following table:

Field	Usage	See also
<b>Specification Name</b>	Select the unique name for this specification.	
<b>Delimiter</b>	Specify the character delimiter to use between record fields.  If a field contains an instance of the delimiter, the field is exported wrapped in " (quotation marks) and all instances of " in the field are doubled (that is, " becomes "").	
<b>Notes</b>	Record a brief description of the specification.	

Field	Usage	See also
<b>Default Filename</b>	Select the default filename.	
<b>Default Direction</b>	Set the default action - <b>Import</b> or <b>Export</b> .  A specification can be used in either direction, but this enables you to set the default type.	
<b>Default Types</b>	Limit the element types being exported, by entering a comma-separated list: for example: <i>class,requirement,component,node,object</i> . <ul style="list-style-type: none"> <li>If you specify element types, <b>ONLY</b> elements of those types are exported or imported; therefore, in order to enable the <b>Preserve Hierarchy</b> option to operate (if selected) you must include <b>Package</b> as an element type, otherwise there are no packages in which to preserve the hierarchy</li> <li>If you do not specify any default element types, all elements including Packages are exported or imported and the hierarchy can be preserved</li> </ul>	
<b>Preserve Hierarchy</b>	Include fields generated by Enterprise Architect to embed/reconstruct the package hierarchy.	<a href="#">Using Preserve Hierarchy</a> <sup>[500]</sup>
<b>Available Fields</b>	Select from a list of possible record fields, not yet allocated.	
<b>File Specification</b>	List the record fields (in the order they are plotted across the spreadsheet) already assigned.	
<b>Add Tagged Value Field</b>	<p>Adds a Tagged Value column to the CSV Specification, to import or export the value that an element has for a particular tag.</p> <p>When you click on this button, a prompt displays to select one of two options:</p> <ul style="list-style-type: none"> <li><b>Value</b> - to import or export the content of the Tagged Value <b>Value</b> field; a column heading with the prefix <i>TagValue_</i> is added to the specification</li> <li><b>Notes</b> - to import or export the content of the Tagged Value <b>Notes</b> field (some Tagged Values use the <b>Notes</b> field to accommodate long values, such as &lt;memo&gt; text); a column heading with the prefix <i>TagNotes_</i> is added to the specification</li> </ul> <p>In each case, the Add Tag Value Column dialog displays.</p>	<a href="#">Package Browser Options</a> <sup>[683]</sup>

Field	Usage	See also
	<p>Select either the</p> <ul style="list-style-type: none"> <li>• <b>Global Tagged Values</b> option, or</li> <li>• <b>Tagged Values from the selected Element</b> option, and select the required element using the <b>Other Element</b> button</li> </ul> <p>Now select the required Tagged Value from the <b>Tagged Value</b> drop-down list, and click on the <b>OK</b> button.</p>	
<b>Add Field</b>	Move all selected fields in the top list to the bottom list.	
<b>Remove Field</b>	Move all selected fields in the bottom list back to the available list.	
<b>New</b>	Create a new specification.	
<b>Save</b>	Save changes to the currently selected specification.	
<b>Save As</b>	Save the current specification with a new name.	
<b>Delete</b>	Delete the current specification.	
<b>Close</b>	Close this dialog.	

### Notes

- In **Available Fields** and **File Specification**, the record fields **Created Date** and **Modified Date** are not set when imported from CSV

### Learn more

- [XML Import and Export](#)<sup>[473]</sup>
- [Enterprise Architect Object Model](#)<sup>[2804]</sup>

#### 3.7.3.2.1.1 Using Preserve Hierarchy

The **Preserve Hierarchy** option on the CSV Import/Export Specification dialog includes fields generated by Enterprise Architect to embed/reconstruct the package hierarchy in a CSV import or export.

When selected, the **Preserve Hierarchy** option inserts two fields into the CSV specification that are:

- automatically populated by Enterprise Architect on export and
- used to reconstruct the exported package's hierarchy upon import

Field	Usage	See also
<b>CSV_KEY</b>	A unique identifier for the current element.  This key is unique per export; subsequent exports produce different keys for the same set of elements.	
<b>CSV_PARENT_KEY</b>	The corresponding CSV_KEY of the current element's parent.  If the field is left blank or references a non-existent CSV_KEY, the element is added to the top level of the package.	

It is highly recommended that you do not change these fields if they have been automatically generated by Enterprise Architect's CSV exporter.

If you intend to import hierarchical information from a spreadsheet that was **not** populated by exporting data from Enterprise Architect, you must add these two fields to your spreadsheet as the last two columns, and populate the columns yourself.

For example:

NAME	TYPE	NOTES	PRIORITY	STATUS	CSV_KEY	CSV_PARENT_KEY
Requirement Package	Package	Notes Package1			Package1	
REQ1	Requirement	Notes on REQ1	High	Approved	REQ1	Package1
REQ2	Requirement	Notes on REQ2	High	Approved	REQ2	Package1
REQ2.1	Requirement	Notes on REQ2.1	High	Approved	REQ2.1	REQ2
REQ2.2	Requirement	Notes on REQ2.2	Med	Approved	REQ2.2	REQ2
REQ2.3	Requirement	Notes on REQ2.3	High	Approved	REQ2.3	REQ2
REQ3	Requirement	Notes on REQ3	High	Approved	REQ3	Package1
REQ3.1	Requirement	Notes on REQ3.1	High	Approved	REQ3.1	REQ3
REQ3.2	Requirement	Notes on REQ3.2	High	Approved	REQ3.2	REQ3
REQ4	Requirement	Notes on REQ4	High	Approved	REQ4	Package1
REQ4.1	Requirement	Notes on REQ4.1	High	Approved	REQ4.1	REQ4
REQ4.2	Requirement	Notes on REQ4.2	High	Approved	REQ4.2	REQ4
REQ4.3	Requirement	Notes on REQ4.3	High	Approved	REQ4.3	REQ4

### 3.7.3.2.2 CSV Export

Enterprise Architect enables you to export information about elements to a CSV text file, using a CSV export specification.

If you intend to *re-import* the exported information into Enterprise Architect at some point, it is recommended that you include the **GUID** field in the CSV export specification. This ensures that Enterprise Architect can identify and update existing elements, rather than creating duplicates.

**Access** [Project Browser](#) | [Package Context Menu](#) | [Import/Export](#) | [CSV Import/Export](#)

**How to**

To export data in CSV format, follow the steps below

1. In the Project Browser, right-click on the package containing the elements to export and select the **CSV Import/Export** menu option.
2. The CSV Import/Export dialog displays; complete the fields as required, as indicated in the following table:

Field	Usage	See also
<b>Package</b>	Confirm the name of the current selected package.	
<b>Specification</b>	Specify the name of the export specification <sup>[475]</sup> to use.	<a href="#">CSV Specifications</a> <sup>[498]</sup>
<b>Edit/New</b>	Edit the export specification or create a new one.	
<b>File</b>	Specify the filename to export to.	
<b>Types</b>	<p>List the element types to export: leave blank for all, or enter a comma-separated list of types.</p> <p>If you specify element types, <b>ONLY</b> elements of those types are exported; therefore, to enable the <b>Preserve Hierarchy</b> option in the specification to operate (if selected) you must include <i>Package</i> as an element type, otherwise no Packages are exported in which to preserve the hierarchy.</p> <p>If you do not specify any element types, all elements including Packages are exported and the hierarchy can be preserved.</p>	
<b>Action</b>	Select the <b>Export</b> radio button to export to file.	
<b>Print Results</b>	Print out the result list.	
<b>View File</b>	View the resultant CSV file with the default Windows application for CSV files.	
<b>Run</b>	Perform the export.	
<b>Close</b>	Exit this dialog.	

#### Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have both **Export XMI** and **Import XMI** permissions to use the **CSV Import/Export** option

**Learn more**

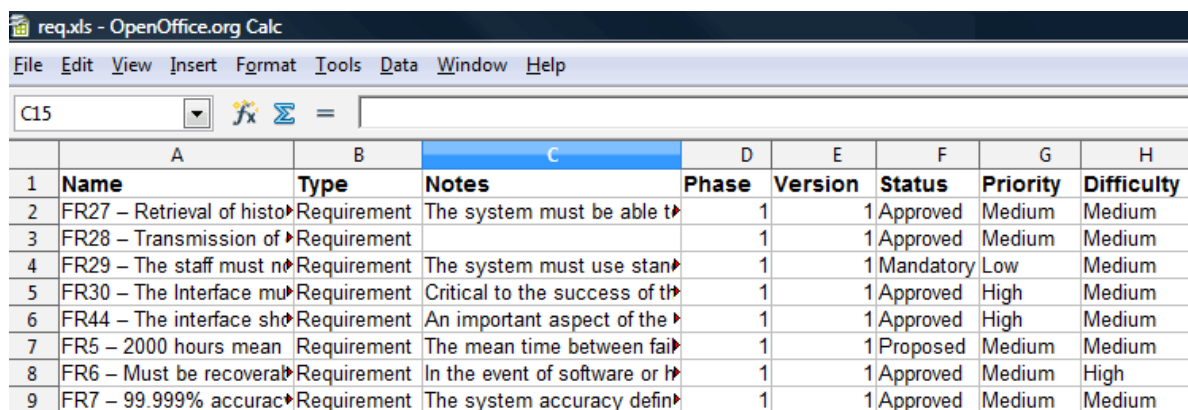
- [List of Available Permissions](#) <sup>329</sup>

**3.7.3.2.3 CSV Import**

Enterprise Architect enables you to import information about elements from a CSV text file, using a CSV export specification.

You import the CSV file into a selected package; if this package or any element within the package has a lock on it, you cannot import the CSV file into it. The **Import** option on the dialog is grayed out.

The format and content of the source data file should resemble the following:



The screenshot shows a spreadsheet window titled 'req.xls - OpenOffice.org Calc'. The spreadsheet contains a table with 9 rows and 9 columns. The columns are labeled A through H. The data is as follows:

	A	B	C	D	E	F	G	H
1	<b>Name</b>	<b>Type</b>	<b>Notes</b>	<b>Phase</b>	<b>Version</b>	<b>Status</b>	<b>Priority</b>	<b>Difficulty</b>
2	FR27 – Retrieval of histo	Requirement	The system must be able to	1	1	Approved	Medium	Medium
3	FR28 – Transmission of	Requirement		1	1	Approved	Medium	Medium
4	FR29 – The staff must not	Requirement	The system must use stan	1	1	Mandatory	Low	Medium
5	FR30 – The Interface mu	Requirement	Critical to the success of th	1	1	Approved	High	Medium
6	FR44 – The interface sho	Requirement	An important aspect of the	1	1	Approved	High	Medium
7	FR5 – 2000 hours mean	Requirement	The mean time between fai	1	1	Proposed	Medium	Medium
8	FR6 – Must be recovera	Requirement	In the event of software or h	1	1	Approved	Medium	High
9	FR7 – 99.999% accurac	Requirement	The system accuracy defin	1	1	Approved	Medium	Medium

**Access** [Project Browser](#) | [Package Context Menu](#) | [Import/Export](#) | [CSV Import/Export](#)

**How to**

To import data in CSV format

1. In the Project Browser, right-click on the package to import into and select the **CSV Import/Export** context menu option.
2. The CSV Import/Export dialog displays; set the required options; as outlined below:

Field	Usage	See also
<b>Package</b>	Confirm the name of the current selected package.	
<b>Specification</b>	Specify the name of the import specification <sup>2613</sup> to use.	<a href="#">Import specification</a> <sup>498</sup>
<b>Edit/New</b>	Edit the import specification or create a new one.	
<b>File</b>	Specify the spreadsheet filename to import from.	
<b>Types</b>	Not used for import.	

Field	Usage	See also
<b>Action</b>	Select the <b>Import</b> radio button to import from the file (grayed-out if the package or a child item in the package is locked).	
<b>Print Results</b>	Print out the result list.	
<b>View File</b>	View the source CSV file with the default Windows application for CSV files.	
<b>Run</b>	Perform the import.	
<b>Close</b>	Exit this dialog.	

#### Notes

- When importing, Enterprise Architect checks the specification to see if there is a GUID field included; if there is, Enterprise Architect attempts to locate the element identified by the GUID and, if successful, updates the current element rather than creating a new one
- If no GUID field is defined, or Enterprise Architect cannot locate the identified element, a new element is created and placed in the current package; during import, **Type** is a mandatory field in the source file and must match one or more of the legal Enterprise Architect element types: for example: Requirement, or Class
- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have both **Export XMI** and **Import XMI** permissions to use the **CSV Import/Export** option

#### Learn more

- [List of Available Permissions](#)<sup>[329]</sup>
- [Enterprise Architect element types](#)<sup>[289b]</sup>

### 3.7.3.3 Perform a Project Data Transfer


In normal system maintenance, it is likely that at some point you need to relocate data between storage systems. In any edition of Enterprise Architect, you might need to transfer all records from one project file to another (recommended after serious network crash or repeated database corruption). In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, you might need to:

- Transfer an existing project file to a DBMS repository such as SQL Server or MySQL
- Transfer a DBMS repository to an existing project file
- Move a repository from one DBMS to another (including moving between servers using the same DBMS)
- Copy all records from a JET 3.5 to JET 4 (Access 2000 or XP) repository - or back the other way



[Access](#) [Tools](#) | [Data Management](#) | [Project Transfer](#)


### Perform a project data transfer

Step	Action	See also
1	Click on the option for the required transfer type: <ul style="list-style-type: none"> <li>• <b>.EAP to .EAP</b></li> <li>• <b>DBMS to .EAP</b></li> <li>• <b>.EAP to DBMS</b></li> <li>• <b>DBMS to DBMS</b></li> </ul>	
2	In the <b>Source Project</b> and <b>Target Project</b> fields, type or select the name or connection string for the Source and Target projects.	
3	If you want to capture the transfer in a log file, select the <b>Logfile</b> checkbox and browse for the appropriate log file location.	
4	Click on the <b>Transfer</b> button.	
5	It is good practice to do a Project Compare after this process to verify that all records are written.	<a href="#">Compare Projects</a> 

### Notes

- Take a backup of the target project to ensure that you can recover any important information it already contains; in a transfer, all records in the target repository are overwritten
- You cannot move a model from a source .EAP file of a version earlier than 3.5.0
- When performing an EAP to DBMS transfer, if the EAP file has replication enabled, the replication must be removed before performing the transfer
- When transferring a model to a DBMS, you cannot transfer the data into an empty database - the tables must already exist; if they do not, run the corresponding script supplied by Sparx Systems to create the required tables
- When transferring a model into a project file, similarly, the target project file must already exist, to provide the data structures to transfer data into
- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Transfer Data** permission to transfer project data between repositories

### Learn more

- [Sparx Systems Scripts webpage](#)
- [Remove Replication](#) 

- [List of Available Permissions](#) <sup>329</sup>

### 3.7.3.3.1 Copy Packages Between Projects

It is possible to copy and move individual **packages** between **projects**, exporting and importing the packages as **XMI** files. You can also duplicate larger structures, such as complete models or projects. This gives you a high level of flexibility in building a project from re-usable parts and from elements produced in widely-dispersed geographic regions.

**Access** **Project Browser | Package Context Menu | Import/Export | Export package to XMI file**  
**Project Browser | Package Context Menu | Import/Export | Import package from XMI file**

#### Copy a package from one Enterprise Architect project to another

Step	Action	See also
1	Open the project to copy <b>from</b> .	
2	In the Project Browser, right-click on the package to copy and select the <b>Export package to XMI file</b> option. The Export Package to XMI dialog displays.	
3	Select the appropriate options and filename.	<a href="#">Export to XMI</a> <sup>475</sup>
4	Click on the <b>Export</b> button to begin the export process.	
5	When the export is complete, open the project to copy <b>into</b> .	
6	In the Project Browser, navigate to the location to import the package into.	
7	Right-click and select the <b>Import package from XMI file</b> option. The Import Package from XMI dialog displays.	
8	Select the appropriate options and filename.	<a href="#">Import from XMI</a> <sup>478</sup>
9	Click on the <b>Import</b> button. The package is copied from the source project to the destination project.	

#### Notes

- If the package you are importing already exists in the target project (that is, it has been imported previously), you must either import **over** the existing package or select the **Strip GUIDs** option

In the latter case, the system creates a copy of the original package

#### Learn more

- [Copy a Package](#)<sup>[775]</sup>
- [Copy Elements Between Packages](#)<sup>[915]</sup>

### 3.7.4 Compare Projects

A number of operations can make changes to your project that you either want to monitor carefully or not have at all. Such events include:

- Recovering from a database crash
- Restoring a backup
- Performing a Project Data Transfer
- Importing from XML, and
- Deleting model elements

You might have made a copy of the original project or the purpose of the operation is to generate a copy, in which case you can compare the size and row counts of the 'before' and 'after' copies as a convenient 'sanity check'. The copies can be on different platforms; you have options to:

- Compare a project file to another project file
- Compare a project file to a DBMS-based repository
- Compare two DBMS repositories

The comparison examines the number of project rows in each database, producing a report indicating the total records in each and the difference in record count between the two. If discrepancies are found, you must investigate further manually. The comparison does not examine the actual **data** in the tables.

**Access** [Tools | Data Management | Project Compare](#)

#### Compare two projects

Step	Action	See also
1	On the Project Compare dialog, select the radio button for the database types of the two projects you want to compare: <ul style="list-style-type: none"> <li>• <b>.EAP to .EAP</b></li> <li>• <b>DBMS to .EAP</b></li> <li>• <b>.EAP to DBMS</b></li> <li>• <b>DBMS to DBMS</b></li> </ul>	
2	In the <b>Source Project</b> and <b>Target Project</b> fields, type the name or connection string for the source and target projects to compare.	

Step	Action	See also
3	Click on the <b>Compare Projects</b> button. The results of the comparison display in the panel at the bottom of the dialog.	
4	If you want to print the results of the comparison, click on the <b>Print List</b> button.	

### 3.8 Project Management



Enterprise Architect provides strong support for:

- Project Management, in estimating project size, measuring risk and effort, and assigning resources to elements
- Change control and maintenance

#### Topics

Topic	Detail	See also
<b>Metrics and Estimation</b>	<p>Project estimation is working out how much time and effort is required to build and deploy a solution.</p> <p>Enterprise Architect provides the Use Case metrics facility as a means of:</p> <ul style="list-style-type: none"> <li>• Measuring the complexity of a system</li> <li>• Getting an indication of the effort required to implement the model</li> <li>• Getting an indication of the project timescale</li> </ul> <p>You base these estimates on carefully-calibrated metrics.</p>	<a href="#">Use Case Estimation</a> <sup>[584]</sup>
<b>Resource Management</b>	<p>Resources are the people who work on a project.</p> <p>You can assign roles to resources and allocate tasks on specific model elements, which enables tracking of effort and estimation of time to complete.</p>	<a href="#">Project Resources</a> <sup>[512]</sup>
<b>Project Maintenance</b>	<p>During a project you monitor and manage the development and progress of individual model elements.</p> <p>You can record problems, changes, issues and tasks that affect these individual elements as they arise, and document the solution and associated details.</p> <p>Similarly, Enterprise Architect helps you to manage changes and issues that apply to the whole system.</p>	<a href="#">Maintenance</a> <sup>[2619]</sup> <a href="#">Changes and Issues</a> <sup>[2631]</sup>
<b>Project Tasks and Issues</b>	<p>In the course of a project, there are various non-technical tasks that are vital to the successful management and completion of the project, such as meetings.</p> <p>Enterprise Architect helps you to record and monitor these, and to manage non-technical project issues as</p>	<a href="#">Project Tasks</a> <sup>[526]</sup> <a href="#">Project Issues</a> <sup>[528]</sup> <a href="#">Project Task Allocation</a> <sup>[538]</sup>

Topic	Detail	See also
	they arise.	

#### Learn more

- [The Project Management Window](#)<sup>[510]</sup>
- [Personal Tasks](#)<sup>[554]</sup>
- [Model Mail](#)<sup>[565]</sup>
- [The Project Status View](#)<sup>[525]</sup>
- [Project Calendar](#)<sup>[571]</sup>
- [Project Glossary](#)<sup>[533]</sup>
- [Update Package Status](#)<sup>[590]</sup>
- [Manage Bookmarks](#)<sup>[591]</sup>
- [Monitor Change Events](#)<sup>[593]</sup>

### 3.8.1 The Project Management Window

The Project Management window enables you to input the following quantities associated with an element contained in the model:

- Resources - the people who work on a project, who can be assigned roles and allocated tasks
- Effort - effort expended in work on the element
- Risks - risk associated with the element
- Metrics - metrics measured for an element

Each of these four quantities has a separate tab in the window.

**Access** **Element | Project Management** (Ctrl+Shift+7)

#### Reference



Action	Usage	See also
<b>New</b>	Create a new item.	<a href="#">Resource Allocation</a> <sup>[512]</sup> <a href="#">Effort Management</a> <sup>[518]</sup> <a href="#">Risk Management</a> <sup>[519]</sup> <a href="#">Metrics</a> <sup>[520]</sup>

<b>Save</b>	Save changes to an item.	
<b>Save As New</b>	Duplicate an existing entry. You must change an item's Role for this to become enabled.	
<b>Delete</b>	Delete an item from the list.	
<b>Sort</b>	Sort the items in the list into alphabetical order.	
<b>Print</b>	Print item data from the list.	
<b>Browse Element</b>	Display the Element Browser window for the selected element, to list and select the project management items for the element.	<a href="#">The Element Browser</a> <sup>[989]</sup>
<b>Show/Hide Properties</b>	Swap between detailed and summary window styles.	
<b>Help</b>	Show help contents for this window.	

### Notes

- Click on an element in the Project Browser to display project management information for that element in the Project Management window
- The window has two formats:
  - *List* mode - provides a detailed list showing several columns of information
  - *Item* mode - a summary list plus an editor to add or edit information for a single selected item
- Toggle between these modes using the **Show/Hide Properties** button in the window toolbar; each tab toggles between Item mode and List mode independently
- Columns in the item list can be reorganized, added, removed, grouped, filtered and sorted using the options provided in the **List Header** facilities
- An asterisk on a tab name indicates that the tab contains saved information; if there is no information for a category of item, or the information has not yet been saved, its tab has no asterisk
- Right-click on the list to view the context menu, which enables you to add and delete list items

### Learn more

- [List Header](#)<sup>[677]</sup>

### 3.8.2 Project Resources

Resources are the people who work on a project. They can be assigned roles and allocated tasks, which enables tracking of effort and estimation of time to complete.

Resources are added, modified and deleted using the Project Management window.

**Access**    **Element | Project Management (Ctrl+Shift+7)**

#### Use to

Usage	See also
Allocate a resource to an element.	<a href="#">Resource Allocation</a> <sup>[512]</sup>
Record additional project management information for an element.	<a href="#">Effort Management</a> <sup>[518]</sup> <a href="#">Risk Management</a> <sup>[519]</sup> <a href="#">Metrics</a> <sup>[520]</sup>
Obtain a report of resource allocation details.	<a href="#">Resource Report</a> <sup>[546]</sup>
Configure Project Management data and populate the drop-down lists used on the Project Management dialog tabs.	<a href="#">Roles</a> <sup>[1155]</sup> <a href="#">Clients</a> <sup>[1157]</sup> <a href="#">Effort Types</a> <sup>[522]</sup> <a href="#">Metric Types</a> <sup>[523]</sup> <a href="#">Risk Types</a> <sup>[524]</sup>

#### Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Project Settings** permission to update and manage project resources, effort, metrics and risks

#### Learn more

- [The Project Management Window](#) <sup>[510]</sup>
- [List of Available Permissions](#) <sup>[329]</sup>

#### 3.8.2.1 Resource Allocation

In developing a model, people (or **resources**) in a number of **roles** might perform **tasks** on model structures. As a Project Manager, you can assign resources to tasks on **elements** (including Packages) in the model, planning and monitoring the work that they do within the timeframe you have allocated for that work to be completed. You can do this using the Project Management and/or Project Gantt View facilities.



**Access** Click on element and:

**Project | Project Gantt View > Element View | right click element | Assign Resource, or  
Element | Project Management (Ctrl+Shift+7) > Resource Allocation: New**

**Enter resource allocation details for an element**

Field/Option/ Button	Action	See also
<b>Resource</b>	<p>Either:</p> <ul style="list-style-type: none"> <li>Type in the name of the resource being assigned to work associated with the element, or</li> <li>Click on the drop-down arrow and click on the name of the resource</li> </ul> <p>If you want to assign more than one resource to the task on the element, you follow a slightly different procedure.</p>	<a href="#">Assign Multiple Resources</a> 515
<b>Role or Task</b>	<p>Either:</p> <ul style="list-style-type: none"> <li>Type in the role name or task performed by the selected resource, or</li> <li>Click on the drop-down arrow and click on the role performed by the resource</li> </ul>	
<b>Start Date</b>	<p>This field defaults to today's date.</p> <p>If you want to assign a different start date for the task:</p> <ol style="list-style-type: none"> <li>Click on the down arrow.</li> <li>If necessary, click on the left or right arrow to select the previous or next month.</li> <li>Click on the appropriate day of the month as the start date.</li> </ol> <p>The field immediately changes to the selected date.</p>	
<b>End Date</b>	<p>This field defaults to today's date.</p> <p>If you want to assign a different end date for the task:</p> <ol style="list-style-type: none"> <li>Click on the down arrow.</li> <li>If necessary, click on the right arrow to select the next month.</li> <li>Click on the appropriate day of the month as the end date.</li> </ol> <p>The field immediately changes to the selected date.</p>	
<b>Complete %</b>	<p>If the task is already in progress, type in the current percentage completion.</p>	

Field/Option/ Button	Action	See also
<b>Expected Time</b>	<p>Type in the number of whole time units the task is expected to take. (The value must be an integer, so you cannot record part units.)</p> <p>The unit of time you adopt is by agreement within the project, and depends on the granularity of the work being recorded. Most tasks are completed in a number of hours or a number of days; use the smallest practical unit that you can record as a whole number.</p> <p>Apply the same units as used for the <b>Allocated Time</b> and <b>Time Expended</b> fields.</p>	
<b>Allocated Time</b>	<p>Type in the number of time units the task can be spread over. (The value must be an integer, so you cannot record part units.)</p> <p>The unit of time you adopt is by agreement within the project, and depends on the granularity of the work being recorded. Most tasks are completed in a number of hours or a number of days; use the smallest practical unit that you can record as a whole number.</p> <p>Apply the same units as used for the <b>Expected Time</b> and <b>Time Expended</b> fields.</p>	
<b>Time Expended</b>	<p>(When the task is eventually 100% complete, you will type in the number of time units it actually took. The value must be an integer, so you cannot record part units.)</p> <p>The unit of time you adopt is by agreement within the project, and depends on the granularity of the work being recorded. Most tasks are completed in a number of hours or a number of days; use the smallest practical unit that you can record as a whole number.</p> <p>Apply the same units as used for the <b>Expected Time</b> and <b>Allocated Time</b> fields.</p>	
<b>Description</b>	Type (and, if you prefer, format) a description of the work being done by the resource (this text is also displayed in the Notes window, but it cannot be edited in that window).	<a href="#">Notes</a> <sup>1142</sup>
<b>History</b>	(As the task progresses, you add text to this tab to record the activities, progress, problems and outcomes of the task. This text is also displayed in the Notes window, but it cannot be edited in that window.)	
<b>Save (Toolbar icon)</b>	<p>(Project Management window toolbar.)</p> <p>Click on this icon to save the data you have entered and to add the entry to the Resource list in the left column of the window.</p>	
<b>OK</b>	<p>(Assigned Resources dialog.)</p> <p>Click on this button to <b>save</b> the data you have entered, and close the</p>	

Field/Option/ Button	Action	See also
	dialog.	
<b>Cancel</b>	(Assigned Resources dialog.) Click on this button to <b>discard</b> the data you have entered, and close the dialog.	
<b>Help</b>	(Assigned Resources dialog.) Click on this button to display the <i>Assign Multiple Resources</i> Help topic.	<a href="#">Assign Multiple Resources</a> 515

### Notes

- If you allocate resources to Use Cases, the information you enter contributes to the Use Case Estimation calculation for estimating the project size in terms of time, resources and cost
- To edit existing Resource Allocation items for this element, click on the required item in the:
  - List panel to the left of the window, in *Item* mode
  - List, in *List* mode
  - Project Management folder in the Element Browser window - if this window is not displayed, click on the **Browse Element** icon in the Project Management window toolbar; resource allocation item icons have an **R** in the bottom right corner

### Learn more

- [The Project Management Window](#) 510
- [The Element Browser](#) 989
- [Use Case Estimation](#) 584
- [Estimating Project Size](#) 588

#### 3.8.2.1.1 Assign Multiple Resources


When you have a large unit of work associated with an element - such as a Package element - you might want to assign several resources to that element to perform that unit of work. You can assign each person individually, but if the resources have the same role or task you can assign all of them together in one operation, using the Assigned Resources dialog.

#### Access **Click on element and:**

**Project | Project Gantt View > Element View | right click element | Assign Resource, or  
Element | Project Management > Resource Allocation: Show/Hide (to show List view): New**

#### Assign multiple resources to an element

Field/Option/ Button	Action	See also
<b>Resource</b>	<p>(To assign a <b>single</b> resource, simply type in the name or click on the drop-down arrow and select the name from the list. In Project Management, it is simpler to assign single resources directly through the Project Management window.)</p> <p>For <b>multiple</b> resources, click on the <b>Browse ( ... )</b> button to display the Assign Resource dialog.</p> <ol style="list-style-type: none"> <li>1. Select the checkbox against each resource to assign to the element; to select all resources listed, click on the <b>Select All</b> button.</li> <li>2. Click on the <b>OK</b> button to close the dialog and add the selected resources to the <b>Resource</b> field.</li> </ol> <p>The <b>Resource</b> field now shows the selected resources, but grayed out. The drop-down arrow also is not active. To change the resources in the field, click on the <b>Browse</b> button again and select/clear the appropriate checkboxes.</p>	<a href="#">Resource Allocation</a> 512
<b>Role or Task</b>	<p>Either:</p> <ul style="list-style-type: none"> <li>• Type in the role name or task that is common to all the selected resources, or</li> <li>• Click on the drop-down arrow and click on the role that is common to all the resources</li> </ul>	
<b>Start Date</b>	<p>This field defaults to today's date.</p> <p>If you want to assign a different start date for the task:</p> <ol style="list-style-type: none"> <li>1. Click on the down arrow.</li> <li>2. If necessary, click on the left or right arrow to select the previous or next month.</li> <li>3. Click on the appropriate day of the month as the start date.</li> </ol> <p>The field immediately changes to the selected date.</p>	
<b>End Date</b>	<p>This field defaults to today's date.</p> <p>If you want to assign a different end date for the task:</p> <ol style="list-style-type: none"> <li>1. Click on the down arrow.</li> <li>2. If necessary, click on the right arrow to select the next month.</li> <li>3. Click on the appropriate day of the month as the end date.</li> </ol> <p>The field immediately changes to the selected date.</p>	
<b>Complete %</b>	If the task is already in progress, type in the current percentage completion.	
<b>Expected Time</b>	Type in the number of time units the task is expected to take. (The value must be an integer, so you cannot record part units.)	

Field/Option/ Button	Action	See also
	<p>The unit of time you adopt is by agreement within the project, and depends on the granularity of the work being recorded. Most tasks are completed in a number of hours or a number of days; use the smallest practical unit that you can record as a whole number.</p> <p>Apply the same units as used for the <b>Allocated Time</b> and <b>Time Expended</b> fields.</p>	
<b>Allocated Time</b>	<p>Type in the number of time units the task can be spread over. (The value must be an integer, so you cannot record part units.)</p> <p>The unit of time you adopt is by agreement within the project, and depends on the granularity of the work being recorded. Most tasks are completed in a number of hours or a number of days; use the smallest practical unit that you can record as a whole number.</p> <p>Apply the same units as used for the <b>Expected Time</b> and <b>Time Expended</b> fields.</p>	
<b>Time Expended</b>	<p>(When the task is eventually 100% complete, you will type in the number of time units it actually took. The value must be an integer, so you cannot record part units.)</p> <p>The unit of time you adopt is by agreement within the project, and depends on the granularity of the work being recorded. Most tasks are completed in a number of hours or a number of days; use the smallest practical unit that you can record as a whole number.</p> <p>Apply the same units as used for the <b>Expected Time</b> and <b>Allocated Time</b> fields.</p>	
<b>Description</b>	Type (and, if you prefer, format) a description of the work being done by the resources (this text is also displayed in the Notes window, but it cannot be edited in that window).	<a href="#">Notes</a> 
<b>History</b>	(As the task progresses, you add text to this tab to record the activities, progress, problems and outcomes of the task. This text is also displayed in the Notes window, but it cannot be edited in that window.)	
<b>OK</b>	Click on this button to <b>save</b> the data you have entered, and close the dialog.	
<b>Cancel</b>	Click on this button to <b>discard</b> the data you have entered, and close the dialog.	
<b>Help</b>	Click on this button to display this Help topic.	

**Notes**

- When the resource details are displayed on the Project Gantt Chart or Project Management dialog, each resource has their own record containing, initially, the same data; however, each resource can be monitored and their progress recorded separately
- If you assign resources that are already in a multiple resource assignment to the selected element, a prompt displays for you to confirm that you are **updating** the assignment of that resource, or now **omitting** that resource from the current assignment

#### Learn more

- [The Project Management Window](#) <sup>[510]</sup>
- [Resource View](#) <sup>[538]</sup>
- [Element View](#) <sup>[542]</sup>
- [The Gantt View](#) <sup>[594]</sup>

### 3.8.2.2 Effort Management

In Enterprise Architect, the Project Manager can allocate effort (as time) to work on a given model element. To select the element to which to allocate effort, click on the required element in the Project Browser.

**Access** **Element | Project Management > Effort: New**

#### Enter effort allocation details for an element

Step	Action	See also
1	Select the element in the Project Browser.	
2	Select the <b>Project Management</b> menu option. The Project Management window displays, showing the Resource Allocation tab.	
3	Click on the Effort tab.	
4	Click on the <b>New</b> icon on the Project Management window toolbar.	
5	On the Effort tab enter the: <ul style="list-style-type: none"> <li>• Name of the effort (a short description)</li> <li>• Type of the effort (type the name or click on the drop-down arrow and select; the selection list is drawn from the global Effort Type list, but any new efforts you type in the Type field are not added to the list)</li> <li>• Time the effort is expected to expend</li> <li>• Notes on effort (this text is also displayed in the Notes window; it cannot be edited in that window)</li> </ul>	<a href="#">Effort Types</a> <sup>[522]</sup> <a href="#">Notes</a> <sup>[1142]</sup>
6	Click on the <b>Save</b> icon in the Project Management window toolbar.	

Step	Action	See also

### Notes

- To edit existing Effort items for this element, click on the required item in the:
  - List panel to the left of the window, in *Item* mode
  - List, in List mode, or
  - Project Management folder in the Element Browser window - if this window is not displayed, click on the **Browse Element** icon in the Project Management window toolbar; effort item icons have an **E** in the bottom right corner
- Although Enterprise Architect does not currently provide detailed reports on effort within a model, you can use the Automation Interface or similar tools to create your own custom reports based on effort information you enter

### Learn more

- [The Element Browser](#)<sup>[989]</sup>
- [Enterprise Architect Object Model](#)<sup>[2804]</sup>

### 3.8.2.3 Risk Management

In Enterprise Architect, the Project Manager can allocate the possible weighting of defined risks to work on a given model element.

To select the element to which to allocate risk weightings, click on the required element in the Project Browser.

**Access**    **Element | Project Management > Risks: New**

### Use to

- Enter the details of risks that might impact work on an element

### How to

To enter risk details for an element

Step	Action	See also
1	Select the element in the Project Browser.	
2	Select the <b>Project Management</b> menu option. The Project Management window displays, showing the Resource Allocation tab.	

Step	Action	See also
3	Click on the Risks tab.	
4	Click on the <b>New</b> icon on the Project Management window toolbar.	
5	On the Risks tab enter the: <ul style="list-style-type: none"> <li>Name of the risk (a short description)</li> <li>Type of the risk (Mandatory: type the name or click on the drop-down arrow and select; the selection list is drawn from the global Risk Type list, but any new risks you type in the Type field are not added to the list)</li> <li>Weighting allocated to the risk</li> <li>Notes on the risk (this text is also displayed in the Notes window; it cannot be edited in that window)</li> </ul>	<a href="#">Risk Types</a> <sup>[524]</sup> <a href="#">Notes</a> <sup>[1142]</sup>
6	Click on the <b>Save</b> icon in the Project Management window toolbar.	

### Notes

- The risks described here are not the same as those represented by Risk elements; the risks described above are properties of a single element, whilst the Risk element represents something that can impact a range of other elements
- If you select a global risk type from the **Type** drop-down list and the associated **Weight** field is empty, the default Weight value is allocated to the **Weight** field on the Risks tab
- To edit existing Risk items for this element, click on the required item in the:
  - List panel to the left of the window, in *Item* mode
  - List, in List mode
  - Project Management folder in the Element Browser window - if this window is not displayed, click on the **Browse Element** icon in the Project Management window toolbar; risk item icons have an **Ri** in the bottom right corner
- Although Enterprise Architect does not currently provide detailed reports on risks within a model, you can use the Automation Interface or similar tools to create your own custom reports based on risk information you enter

### Learn more

- [The Element Browser](#) <sup>[989]</sup>
- [Enterprise Architect Object Model](#) <sup>[2804]</sup>
- [Risk Elements](#) <sup>[2009]</sup>

### 3.8.2.4 Metrics

In Enterprise Architect, the Project Manager can allocate the possible weighting of defined metrics to work on a given model element.

To select the element to which to allocate metric weightings, click on the required element in the Project



Browser.

**Access** **Element | Project Management > Metrics: New**

### **Use to**

- Enter the details of metrics that you might apply to work on an element

### **How to**

To enter metric details for an element

Step	Action	See also
1	Select the element in the Project Browser.	
2	Select the <b>Project Management</b> menu option. The Project Management window displays, showing the Resource Allocation tab.	
3	Click on the Metrics tab.	
4	Click on the <b>New</b> icon on the Project Management window toolbar.	
5	On the Metrics tab enter the: <ul style="list-style-type: none"> <li>• Name of the metric (a short description)</li> <li>• Type of the metric (Mandatory: type the name or click on the drop-down arrow and select; the selection list is drawn from the global Metric Type list, but any new metrics you type in the Type field are not added to the list)</li> <li>• Weighting allocated to the metric</li> <li>• Notes on the metric (this text is also displayed in the Notes window; it cannot be edited in that window)</li> </ul>	<a href="#">Metric Types</a> <sup>[523]</sup> <a href="#">Notes</a> <sup>[1142]</sup>
6	Click on the <b>Save</b> icon in the Project Management window toolbar.	

### **Notes**

- If you select a global metric type from the **Type** drop-down list and the associated **Weight** field is empty, the default Weight value is allocated to the **Weight** field on the Metrics tab
- To edit existing Metric items for this element, click on the required item in the:
  - List panel to the left of the window, in *Item* mode
  - List, in List mode

- Project Management folder in the Element Browser window - if this window is not displayed, click on the **Browse Element** icon in the Project Management window toolbar; metric item icons have an **M** in the bottom right corner
- Although Enterprise Architect does not currently provide detailed reports on metrics within a model, you can use the Automation Interface or similar tools to create your own custom reports based on metric information you enter

#### Learn more

- [The Element Browser](#)<sup>[989]</sup>
- [Enterprise Architect Object Model](#)<sup>[2804]</sup>

### 3.8.2.5 Effort Types

Enterprise Architect enables you to add an effort *type* to the global list of effort types that can be added to any element in the model.

The global list of effort types displays in the **Type** field drop-down list on the Effort tab of the Project Management window.

**Access** **Settings | Project Types | Project Indicators > Effort: New**

#### Use to

- Specify the effort types used when assigning effort to an element

#### How to

To add a new effort type to the global list

Step	Action	See also
1	Select the <b>Project Indicators</b> menu option. The Project Indicators dialog displays.	
2	Click on the Effort tab.	
3	Click on the <b>New</b> button. (To edit an existing effort type, click on the effort type name in the <b>Defined Effort Types</b> list.)	
4	Complete the fields as follows: <ul style="list-style-type: none"> <li>• In the <b>Effort</b> field type the name of the effort type</li> <li>• In the <b>Description</b> field type a short description of the effort type</li> <li>• In the <b>Weight</b> field type the default weighting to apply to the effort type</li> <li>• In the <b>Note</b> field, type any additional information on the effort type</li> </ul>	

Step	Action	See also
5	Click on the <b>Save</b> button.	

### Notes

- Although Enterprise Architect does not currently provide detailed reports on effort within a model, you can use the Automation Interface or similar tools to create your own custom reports based on effort information you enter
- You can transport effort types between models, using Export Reference Data and Import Reference Data

### Learn more

- [The Project Management Window](#)<sup>[510]</sup>
- [Enterprise Architect Object Model](#)<sup>[2804]</sup>
- [Export Reference Data](#)<sup>[376]</sup>
- [Import Reference Data](#)<sup>[380]</sup>

## 3.8.2.6 Metric Types

Enterprise Architect enables you to add a metric *type* to the global list of metric types that can be added to any element in the model.

The global list of metric types displays in the **Type** field drop-down list on the Metrics tab of the Project Management window.

**Access** **Settings | Project Types | Project Indicators > Metric: New**

### Use to

- Specify the metric types used when assigning metrics to an element

### How to

To add a new metric type to the global list

Step	Action	See also
1	Select the <b>Project Indicators</b> menu option. The Project Indicators dialog displays.	
2	Click on the Metric tab.	
3	Click on the <b>New</b> button. (To edit an existing metric type, click on the metric type name in the <b>Defined Metrics</b> list.)	

Step	Action	See also
4	Complete the fields as follows: <ul style="list-style-type: none"> <li>• In the <b>Metric Type</b> field type the name of the metric type</li> <li>• In the <b>Description</b> field type a short description of the metric type</li> <li>• In the <b>Weight</b> field type the default weighting to apply to the metric type</li> <li>• In the <b>Note</b> field, type any additional information on the metric type</li> </ul>	
5	Click on the <b>Save</b> button.	

### Notes

- Although Enterprise Architect does not currently provide detailed reports on metrics within a model, you can use the Automation Interface or similar tools to create your own custom reports based on metric information you enter
- You can transport metric types between models, using Export Reference Data and Import Reference Data

### Learn more

- [The Project Management Window](#) <sup>[510]</sup>
- [Enterprise Architect Object Model](#) <sup>[2804]</sup>
- [Export Reference Data](#) <sup>[376]</sup>
- [Import Reference Data](#) <sup>[380]</sup>

### 3.8.2.7 Risk Types

Enterprise Architect enables you to add a risk *type* to the global list of risk types that can be added to any element in the model.

The global list of risk types displays in the **Type** field drop-down list on the Risks tab of the Project Management window.

**Access** **Settings | Project Types | Project Indicators > Risk: New**

### Use to

- Specify the risk types used when defining risks to an element

### How to

To add a new risk type to the global list

Step	Action	See also
1	Select the <b>Project Indicators</b> menu option. The Project Indicators dialog displays.	
2	Click on the Risk tab.	
3	Click on the <b>New</b> button. (To edit an existing risk type, click on the risk type name in the <b>Defined Risks</b> list.)	
4	Complete the fields as follows: <ul style="list-style-type: none"> <li>• In the <b>Risk Type</b> field type the name of the risk type</li> <li>• In the <b>Description</b> field type a short description of the risk type</li> <li>• In the <b>Weight</b> field type the default weighting to apply to the risk type</li> <li>• In the <b>Note</b> field, type any additional information on the risk type</li> </ul>	
5	Click on the <b>Save</b> button.	

### Notes

- Although Enterprise Architect does not currently provide detailed reports on risks within a model, you can use the Automation Interface or similar tools to create your own custom reports based on risk information you enter
- You can transport risk types between models, using Export Reference Data and Import Reference Data

### Learn more

- [The Project Management Window](#)<sup>[510]</sup>
- [Enterprise Architect Object Model](#)<sup>[2804]</sup>
- [Export Reference Data](#)<sup>[376]</sup>
- [Import Reference Data](#)<sup>[380]</sup>

## 3.8.3 The Project Status View

The Project Status view documents tasks and issues that relate directly to the current project, and lists the statistics of the project.

**Access** **Project | Project Status > <Tab>**

### Tabs

Tab	Detail	See also
<b>Tasks</b>	Review major <b>project tasks</b> that require attention.	<a href="#">Project Tasks</a> [526]
<b>Issues</b>	Review <b>events</b> , <b>occurrences</b> and <b>situations</b> that impact on project development and delivery. You can also generate a document report on these <b>issue</b> items.	<a href="#">Project Issues</a> [526]
<b>Project Statistics</b>	Review the <b>project statistics</b> , on the Project Statistics tab. This simply lists the numbers of packages, diagrams, elements, connectors, elements of each type and element features of each type in the project.  For certain items, such as element risks, metrics and effort, if there are no instances of that particular object type it is still listed but with a count of 0.	

#### Notes

- You can add or work on an item in the Project Status view by *right-clicking* (context menu) or *double-clicking* on the blank or completed item line
- The Project Status view context menu has options for filtering tasks and issues by status; you can also rearrange the sort-order by clicking in the title bar of the column that the items are to be indexed on

### 3.8.3.1 Project Tasks

The **Tasks** tab of the Project Status view provides a convenient 'To Do' list of major project work items that are not recorded elsewhere, and can be used to track events such as requests or corrections.

**Access**    **Project | Project Status > Tasks**

#### Notes

- Right-click on the list to view the context menu, and select to add, modify or delete tasks, or to set a status filter
- You can re-organize the display of the listed tasks using the List Header facilities for reported information
- To print out the currently displayed items, select the **Print List** context menu option
- You can transport task definitions between models using Export Reference Data and Import Reference Data

#### Learn more

- [Add, Modify and Delete Tasks](#) [527]
- [Export Reference Data](#) [376]
- [Import Reference Data](#) [380]
- [List Header](#) [677]

### 3.8.3.1.1 Add, Modify and Delete Tasks

This topic explains how to add, edit and delete project tasks.

**Access**    **Project | Project Status > Tasks**  
**Project | Calendar > Project Tasks mode**

#### Maintain Tasks

Step	Action	See also
1	<p>If creating a new task:</p> <ul style="list-style-type: none"> <li>Double-click in a blank area of the Tasks tab, or on a cell of the Calendar</li> </ul> <p>If editing an existing item:</p> <ul style="list-style-type: none"> <li>Double click on the item on the Tasks tab or Calendar</li> </ul> <p>The Task Detail dialog displays.</p>	<p><a href="#">The Project Status View</a> <sup>[525]</sup></p> <p><a href="#">Project Calendar</a> <sup>[571]</sup></p>
2	<p>Enter or update the following details of the task, selecting a value from the field's drop-down list where appropriate:</p> <ul style="list-style-type: none"> <li>The task name</li> <li>Automatic naming - if you have set up automatic naming conventions, click on the <b>Auto</b> button to insert the predefined element name and counter text; if you already have some text in the <b>Name</b> field, it is over-written by the automatic naming text</li> <li>The task type</li> <li>The task owner</li> <li>The expected start and end date for the task (select the check boxes to activate the dates)</li> <li>The current status of the task</li> <li>The person this task has been assigned to</li> <li>The task priority: high, medium or low</li> <li>The expected total time for the task and the actual time expended (in complete units; the type of unit must be agreed across the project)</li> <li>The percentage completion</li> <li>The phase associated with this task</li> <li>A description of the task</li> <li>Any progress history appropriate to the task</li> </ul>	<p><a href="#">Set Auto Naming and Auto Counters</a> <sup>[904]</sup></p>
3	Click on the <b>Apply</b> button.	

Step	Action	See also
4	To create another entry click on the <b>New</b> button or, to close, click on the <b>OK</b> button.	
5	To delete a task: <ul style="list-style-type: none"> <li>Right-click on the task and select the <b>Delete</b> context menu option</li> <li>Click on the <b>Yes</b> button on the confirmation prompt</li> </ul>	

#### Notes

- Owner** and **Assigned** fields are filled from the Project Authors, Resources and Project Clients

#### Learn more

- [Project Authors](#) <sup>[1153]</sup>
- [Resources](#) <sup>[1156]</sup>
- [Project Clients](#) <sup>[1157]</sup>
- [General Options](#) <sup>[605]</sup>

### 3.8.3.2 Project Issues

Any identified issues can be recorded against the current project, on either the:

- Issue Detail dialog from the Issues tab of the Project Status view, or
- Project Issues dialog

These two dialogs have very similar fields.

For each Issue, you record the description, date, owner and status.

**Access** **Project | Project Status > Issues** or  
**Project | Documentation | Issues**

Usage	See also
Add, edit and delete Issues	<a href="#">Add, Delete and Modify Issues</a> <sup>[529]</sup>
Generate and view a rich text format report of your issue list	<a href="#">Report From Project Issues Dialog</a> <sup>[531]</sup> <a href="#">Report From Project Issues Tab</a> <sup>[531]</sup> <a href="#">Report Output Sample</a> <sup>[532]</sup>



**Notes**

- You can transport these issue definitions between models, using **Export Reference Data** and **Import Reference Data**
- To print out the currently displayed items, select the **Print List** context menu option
- You can re-organize the display of the listed tasks using the List Header facilities for reported information
- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Issues** permission to update and delete Issues records

**Learn more**

- [Export Reference Data](#)<sup>[378]</sup>
- [Import Reference Data](#)<sup>[380]</sup>
- [List Header](#)<sup>[677]</sup>
- [List of Available Permissions](#)<sup>[329]</sup>

**3.8.3.2.1 Add, Delete and Modify Issues**

You can maintain project issue records using either the:


- Issue Detail dialog from the Issues tab of the Project Status view, or
- Project Issues dialog

These two dialogs have very similar fields.

**Access** **Project | Project Status > Issues** or  
**Project | Documentation | Issues**

**Maintain Project Issues**

Step	Action	See also
1	<p>If creating a new issue:</p> <ul style="list-style-type: none"> <li>• On the Issues tab of the Project Status view, double-click in a blank area to display the Issue Detail dialog, click on the <b>New</b> button</li> </ul> <p>or</p> <ul style="list-style-type: none"> <li>• On the Project Issues dialog, click on the <b>New</b> button</li> </ul> <p>If editing an existing item:</p> <ul style="list-style-type: none"> <li>• On the Issues tab of the Project Status view, double-click on an item to display the Issue Detail dialog</li> </ul> <p>or</p> <ul style="list-style-type: none"> <li>• On the Project Issues dialog, click on the item within the <b>Project Issues &amp; Discussion</b> list</li> </ul>	<p><a href="#">The Project Status View</a><sup>[625]</sup></p>

2	<p>Enter or update the following issue details, selecting a value from the field's drop-down list where appropriate:</p> <ul style="list-style-type: none"> <li>• The issue name</li> <li>• Automatic naming - if you have set up automatic naming conventions, click on the <b>Auto</b> button to insert the predefined element name and counter text; if you already have some text in the <b>Name</b> field, it is over-written by the automatic naming text</li> <li>• The issue priority</li> <li>• The date the issue was raised</li> <li>• The issue status</li> <li>• The issue owner</li> <li>• A description of the issue</li> <li>• The name of the person who resolved the issue</li> <li>• The date on which the issue was resolved (select the check box to activate the date)</li> <li>• Any comments on the resolution</li> </ul>	<a href="#">Set Auto Naming and Auto Counters</a> 
3	Click on the <b>Apply</b> or <b>Save</b> button.	
4	If the issue is closed (and all the <b>Resolution</b> fields are completed), click on the <b>Close Issue</b> button.	
5	To create another entry, click on the <b>New</b> button, or to close, click on the <b>OK</b> or <b>Close</b> (Project Issues dialog) button.	
6	<p>To delete an issue:</p> <ul style="list-style-type: none"> <li>• On the Issues tab, right-click on the entry and select the <b>Delete</b> context menu option, then click on the <b>Yes</b> button on the confirmation prompt</li> </ul> <p>or</p> <ul style="list-style-type: none"> <li>• On the Project Issues dialog, click on the item in the <b>Project Issues &amp; Discussion</b> list, ensure that the <b>Resolution</b> fields are complete, then click on the <b>Delete</b> button</li> </ul>	

### Notes

- You cannot delete a Closed issue through the Project Issues dialog
- You can filter the list of issues by status, to include or exclude:
  - On the Issues tab, all issues or just Open, Closed or Under Review issues, using the **Set Term Filter** context menu option
  - On the Project Issues dialog, closed issues, using the **Show Closed Issues** checkbox

### 3.8.3.2.2 Report From Project Issues Dialog

This topic explains how to generate a document report on your project issue records using the Project Issues dialog.

**Access** [Project | Documentation | Issues](#)

#### Use to

- Generate and view a document report on project issues

#### How to

To generate your project issues report

Step	Action	See also
1	On the Project Issues dialog, click on the <b>Report</b> button. The Save As dialog displays.	
2	Browse for and select the appropriate file location. In the <b>File name</b> field, type the file name for the report.	
3	Click on the <b>Save</b> button. A status message displays when the report has been generated.	
4	Click on the <b>OK</b> button and then on the <b>View RTF</b> button. The report displays in your default viewer.	<a href="#">Report Output Sample</a> <sup>[532]</sup>

### 3.8.3.2.3 Report From Project Issues Tab

This topic explains how to generate a document report on your project issue records using the Issues tab of the Project Status view.

**Access** [Project | Project Status > Issues | Create RTF Report](#)

#### Use to

- Generate and view a document report on project issues

#### Generate your project issues report

Step	Action	See also
1	Right-click on the Issues tab; select the <b>Create RTF Report</b> context menu option. The Save As dialog displays.	
2	Browse for and select the appropriate file location. In the <b>File name</b> field, type the file name for the report.	
3	Click on the <b>Save</b> button. A status message displays when the report has been generated.	
4	Click on the <b>OK</b> button and then on the <b>View RTF</b> button. The report displays in your default viewer.	<a href="#">Report Output Sample</a> <sup>[532]</sup>

#### 3.8.3.2.4 Report Output Sample

The following illustration is an example of the output from an *Issues* report.

### List of Project Issues: 24-Jul-2010 9:47:00 AM

Issue	Date/Owner	Description	Resolution
Test servers will be delayed	24/07/2010 Eloise Norman	The test server builds have been delayed because the particular (unusual) memory requirements to match the customer's site are not available on shore. They are being sourced from Singapore but it will delay the builds and delivery of the machines.	Closed: 24/07/2010 Geoffrey Sparks  The machines will be built and delivered using standard memory and the proprietary memory will be added later. All performance tests will be delayed until the memory is available.
Public Holidays	24/07/2010 Joanna Stoa	The schedule includes staff working on public holidays. A number of staff have indicated that contrary to what they stated earlier they are not available.	Open: 24/07/2010
Compiler Version disparity	24/07/2010 Eloise Norman	A number of the developers have downloaded different versions of a number of the compilers. This has lead to	Under Review: 24/07/2010

		unpredictable builds impacting on testing.	
--	--	--	--

### 3.8.4 Project Glossary

The Project Glossary enables you to set up a list of defined terms for your project, review the technical and business terms already defined for a model, add to the list, delete or change items, and filter the list to exclude by type.

**Access** **Project | Glossary** ( Alt+2 )

Usage	See also
Add, delete and modify the project glossary entries through either the: <ul style="list-style-type: none"> <li>Glossary view, or</li> <li>Glossary dialog</li> </ul>	<a href="#">Project Glossary View</a> <sup>[534]</sup> <a href="#">The Glossary Dialog</a> <sup>[535]</sup>
Separate glossary items by category; for example, Business terms and Technical terms.	<a href="#">Project Glossary View</a> <sup>[534]</sup> <a href="#">The Glossary Dialog</a> <sup>[535]</sup>
Save the glossary in Rich Text format for inclusion as part of a larger project document.	<a href="#">Generate a Report</a> <sup>[536]</sup>
Create glossary terms and definitions from text in the Notes window, or from any <b>Notes</b> or <b>Description</b> fields that have the Notes toolbar.	<a href="#">Notes</a> <sup>[1142]</sup>
Insert existing glossary terms into any other <b>Notes</b> fields.	<a href="#">Notes</a> <sup>[1142]</sup>

#### Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Glossary** permission in order to create, update or delete Glossary items; if security is not enabled, you can manage Glossary items without any permissions
- You might include a Glossary Report in your project requirements or functional specifications documents
- You can transport glossary definitions between models, using Export Reference Data and Import Reference Data

#### Learn more

- [Export Reference Data](#) <sup>[376]</sup>
- [Import Reference Data](#) <sup>[380]</sup>

### 3.8.4.1 Project Glossary View


The Project Glossary view displays the glossary terms already defined for your model. You can use this View to:

- Add glossary terms
- Modify glossary terms
- Change a definition type for all terms of that type
- Filter the display show terms of a specific type only
- Print the displayed list of terms
- Delete glossary terms

**Access** **Project | Glossary** ( Alt+2 )

Right-click on an entry on the Glossary view, and select the appropriate context menu option for the operation you intend to perform (as listed above); if you select the:

- **Add New** or **Modify Selected** options, the Glossary Detail dialog displays; complete the fields as indicated:

Field	Action	See also
<b>Term</b>	(Mandatory) Type the term to include in the glossary.	
<b>Type</b>	(Mandatory) Select the required type.  If you require a different glossary type, click on the ( ... ) (browse) button and specify the name of the new type.  This field applies the type <i>only</i> to the selected term; you can rename a type for <i>all</i> terms of that type, using the context menu as described below.	
<b>Meaning</b>	Type the definition or description of the term.  If necessary, format the text of this description using the Notes toolbar at the top of the field.	<a href="#">Notes</a> 
<b>New</b>	Clear the dialog fields so that you can define a new Glossary term.	
<b>Apply</b>	Save the new or updated glossary entry.	
<b>Delete</b>	Delete the entry.  A prompt displays to confirm the deletion; click on the <b>Yes</b> button to remove the term from the glossary.	

- **Rename type** option, the Rename Glossary Type dialog displays on which you enter the alternative type name (either one of the existing types or a new type); when you click on **OK**, *all* entries of the

original type are changed to the new type

- **Set term filter** option, the Term Type Filter dialog displays on which you enter the type of glossary term to list; when you click on **OK**, only items of that type are shown on the Glossary view
- **Remove term filter** option, the Glossary view refreshes to show glossary items of all types
- **Print List** option, the Print dialog displays on which you define the printing parameters; when you click on **OK**, the currently-displayed list prints out
- **Delete** option, a prompt displays to confirm the deletion; click on the **Yes** button to remove the term from the glossary

#### Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Glossary** permission in order to create, update or delete Glossary items; if security is not enabled, you can manage Glossary items without any permissions

### 3.8.4.2 The Glossary Dialog

Using the Glossary dialog, you can create, maintain and delete definitions of terms used in your project, as a Glossary. When reviewing the Glossary terms, you can filter the list to display terms of a specific type.

**Access** [Project | Documentation | Glossary](#)

#### Maintain Glossary terms

Field	Usage	See also
<b>Glossary Term</b>	Type the term to include in the glossary.	
<b>Glossary Type</b>	(Mandatory) Select the required type.  If you require a different glossary type, click on the ( ... ) (browse) button and specify the name of the new type.  This field applies the type <i>only</i> to the selected term; you can <i>rename</i> a type for all terms of that type <sup>[534]</sup> , using the Glossary view.	<a href="#">Project Glossary View</a> <sup>[534]</sup>
<b>Description</b>	(Mandatory) Type the definition or description of the term.  If necessary, format the text of this description using the Notes toolbar at the top of the field.	<a href="#">Notes</a> <sup>[1143]</sup>
<b>Limit Display To</b>	Select the appropriate glossary type to filter the <b>Type   Term</b> list to show entries of that type only.  Select <b>All</b> to display all glossary entries.	
<b>New</b>	Clear the dialog fields so that you can define a new Glossary term.	
<b>Type   Term</b>	Review the list of defined glossary terms; click on an entry to edit it in	

Field	Usage	See also
	the dialog fields.	
<b>Save</b>	Save the new or updated glossary entry.	
<b>Delete</b>	Delete the entry selected from the <b>Type   Term</b> list.	
<b>Report</b>	Generate and print a glossary report.	<a href="#">Generate a Report</a> <a href="#">Glossary Report Output Sample</a>

#### Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Glossary** permission in order to create, update or delete Glossary items; if security is not enabled, you can manage Glossary items without any permissions

### 3.8.4.3 Generate a Report

You can generate a rich text format file of your model's glossary, which you can either print or attach to any other appropriate document file.

The file can contain all glossary entries, or just those of selected types.

Access **Project | Documentation | Glossary: Report**

#### Use to

- Generate a formatted RTF document file of your model glossary

#### Generate an RTF report of the project Glossary

Step	Action	See also
<b>1</b>	Select the <b>Glossary</b> menu option. The Glossary dialog displays.	
<b>2</b>	Click on the <b>Report</b> button. The Glossary Report dialog displays.	
<b>3</b>	In the <b>Filename</b> field, type or select a filename for the glossary.	



Step	Action	See also
4	In the <b>Heading</b> field, type a suitable heading for the glossary.	
5	In the <b>Include Glossary Items</b> panel, select the checkbox for each type of glossary entry to include. To select all types of entry, click on the <b>Select All</b> button.	
6	If necessary, to define the page setup and language for the report, click on the <b>Page Setup</b> and/or <b>Language</b> buttons.	
7	To include page breaks, select the <b>Page break between sections</b> checkbox.	
8	Click on the <b>Generate</b> button to generate the report.	
9	Click on the <b>View</b> button to open the report.	<a href="#">Glossary Report Output Sample</a> 537

#### 3.8.4.3.1 Glossary Report Output Sample

An example of the output from a Glossary report is shown below:

## Glossary

### **Business Terms**

#### *Accounting Periods*

A defined period of time whereby performance reports can be extracted. (normally 4 week periods).

#### *Customer*

A person or a company that requests An entity to transport goods on their behalf.

### **Technical Terms**

#### *Association*

A relationship between two or more entities. Implies a connection of some type - for example one entity uses the services of another, or one entity is connected to another over a network link.

#### *Component Model*

The component model provides a detailed view of the various hardware and software

components that make up the proposed system. It shows both where these components reside and how they inter-relate with other components. Component requirements detail what responsibilities a component has to supply functionality or behavior within the system.

#### ***Deployment Model***

A model of the system as it is physically deployed.

#### ***Extends Relationship***

A relationship between two Use Cases in which one Use Case 'extends' the behavior of another. Typically this represents optional behavior in a Use Case scenario - for example a user might optionally request a list or report at some point in a performing a business Use Case.

### **3.8.5 Project Task Allocation**

Using the Project Gantt View, you can review the allocation of work to elements in the project, focussing on either the elements that require work, or the resources required to perform the work. The window primarily shows information that is entered through other windows and dialogs, but once a record exists in the window you can edit it and, for example, add to or change the resources on an element.

**Access** [Project | Project Gantt View](#)

#### **Project Task Views**

View	Detail	See also
<b>Resource View</b>	The Resource View shows the current commitments of each allocated resource in the project as a list of allocation records and a Gantt chart of the progress of the allocations.	<a href="#">Resource View</a> <sup>[538]</sup>
<b>Element View</b>	The Element View lists the elements in the project that have resources assigned to them, and the resources assigned to each element, as a list of allocation records and a Gantt chart of the progress of the allocations.	<a href="#">Element View</a> <sup>[542]</sup>
<b>Report View</b>	A Resource report shows how your resources are deployed in your project, displaying a list of all elements that have resources allocated to them.	<a href="#">Report View</a> <sup>[546]</sup>

#### **Learn more**

- [The Gantt View](#) <sup>[594]</sup>

#### **3.8.5.1 Resource View**

The Resource view shows the current commitments of each allocated resource in the project as a list of allocation records and a Gantt chart of the progress of the allocations.

The display initially shows the resources and their overall commitment; click on the 'plus-box' to the left of the name of the resource to expand the entry to show the elements and the allocation period for each element.


The display shows both complete allocations and those that are still in progress; an internal filter hides completed allocations two weeks after the end date, and incomplete allocations one month after the end

date. Tasks you can perform include checking the:

- Dates on which specific resources or all resources are currently allocated
- Elements to which each resource is allocated
- Dates on which a resource is allocated to work on a specific element
- General progress of the resource in completing the work
- Specific details of the allocation of a resource to an element and task or role

**Access** **Project | Project Gantt View > Resource View**

#### Resource View options

Option	Action	See also
<b>Filter the display to include elements having certain properties</b>	<p>Right-click on the list and select the <b>Show Filter Bar</b> context menu option.</p> <p>The Filter Bar displays underneath the heading bar. Type a text string in the field above a column to immediately filter the list for entries that have the text string in the values in that column.</p> <p>To delete a filter string altogether, click on the blue cross at the right of the field.</p> <p>If you no longer want to use the filter bar, right click on the list and select the <b>Hide Filter Bar</b> option.</p>	<a href="#">List Header</a> 
<b>Filter the display to exclude elements by status</b>	<p>(A feature of the Project Gantt View only, taking effect under your user ID.)</p> <p>Right-click on the window and select the <b>Apply Element Status Filter</b> context menu option.</p> <p>The Excluded Status Types dialog displays; select the checkbox against one or more statuses to exclude elements having those statuses from the list.</p> <p>You can select all checkboxes at once by clicking on the <b>Select All</b> button, and clear all selections by clicking on the <b>Clear All</b> button.</p> <p>Click on the <b>OK</b> button to immediately apply the filter, which stays in effect until you specifically change it.</p>	
<b>Filter the display by Start or Completion date</b>	<p>The display default is to show current tasks for which the end date has not yet occurred. Right-click on the list and select one of the context menu options:</p> <ul style="list-style-type: none"> <li>• <b>Include Completed Tasks Within the Last...</b> - to display incomplete tasks <b>and</b> tasks completed only within the last period; you can set this period to <b>7</b>, <b>30</b>, or <b>90</b> days, or you can include <b>all</b> completed tasks, or <b>hide</b> all completed tasks</li> <li>• <b>Include Future Tasks Starting in...</b> - to display current incomplete tasks and completed tasks (depending on the setting of the option above) <b>and future</b> tasks that have been</li> </ul>	

Option	Action	See also
	recorded and are due to start within the next period; you can set this period to <b>7</b> , <b>30</b> or <b>90</b> days, or you can show <b>all</b> tasks that have been recorded but are not yet due to start	
<b>Identify overdue tasks amongst the items</b>	Right-click on the display and select the <b>Display Highlight For Overdue items</b> option and one of its sub-options: <ul style="list-style-type: none"> <li>• <b>Show in Red</b> - to display the uncompleted percentage of the task bars for overdue items on the Gantt chart in red</li> <li>• <b>Show in Red to Current Date</b> - to extend the task bars of the overdue items to today's date, and display them in red</li> <li>• <b>None</b> - to cancel any overdue item highlighting that has been set</li> </ul>	
<b>Display ONLY overdue tasks</b>	Right-click on the display and select the <b>Show Overdue Items Only</b> option.  The display shows only those items for which the end date has passed but that are not 100% complete. These items do not have red highlighting.	
<b>Display tasks for today only, or for another day only</b>	Right-click on the display and select the option: <ul style="list-style-type: none"> <li>• <b>Show only Active tasks for today</b> - to show only tasks that are in progress today</li> <li>• <b>Show only Active tasks for another day</b> - to show only tasks that were in progress on a specific day in the past, or that are scheduled to be in progress on a day in the future; a calendar dialog displays from which you select the day to examine</li> </ul>	
<b>Display the properties of the element to which the resource is assigned</b>	Right-click on the entry and select the <b>Show Element Properties</b> context menu option.  The Properties dialog for the element displays.	<a href="#">Properties Dialog</a> <sup>[956]</sup>
<b>Display the resource allocation details</b>	Right-click on the entry and select the <b>Show Task Properties</b> context menu option.  The Assigned Resources dialog displays; you can edit the details and, if necessary, change the resource allocated to the element.	<a href="#">Resource Allocation</a> <sup>[512]</sup>  <a href="#">The Project Management Window</a> <sup>[510]</sup>
<b>Refresh the report</b>	Right-click on the entry and select the <b>Refresh</b> context menu option.  The content of the display is refreshed and collapsed to resource level.	
<b>Display the</b>	Right-click on the entry and select either of the context menu	

Option	Action	See also
<b>Resource Allocation records for the element</b>	<p>options:</p> <ul style="list-style-type: none"> <li>• <b>Show Element Project Management Window</b> (if the window is closed or hidden)</li> <li>• <b>Find Task in Element Project Management Window</b> (if the window is visible but showing the details of another element)</li> </ul> <p>The Project Management window displays at the Resource Allocation tab, with the details of the selected entry shown in the fields and the other resource allocations for the element listed in the left-hand panel.</p> <p>You can edit the details and, if necessary, change the resource allocated to the element.</p>	<p><a href="#">The Project Management Window</a> <sup>[510]</sup></p> <p><a href="#">Resource Allocation</a> <sup>[512]</sup></p> <p><a href="#">Assign Multiple Resources</a> <sup>[515]</sup></p>
<b>Assign a new resource to an element</b>	<p>Right-click on the appropriate element in an expanded resource item, and select the <b>Assign Resource</b> context menu option.</p> <p>The Assigned Resources dialog displays; complete this as for the Resource Allocation tab of the Project Management window.</p>	<p><a href="#">Resource Allocation</a> <sup>[512]</sup></p> <p><a href="#">The Project Management Window</a> <sup>[510]</sup></p>
<b>Reposition the Gantt chart to automatically show the end date of a selected allocation</b>	<p>Right-click on the display and select the <b>Go to   Auto Sync with Task End Date</b> option.</p> <p>Whilst this option is selected (with a tick next to it), whenever you click on an allocation in the Resource view the display adjusts to show the end date of the task in the center of the chart.</p>	
<b>Reposition the Gantt chart to show the start date or end date of an allocation, or today's date</b>	<p>Right-click on the entry and select the required option:</p> <ul style="list-style-type: none"> <li>• <b>Go To   Task Start Date</b></li> <li>• <b>Go To   Task End Date</b></li> <li>• <b>Go To   Today's Date</b></li> </ul> <p>The display shifts to put the required date in the <b>center</b> of the chart.</p>	
<b>Locate the element in the Project Browser</b>	<p>Right-click on the element name and select the <b>Find in Project Browser</b> context menu option.</p> <p>The area of the Project Browser containing the element is brought into focus and expanded, and the element is highlighted.</p>	
<b>Save an image of the Gantt chart to file</b>	<p>Right-click on the tab and select the <b>Save Image to File</b> context menu. option</p> <p>The Save As Image dialog displays, on which you specify the file name, location and graphics file type to save to.</p>	
<b>Save an image</b>	Right-click on the tab and select the <b>Copy Image to Clipboard</b>	

Option	Action	See also
<b>of the Gantt chart to the clipboard</b>	context menu option.  You can paste the image from the clipboard into your preferred graphics package.	

#### Learn more

- [The Gantt View](#)<sup>[594]</sup>

### 3.8.5.2 Element View

The Element View of the Project Gantt View lists the elements in the project that have resources assigned to them. It shows the resources assigned to each element as a list of allocation records and as a Gantt chart of the progress of the allocations. An allocation can be partial, in that a role or task is assigned but no specific resource has been identified.

The display initially shows the elements and their overall resource commitment; click on the 'plus-box' to the left of the name of the element to expand the entry to show the resources and the allocation period for each resource. You can then:

- Check the dates on which specific elements have resources currently allocated
- Check which resources are currently allocated to an element
- Assign further resources to the element
- Check the dates on which a resource is allocated to work on a specific element
- Check the general progress of the resource in completing the work
- Check the specific details of the allocation of a resource to an element and task or role

The display shows both complete allocations and those that are still in progress; an internal filter hides completed allocations two weeks after the end date.

The Element View provides most of the same features as the **diagram Gantt View** and the **Package Browser Gantt View**, and complements the Resource View and Personal Tasks View of work allocation on the project.

**Access** **Project | Project Gantt View > Element View**  
**(Right-click on a diagram | Switch to Gantt View)**  
**(Project Browser | right-click Package | Package Browser | Gantt View)**

#### Options

Option	Action	See also
<b>Filter the display to exclude elements by status</b>	(A feature of the Project Gantt View only, taking effect under your user ID.)  Right-click on the window and select the <b>Apply Element Status Filter</b> context menu option.  The Excluded Status Types dialog displays; select the checkbox against one or more statuses to exclude elements having those statuses from the list.	

Option	Action	See also
	<p>You can select all checkboxes at once by clicking on the <b>Select All</b> button, and clear all selections by clicking on the <b>Clear All</b> button.</p> <p>Click on the <b>OK</b> button to immediately apply the filter, which stays in effect until you specifically change it.</p>	
<b>Filter the display to include elements having certain properties</b>	<p>Right-click on the display and select the <b>Show Filter Bar</b> option.</p> <p>The Filter Bar displays underneath the heading bar. Type a text string in the field above a column to immediately filter the list for entries that have the text string in the values in that column.</p> <p>To delete a filter string altogether, click on the blue cross at the right of the field.</p> <p>If you no longer want to use the filter bar, right click on the list and select the <b>Hide Filter Bar</b> option.</p>	<a href="#">List Header</a> <sup>677</sup>
<b>Identify overdue tasks</b>	<p>Right-click on the display and select the <b>Display Highlight For Overdue items</b> option and one of its sub-options:</p> <ul style="list-style-type: none"> <li>• <b>Show in Red</b> - to display the uncompleted percentage of the task bars for overdue items on the Gantt chart in red</li> <li>• <b>Show in Red to Current Date</b> - to extend the task bars of the overdue items to today's date, and display them in red</li> <li>• <b>None</b> - to cancel any overdue item highlighting that has been set</li> </ul>	
<b>Display ONLY overdue items</b>	<p>Right-click on the display and select the <b>Show Overdue Items Only</b> option.</p> <p>The display shows only those items for which the end date has passed but that are not 100% complete. These items do not have red highlighting.</p>	
<b>Filter the display by Start or Completion date</b>	<p>The display default is to show current tasks for which the end date has not yet occurred. Right-click on the list and select one of the context menu options:</p> <ul style="list-style-type: none"> <li>• <b>Include Completed Tasks Within the Last...</b> - to display incomplete tasks <b>and</b> tasks completed only within the last period; you can set this period to <b>7</b>, <b>30</b>, or <b>90</b> days, or you can include <b>all</b> completed tasks, or <b>hide</b> all completed tasks</li> <li>• <b>Include Future Tasks Starting in...</b> - to display current incomplete tasks and completed tasks (depending on the setting of the option above) <b>and</b> future tasks that have been recorded and are due to start within the next period; you can set this period to <b>7</b>, <b>30</b> or <b>90</b> days, or you can show <b>all</b> tasks that have been recorded but are not yet due to start</li> </ul>	
<b>Display tasks for</b>	Right-click on the display and select the option:	

Option	Action	See also
today only, or for another day only	<ul style="list-style-type: none"> <li>• <b>Show only Active tasks for today</b> - to show only tasks that are in progress today</li> <li>• <b>Show only Active tasks for another day</b> - to show only tasks that were in progress on a specific day in the past, or that are scheduled to be in progress on a day in the future; a calendar dialog displays from which you select the day to examine</li> </ul>	
Display the properties of an element	<p>Right-click on the element and select the <b>Show Element Properties</b> context menu option.</p> <p>The Properties dialog for the element displays.</p>	<a href="#">Properties Dialog</a> <sup>[956]</sup>
Display the resource allocation details for an element	<p>Right-click on the resource and select the <b>Show Task Properties</b> context menu option.</p> <p>The Assigned Resources dialog displays; you can edit the details and, if necessary, change the resource allocated to the element.</p>	<a href="#">Resource Allocation</a> <sup>[512]</sup> <a href="#">The Project Management Window</a> <sup>[510]</sup>
Assign a new resource to the element	<p>Right-click on the element and select the <b>Assign Resource</b> context menu option.</p> <p>The Assigned Resources dialog displays; complete this as for the Resource Allocation tab of the Project Management window.</p>	<a href="#">Resource Allocation</a> <sup>[512]</sup> <a href="#">The Project Management Window</a> <sup>[510]</sup>
Refresh the report	<p>Right-click on the display and select the <b>Refresh</b> context menu option.</p> <p>The content of the display is refreshed and collapsed to element level.</p>	
Display the Resource Allocation records for an element	<p>Right-click on the entry and select either of the context menu options:</p> <ul style="list-style-type: none"> <li>• <b>Show Element Project Management Window</b> (if the window is closed or hidden)</li> <li>• <b>Find Task in Element Project Management Window</b> (if the window is visible but showing the details of another element)</li> </ul> <p>The Project Management window displays at the Resource Allocation tab, with the details of the selected entry shown in the fields and the other resource allocations for the element listed in the left-hand panel.</p> <p>You can edit the details and, if necessary, change the resource allocated to the element.</p>	<a href="#">Resource Allocation</a> <sup>[512]</sup> <a href="#">The Project Management Window</a> <sup>[510]</sup>
Reposition the Gantt chart to	<p>Right-click on the display and select the <b>Go to   Auto Sync with Task End Date</b> option.</p>	



Option	Action	See also
<b>automatically show the end date of a selected allocation</b>	Whilst this option is selected (with a tick next to it), whenever you click on an allocation in the Element view the display adjusts to show the end date of the task in the center of the chart.	
<b>Reposition the Gantt chart to show the start date or end date of an allocation, or today's date</b>	<p>Right-click on the entry and select the required option:</p> <ul style="list-style-type: none"> <li>• <b>Go To   Task Start Date</b></li> <li>• <b>Go To   Task End Date</b></li> <li>• <b>Go To   Today's Date</b></li> </ul> <p>The display shifts to put the required date in the center of the chart.</p>	
<b>Locate the element in the Project Browser</b>	<p>Right-click on the element name and select the <b>Find in Project Browser</b> context menu option.</p> <p>The area of the Project Browser containing the element is brought into focus and expanded, and the element is highlighted.</p>	
<b>Save an image of the Gantt chart to file</b>	<p>Right-click on the list and select the <b>Save Image to File</b> context menu option.</p> <p>The Save As Image dialog displays, on which you specify the file name, location and graphics file type to save to.</p>	
<b>Save an image of the Gantt chart to the clipboard</b>	<p>Right-click on the tab and select the <b>Copy Image to Clipboard</b> context menu option.</p> <p>You can paste the image from the clipboard into your preferred graphics package.</p>	

#### Learn more

- [The Gantt View](#)<sup>[594]</sup>
- [Resource View](#)<sup>[538]</sup>
- [Monitor Your Tasks](#)<sup>[559]</sup>
- [Package Browser](#)<sup>[673]</sup>
- [Diagram View](#)<sup>[784]</sup>

### 3.8.5.3 Report View

The *Report View* shows how your resources are deployed in your project, displaying:

- A list of all elements that have resources allocated to them, and the type of each element
- The resource allocated, and the role played by that resource
- The start and end dates of the allocation
- The time allocated, expected and expended
- The percentage completion of the allocation


You can tailor the displayed information by:

- Hiding columns of information
- Grouping types of information
- Filtering the data by status
- Filtering the data by start date or end date
- Filtering any column to show only a specific value

Having displayed the information you require, you can print it.

**Access** [Project | Project Gantt View > Report View](#)

#### Report View Options

Option	Action	See also
<b>Run the report</b>	Click on the first icon in the Report View toolbar (the rotating arrows). The report results display. If you have the report open for a while, you can update the information; either: <ul style="list-style-type: none"> <li>• Run the report again or</li> <li>• Right-click on the content and select the <b>Refresh</b> context menu option</li> </ul>	
<b>Adjust column headings</b>	Drag and drop column headings into the sequence you require. Right-click on the column headings and select the <b>Field Chooser</b> context menu option, to add or remove columns using the Field Chooser dialog.	
<b>Group entries by column heading</b>	Right-click on the column headings and select the <b>Enable Group Box</b> context menu option, to cluster the report information according to your preferred hierarchy of column headings.	<a href="#">List Header</a> 
<b>Filter columns</b>	Either: <ul style="list-style-type: none"> <li>• Click on the third toolbar icon from the <i>right</i> (the spy glass),</li> </ul>	

Option	Action	See also
	<p>or</p> <ul style="list-style-type: none"> <li>Right-click on the column headings and select the <b>Toggle Filter Bar</b> context menu option</li> </ul> <p>The filter field displays at the top of every column.</p> <p>Type in whatever text string you require in the appropriate column to filter the report to show only entries containing that text string in that column.</p>	
<b>Filter by degree of completion</b>	<p>In the first field in the toolbar, click on the drop-down arrow and select one of the following values:</p> <ul style="list-style-type: none"> <li><b>All</b> - Display all entries regardless of degree of completion</li> <li><b>Completed</b> - Display only those entries where the allocation is 100% completed</li> <li><b>Above cut-off</b> - Display only those entries that are more than a certain percentage complete</li> <li><b>Below cut-off</b> - Display only those entries that are less than a certain percentage complete</li> </ul> <p>In the second field in the toolbar, either type a threshold value or increment to the value using the up/down arrows, to set the percentage completion for the <b>Above/Below cut-off</b> options.</p>	
<b>Filter according to start/end date</b>	<p>Click on the fourth toolbar icon from the right (the funnel).</p> <p>The Resource Filters dialog displays.</p> <p>In the <b>Start Date</b> and/or <b>End Date</b> field, click on the drop-down arrow and select the appropriate qualifier:</p> <ul style="list-style-type: none"> <li><b>After</b></li> <li><b>Before</b></li> <li><b>Equal To</b></li> <li><b>Not Equals</b></li> </ul> <p>In the date fields, click on the checkbox to activate the fields and either type in the day, month and year or click on the drop-down arrow to select the date from a calendar.</p> <p>The fields have an AND relationship; an entry must satisfy both date criteria before it is displayed.</p>	
<b>Display the properties of the element to which the resource is assigned</b>	<p>Right-click on the entry and select the <b>Show Element Properties</b> context menu option.</p> <p>The Properties dialog for the element displays.</p>	<a href="#">Properties Dialog</a> <sup>[956]</sup>

Option	Action	See also
<b>Display the resource allocation details</b>	<p>Right-click on the entry and select the <b>Show Task Properties</b> context menu option.</p> <p>The Assigned Resources dialog displays.</p>	<p><a href="#">Resource Allocation</a><sup>[512]</sup></p> <p><a href="#">The Project Management Window</a><sup>[510]</sup></p>
<b>Display the Resource Allocation records for the element</b>	<p>Right-click on the entry and select either of the context menu options:</p> <ul style="list-style-type: none"> <li>• <b>Show Element Project Management Window</b> (if the window is closed or hidden)</li> <li>• <b>Find Task in Element Project Management Window</b> (if the window is visible but showing the details of another element)</li> </ul> <p>The Project Management window displays at the Resource Allocation tab, with the details of the selected entry shown in the fields and the other resource allocations for the element listed in the left-hand panel.</p> <p>You can edit the details and, if necessary, change the resource allocated to the element.</p>	<p><a href="#">Resource Allocation</a><sup>[512]</sup></p> <p><a href="#">The Project Management Window</a><sup>[510]</sup></p>
<b>Locate the element in the Project Browser</b>	<p>Right-click on the entry and select the <b>Find in Project Browser</b> context menu option.</p> <p>The area of the Project Browser containing the element is brought into focus and expanded, and the element is highlighted.</p>	
<b>Print the report</b>	<p>Either:</p> <ul style="list-style-type: none"> <li>• Click on the second toolbar icon from the right (the printer) or</li> <li>• Right-click on an entry in the report and select the <b>Print</b> context menu option</li> </ul> <p>The Print dialog displays, on which you specify the printer to use and the characteristics of the print job.</p>	

#### Learn more

- [The Gantt View](#)<sup>[594]</sup>
- [Project Task Allocation](#)<sup>[538]</sup>
- [Resource View](#)<sup>[538]</sup>
- [Element View](#)<sup>[542]</sup>

### 3.8.6 Spell Checking



Enterprise Architect provides a powerful spell checking facility. This operates at both project level and Package level.

#### Topics

Topic	Detail	See also
<b>Automatic Spell Checking</b>	The spell checker can be set to run automatically, so that it highlights possible errors in text as it is created or pasted in: <ul style="list-style-type: none"> <li>• Select <b>Tools   Options &gt; Objects</b></li> <li>• Deselect or select the <b>Disable spelling</b> checkbox</li> </ul>	
<b>Spelling Configuration</b>	You can configure what types of text error the spell checker should detect, using the Spelling Options dialog.	<a href="#">Select Spell Checker Options</a> [550]
<b>Using Languages Other Than English</b>	Enterprise Architect is supplied with dictionaries for US English, Canadian English and British English; you can download a set of dictionaries for other languages from the Sparx Systems website.	<a href="#">Use Languages Other Than English</a> [552]
<b>Using the Spell Checker</b>	You can run the spell checker manually on the whole model, or on a selected Package.	<a href="#">Using the Spell Checker</a> [553]
<b>Correcting Text</b>	As the spell check progresses, Enterprise Architect highlights any errors or unknown words; you have several options for responding to these potential spelling errors.	<a href="#">Correcting Words</a> [554]

#### Notes

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Spell Check** permission to perform any spell check or change any spell check options

#### Learn more

- [List of Available Permissions](#) [329]

### 3.8.6.1 Select Spell Checker Options

You can configure the spell checker to ignore or detect specific types of text conformation. Ignoring the conformation means to skip over the text without identifying it as an error.

**Access** [Tools](#) | [Spelling Tools](#) | [Spelling Options](#)

#### Configure the spell checker

Option	Action	See also
<b>Ignore capitalized words</b>	Ignore any words beginning with a capital letter, such as <i>David</i> . You might enable this option if the text being checked contains many proper names.	
<b>Ignore all-caps words</b>	Ignore any words containing all capital letters, such as <i>RADAR</i> . You might enable this option if the text being checked contains many acronyms.	
<b>Ignore Words with Numbers</b>	Ignore any words containing embedded digits, such as <i>Win95</i> and <i>Q4</i> . You might enable this option if the text being checked contains many code words or other symbols containing digits.	
<b>Ignore Words with Mixed Case</b>	Ignore any words containing an unusual mixture of upper- and lower-case letters, such as <i>MicroHouse</i> and <i>CapsLock</i> . You might enable this option if the text being checked contains many variable names or other symbols that use changes in letter-case to distinguish words.	
<b>Ignore Domain Names</b>	Ignore any words that appear to be Internet domain names, such as <i>wintertree-software.com</i> .	
<b>Report Doubled Words</b>	Detect any word appearing twice in a row, such as <i>the the</i> .	
<b>Case Sensitive</b>	Distinguish between capitalized and non-capitalized words. For example, <i>canada</i> would be considered as different from <i>Canada</i> , and therefore reported as a misspelling. When the option is <b>disabled</b> , <i>canada</i> and <i>Canada</i> are considered to be identical. Note that the performance of the spell checker is reduced if this option is disabled.	
<b>Phonetic</b>	Suggest alternative words based on phonetic (sounds-like)	

Option	Action	See also
<b>Suggestions</b>	<p>similarity to the misspelled word; this option tends to improve the correction of badly misspelled words.</p> <p>Enabling this option increases the time required to locate suggestions.</p> <p>Note that either this option or the <b>Typographical Suggestions</b> option must be enabled in order to list suggestions of alternative words.</p>	
<b>Typographical Suggestions</b>	<p>Suggest alternative words based on typographical (looks-like) similarity to the misspelled word. This option is appropriate for people who are generally good spellers.</p> <p>Note that either this option or the <b>Phonetic Suggestions</b> option must be enabled in order to list suggestions of alternative words.</p>	
<b>Suggest Split Words</b>	<p>Suggest two separate words as a replacement for a misspelling containing two joined words.</p> <p>For example, <i>is the</i> would be suggested as a replacement for <i>isthe</i>.</p>	
<b>Auto Correct</b>	<p>Automatically change words marked with <i>Auto Change</i> actions to their specified replacements.</p> <p>When this option is <b>disabled</b>, a confirmation prompt displays before the words are changed.</p>	
<b>Main Dictionary Language</b>	<p>Set the language of the main dictionary used to check spelling; the drop-down list shows only languages for which dictionaries are installed on your system.</p> <p>To check spelling in a different language, select the language in the list.</p>	<a href="#">Use Languages Other Than English</a> <sup>[552]</sup>
<b>Suggestions</b>	<p>Determine the speed and accuracy of the initial automatic search for suggested replacements for misspelled words.</p> <p>Click on the appropriate radio button for your choice.</p>	
<b>OK</b>	Close the <b>Options</b> dialog, and save any changes made to the option settings.	
<b>Cancel</b>	Close the <b>Options</b> dialog, and discard any changes made to the option settings.	

### Notes

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Spell Check** permission to perform any

spell check or change any spell check options

#### Learn more

- [List of Available Permissions](#) <sup>[329]</sup>

### 3.8.6.2 Use Languages Other Than English

Enterprise Architect is supplied with three dictionaries, for US English, Canadian English and British English. Additional dictionaries are available as a pack, for download from the registered pages of the Sparx Systems website.

The dictionaries in this downloadable file enable spell checking in the following languages:

- Brazilian
- Danish
- Dutch
- Finnish
- French (Continental)
- German
- Italian
- Norwegian
- Portuguese
- Spanish and
- Swedish

#### Download the additional language dictionary pack

Step	Action	See also
1	Access the registered user page on the Sparx Systems web site.	<a href="#">Registered User page</a>
2	At the end of the page, download the <i>EADict.zip</i> file from the Enterprise Architect Dictionary section.	
3	Unzip the file into the Enterprise Architect install directory - <i>C:\Program Files\Sparx Systems\EA</i> .  This makes the non-English spelling dictionaries available to the Enterprise Architect spell checker.	<a href="#">Select Spell Checker Options</a> <sup>[550]</sup>

#### Notes

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Spell Check** permission to perform any spell check or change any spell check options



- Enterprise Architect .EAP files default to use Jet 3.5 as the database engine; this does not support unicode character sets, which makes the use of certain languages - such as Polish - difficult, including spell checking words in those languages

If you use such languages, it is recommended that you either upsize to a DBMS repository or set **JET 4.0** as the database engine

If your .EAP project is not in a Jet 4.0 database, you should also download a copy of the Jet 4.0 EABase model from the Sparx Systems website, and do an **EAP to EAP** transfer of your model into the Jet 4.0 file

#### Learn more

- [Server Based Repositories](#) <sup>[214]</sup>
- [General Options](#) <sup>[605]</sup>
- [EABase Jet4](#) (Online Resource)
- [Perform a Project Data Transfer](#) <sup>[504]</sup>
- [List of Available Permissions](#) <sup>[329]</sup>

### 3.8.6.3 Using the Spell Checker

Enterprise Architect has an inbuilt spell checker, which you can configure to detect a range of types of possible spelling error in your project, and to use one of several language dictionaries other than English.

**Access** **Tools | Spelling Tools | Spell Check Project**  
**Tools | Spelling Tools | Spell Check Current Package**

#### Run the spell checker

Step	Action	See also
1	Select the <b>Spell Check Project</b> or <b>Spell Check Current Package</b> menu option, as required.  The Spell Check dialog displays.	
2	( <b>Spell Check Project</b> option.) Select the checkbox against each of the items to spell check within your model.	
3	Click on the <b>Start</b> button to begin the spell check.  As the spell check proceeds, the text being checked displays in the text panel at the bottom of the screen; if an error is detected, the Check Spelling dialog displays, offering several options to correct the error.	<a href="#">Correcting Words</a> <sup>[554]</sup>

#### Notes

- You can use the **Spell Check Project** menu option to check spelling for the entire project
- The **Spell Check Current Package** option only checks the Package currently open, and you cannot

select items to check

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Spell Check** permission to perform any spell check or change any spell check options

#### Learn more

- [List of Permissions](#)

### 3.8.6.3.1 *Correcting Words*

As a spell check progresses, Enterprise Architect highlights any errors or unknown words in the Check Spelling dialog. The inbuilt spell check stores user-defined words in the user dictionary (%APPDATA%\Sparx Systems\EALuserdic.tlx). During the spell check process, if you add a word, it is written into this file for later reference. While the spell check is in progress you can:

- Correct the spelling of a word
- Ignore the error
- Add the word to the user dictionary
- Suggest alternatives or
- Otherwise assist in the spelling correction process

#### Correct spelling

As the Spell Checker identifies possibly mis-spelled words, make the appropriate response:

Action	See also
<ul style="list-style-type: none"> <li>• Modify the spelling by hand and click on the <b>Change</b> or <b>Change All</b> button to change the word to that spelling</li> <li>• Click on a suggested alternative and click on the <b>Change</b> or <b>Change All</b> button to change the word to that spelling</li> <li>• Click on the <b>Ignore</b> or <b>Ignore All</b> button to exclude the word from the spell check</li> <li>• Click on the <b>Add</b> button to add the word to the user dictionary</li> <li>• Click on the <b>Suggest</b> button to list alternative spellings or words</li> <li>• Click on the <b>Cancel</b> button to abort the spell check entirely</li> </ul>	

### 3.8.7 *Personal Tasks*

Using the Personal Tasks view, you can record and manage your personal work within the project. This view displays information based upon your identity as a defined Author on the project.

Access **View | Personal Tasks**

#### Topics

Topic	Detail	See also
<b>Allocated Work</b>	Presents a Gantt chart on which you record the work that you are currently engaged in.	<a href="#">Review Allocated Work</a> <sup>[555]</sup>
<b>Project Tasks</b>	Enables you to monitor and maintain the work tasks that have been assigned to you, or that you have created yourself.	<a href="#">Monitor Your Tasks</a> <sup>[559]</sup>
<b>Workflow</b>	Enables you to list and run model searches defined by Workflow scripts.	<a href="#">Monitor Workflow</a> <sup>[560]</sup>
<b>Working Sets</b>	Enables you to capture your current work area (diagrams and facilities currently open) as a working set for subsequent re-use.	<a href="#">Working Sets</a> <sup>[561]</sup>

#### Learn more

- [The Gantt View](#) <sup>[594]</sup>

### 3.8.7.1 Review Allocated Work

The Allocated Work tab lists the elements to which *your* model Author ID has been allocated as a resource, where your Author ID is the same as:

- Your Enterprise Architect security user ID, if security has been enabled, or
- Your workstation login ID if security has not been enabled

For each element, the tab:

- In the left-hand panel lists the roles or tasks assigned to you as a resource on that element, for which the **Complete %** field value is less than **100**
- In the right hand panel displays a Gantt chart showing your progress in performing each role or task

You can add further work items for an element through the Allocated Work tab; however, you cannot delete any records. A record is no longer listed when the **Complete %** field value is **100**.

You or your supervisors can also add records through the Resource Allocation tab of the Project Management window.

**Access** [View | Personal Tasks > Allocated Work](#)

#### Select Allocated Work options

Option	Action	See also
<b>Review element properties</b>	Double-click on the element name. (Alternatively, right-click on the element name and select the <b>Show Element Properties</b> context menu option.)	<a href="#">Properties Dialog</a> <sup>[956]</sup>

Option	Action	See also
	The Properties dialog for the element displays; review the pages as required.	
<b>Review resource task details</b>	<p>Double-click on the task item.</p> <p>(Alternatively, either:</p> <ul style="list-style-type: none"> <li>• Double-click on the progress bar for the item on the Gantt chart, or</li> <li>• Right-click on the task or role name and select the <b>Show Task Properties</b> context menu option)</li> </ul> <p>The Assigned Resources dialog displays, which has the same content, format and functions as the Resource Allocation tab of the Project Manager window, in Item mode.</p> <p>Should it be necessary to reassign the item to another resource, click on the drop-down arrow on the <b>Resource</b> field and select the appropriate Author ID; when you save the changes, the item no longer appears in the list of tasks assigned to you.</p>	<p><a href="#">Resource Allocation</a> <sup>[512]</sup></p> <p><a href="#">The Project Management Window</a> <sup>[510]</sup></p>
<b>Create new task item</b>	<p>Right-click on the element name and select the <b>Add Resource</b> context menu option.</p> <p>The Assigned Resources dialog displays, with your Author ID in the grayed-out <b>Resource</b> field.</p> <p>Complete the dialog as for the Resource Allocation tab of the Project Manager window.</p>	<p><a href="#">Resource Allocation</a> <sup>[512]</sup></p>
<b>Refresh display to incorporate changes</b>	<p>Your work item can be edited in a number of places in Enterprise Architect, such as the Project Task Allocation window and the Resource Allocation tab; the element to which it is assigned can be edited in these and many other areas.</p> <p>To refresh the display with any changes made elsewhere, right-click on the display and select the <b>Refresh</b> context menu option.</p>	<p><a href="#">Resource Allocation</a> <sup>[512]</sup></p> <p><a href="#">Project Task Allocation</a> <sup>[538]</sup></p> <p><a href="#">Properties Dialog</a> <sup>[956]</sup></p>
<b>Display the Resource Allocation records for the element</b>	<p>Right-click on the entry and select either of the context menu options:</p> <ul style="list-style-type: none"> <li>• <b>Show Element Project Management Window</b> (if the window is closed or hidden)</li> <li>• <b>Find Task in Element Project Management Window</b> (if the window is visible but showing the details of another element)</li> </ul> <p>The Project Management window displays at the Resource Allocation tab, with the details of the selected entry shown in the fields and the other resource allocations for the element listed in the left-hand panel.</p> <p>You can edit the details and, if necessary, change the resource allocated to the element.</p>	<p><a href="#">The Project Management Window</a> <sup>[510]</sup></p> <p><a href="#">Resource Allocation</a> <sup>[512]</sup></p> <p><a href="#">Assign Multiple Resources</a> <sup>[515]</sup></p>

Option	Action	See also
<b>Display tasks for today only, or for another day only</b>	<p>Right-click on the display and select the option:</p> <ul style="list-style-type: none"> <li>• <b>Show only Active tasks for today</b> - to show only tasks that are in progress today</li> <li>• <b>Show only Active tasks for another day</b> - to show only tasks that were in progress on a specific day in the past, or that are scheduled to be in progress on a day in the future; a calendar dialog displays from which you select the day to examine</li> </ul>	
<b>Filter the display by Start or Completion date</b>	<p>The display default is to show current tasks for which the end date has not yet occurred. Right-click on the list and select one of the context menu options:</p> <ul style="list-style-type: none"> <li>• <b>Include Completed Tasks Within the Last...</b> - to display incomplete tasks <b>and</b> tasks completed only within the last period; you can set this period to <b>7</b>, <b>30</b>, or <b>90</b> days, or you can include <b>all</b> completed tasks, or <b>hide</b> all completed tasks</li> <li>• <b>Include Future Tasks Starting in...</b> - to display current incomplete tasks and completed tasks (depending on the setting of the option above) <b>and future</b> tasks that have been recorded and are due to start within the next period; you can set this period to <b>7</b>, <b>30</b> or <b>90</b> days, or you can show <b>all</b> tasks that have been recorded but are not yet due to start</li> </ul>	
<b>Identify overdue tasks amongst the items</b>	<p>Right-click on the display and select the <b>Display Highlight For Overdue items</b> option and one of its sub-options:</p> <ul style="list-style-type: none"> <li>• <b>Show in Red</b> - to display the uncompleted percentage of the task bars for overdue items on the Gantt chart in red</li> <li>• <b>Show in Red to Current Date</b> - to extend the task bars of the overdue items to today's date, and display them in red</li> <li>• <b>None</b> - to cancel any overdue item highlighting that has been set</li> </ul>	
<b>Display ONLY overdue tasks</b>	<p>Right-click on the display and select the <b>Show Overdue Items Only</b> option.</p> <p>The display shows only those items for which the end date has passed but that are not 100% complete. These items do not have red highlighting.</p>	
<b>Locate the element in the Project Browser</b>	<p>Right-click on either the element or the work item and select the <b>Find in Project Browser</b> context menu option.</p> <p>The appropriate package hierarchy expands in the Project Browser, and the selected element is highlighted.</p>	
<b>Reposition the Gantt chart to automatically show the end date of a selected</b>	<p>Right-click on the display and select the <b>Go to   Auto Sync with Task End Date</b> option.</p> <p>Whilst this option is selected (with a tick next to it), whenever you click on an allocation in the Allocated Work tab the display adjusts to show the end date of the task in the center of the chart.</p>	

Option	Action	See also
<b>allocation</b>		
<b>Expose hidden sections of the work item progress</b>	<p>Some items might cover a long period of time, and you might not be able to display the complete progress line for the item in the Gantt chart.</p> <p>To locate the start point, expected end point, or today's date on the progress line for an item, right-click on either the item or the progress line and select the appropriate context menu option:</p> <ul style="list-style-type: none"> <li>• <b>Go To   Task Start Date</b></li> <li>• <b>Go To   Task End Date</b></li> <li>• <b>Go To   Today's Date</b></li> </ul> <p>The Gantt chart shifts left or right to position the required point in the center of the display.</p>	
<b>Execute Scripts</b>	<p>If scripts have been defined for analyzing the task data, click on the <b>Scripts</b> option to display the list of script names. Click on the appropriate name to execute the script on the selected item or items.</p>	
<b>Filter the work items</b>	<p>You can refine the list of work items to show only those containing text that matches the filter item.</p> <p>Right-click on the tab and select the <b>Show Filter Bar</b> context menu option.</p> <p>The filter bar displays at the top of the panel; type in the filter text.</p> <p>As you type, the items listed and the Gantt chart are filtered to show only items where the item names match the text string.</p> <p>The filter does not operate on the element names.</p> <p>If you do not want to use the filter, right-click on the tab and select the <b>Hide Filter Bar</b> context menu option.</p>	
<b>Capture an image of the Allocated Work data as a graphics file</b>	<p>Right-click on the tab and select the <b>Save Image to File</b> context menu option.</p> <p>The Save As Image dialog displays, on which you specify the file name, location and graphics file type to save to.</p>	
<b>Copy an image of the Allocated Work data to the clipboard</b>	<p>Right-click on the tab and select the <b>Copy Image to Clipboard</b> context menu option.</p> <p>You can paste the image from the clipboard into your preferred graphics package.</p>	

[Learn more](#)

- [The Gantt View](#)<sup>[594]</sup>

### 3.8.7.2 Monitor Your Tasks

The Project Tasks tab lists the tasks that you either:

- Own, or
- Are assigned to

These tasks can be created through the:

- Project Tasks tab itself
- Tasks tab of the Project Status window
- Project Tasks view of the Resource Calendar

**Access**   **View | Personal Tasks > Project Tasks**

#### Use to

- Review the status and progress of tasks that you are responsible for
- Create new tasks
- Update existing tasks
- Delete tasks
- Modify and filter the display of task information

#### Make selections from the options

Option	Action	See also
<b>Add or Modify a task</b>	You add or modify a task through the Task Detail dialog, which displays when you double-click on an entry (edit) or blank line (create).	<a href="#">Add, Modify and Delete Tasks</a> <sup>[527]</sup>
<b>Delete a task</b>	Right-click on the message and select the <b>Delete</b> context menu option.  You are prompted to confirm the deletion.	
<b>Select columns</b>	Right-click on the column headings and select the <b>Field Chooser</b> menu option, which enables you to add or remove specific columns from the display.  You can also click on the column headings and drag them across the header bar to reposition the columns in a different sequence.	<a href="#">List Header</a> <sup>[677]</sup>
<b>Reorganize tasks in the list</b>	Either: <ul style="list-style-type: none"> <li>• Click on the column heading and click on the arrow head to list items in order or reverse order, or</li> </ul>	<a href="#">List Header</a> <sup>[677]</sup>

Option	Action	See also
	<ul style="list-style-type: none"> <li>Right-click on the column headings and select the <b>Enable Group Box</b> option to organize the messages into groups</li> </ul> <p>You can also use the filter bar to filter the display on an appropriate column value, such as the value <b>New</b> in the <b>Status</b> column; to display or hide the filter bar, right-click on the column headings and select the <b>Toggle Filter Bar</b> context menu option.</p>	
<b>Set persistent Status filter</b>	<p>Right click on the tab and select the <b>Set term filter</b> context menu option.</p> <p>The Filter by Status dialog displays, which enables you to select to list tasks of any status or only of one specific status.</p> <p>The filter you set persists when you close the Personal Tasks window or exit from Enterprise Architect.</p>	
<b>Print the task list</b>	<p>Right click on the tab and select the <b>Print List</b> context menu option.</p> <p>The Print dialog displays, on which you specify the local printer and the print characteristics.</p>	

#### Learn more

- [Project Task Display](#)<sup>[58]</sup>
- [Project Tasks](#)<sup>[52]</sup>

### 3.8.7.3 Monitor Workflow

The Workflow tab lists all of the workflow-related searches that have been defined for you. You can select and execute any of the searches listed within this tab.

**Access** **View | Personal Tasks > Workflow**

On the Workflow tab, double-click on the required script to execute it.

#### Notes

- This facility is available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect

#### Learn more

- [Workflow Scripts](#)<sup>[36]</sup>



### 3.8.7.4 Working Sets

As you perform work on your model, you open various windows, diagrams and views, and you will probably need to return to these same views in a later work session. At the end of your work session, you can capture your current work environment as a **working set**, which you use to return to that exact environment later, either when you choose to or by having the system automatically recreate it when you next log on to Enterprise Architect. You can also create a working set as an empty definition to which you add specific windows, diagrams and views, from either the **Working Sets tab** or from a **Working Set Artifact element**.



You can use the working set to pick up your work from where you left off, or as a set of 'home pages' to start from each day. You can also open multiple working sets, which combines the views from each set. This can be useful to check the progression between different stages in a piece of work.

A working set is initially a personal list, not visible to or active for any other user. You can convert it into a global set to share with other users of your model; for example, if a working set shows a cross-section of an aspect of the model you want to demonstrate or have reviewed.

**Access** [View | Personal Tasks > Working Sets](#)



#### Manage working sets

Option	Action	See also
<b>Create a working set from open diagrams or files</b>	<p>Either:</p> <ul style="list-style-type: none"> <li>Click on the first icon in the toolbar, or</li> <li>Right-click on the body of the tab and select the <b>Create Working Set</b> context menu option, or</li> <li>Right-click on the body of the tab and select the <b>Create Working Set From</b> context menu option, and either the <b>Open Diagrams</b> or <b>Open Files</b> sub-option as appropriate</li> </ul> <p>The Create Working Set dialog displays, listing the diagrams and views that are currently open, in the order in which they were opened. If you wish, you can change the order using the 'hand' buttons at the bottom of the dialog.</p> <p>Select the checkbox against each item that you want to retain in the working set, or click on the <b>Include All</b> button. If you selected the <b>Create Working Set From</b> option, the checkboxes against the open diagrams or the open files are already selected.</p> <p>In the <b>Name</b> field, type a name for the working set.</p> <p>You can either:</p> <ul style="list-style-type: none"> <li>Immediately click on the <b>OK</b> button to create the working set from the selected items, or</li> <li>Add other selections as described for <i>Create a new working set</i> (below)</li> </ul>	
<b>Create a new working set</b>	<p>Either:</p> <ul style="list-style-type: none"> <li>Click on the first icon in the toolbar, or</li> <li>Right-click on the body of the tab and select the <b>Create Working Set</b> context menu option, or</li> </ul>	

Option	Action	See also
	<ul style="list-style-type: none"> <li>Switch to an open diagram, drag a <b>Working Set</b> icon from the Artifacts page of the Diagram Toolbox onto the diagram, right-click on the new Working Set Artifact element and select the <b>Edit Working Set</b> option</li> </ul> <p>The Create Working Set dialog displays.</p> <p>In the <b>Name</b> field, type a name for the working set.</p> <p>To add a diagram or view, click on the <b>Add Other</b> button and select an option from the list:</p> <ul style="list-style-type: none"> <li><b>Add Diagram</b></li> <li><b>Add Matrix Profile</b></li> <li><b>Add Search</b></li> <li><b>Add Team Review</b></li> <li><b>Add Document Template</b></li> <li><b>Add Resource Document</b></li> </ul> <p>You are prompted to select the required diagram, profile, search, Team Review topic, document report template or Resource document name(s); each view is then added to the list with its checkbox selected.</p> <p>Click on the <b>Add Other</b> button again, to select the next item.</p> <p>When you have all the diagrams and views you need, click on the <b>OK</b> button to create the working set.</p>	<a href="#">Artifact</a> <sup>[1358]</sup>
<b>Open an existing working set</b>	<p>A Working Set created on the Working Sets tab is indicated by this icon .</p> <p>A Working Set created as an Artifact element is indicated in the list by this icon .</p> <p>Click on a working set name and either:</p> <ul style="list-style-type: none"> <li>Select the third icon in the toolbar or</li> <li>Right-click and select the <b>Open Working Set</b> context menu option</li> </ul> <p>Enterprise Architect opens all the diagrams and views listed in that set.</p> <p>If you already have a working set open, the 'new' set of views is <b>added</b> to the original set, with the last-listed diagram or view on top and in focus.</p> <p>You open one Model Search, Matrix profile or Team Review at a time; therefore, the last-opened Search, profile or Team Review replaces the previous instances.</p>	
<b>Automatically apply a working set when the</b>	Right-click on the working set name and select the <b>Apply when model opens</b> context menu option.	

Option	Action	See also
<b>model is opened</b>	<p>This option is a toggle that turns the same option off for the previous working set on which it was selected.</p> <p>The folder turns green, indicating that the option is 'on' for this working set.</p> <p>When you close the model and subsequently open it, the diagrams and views in the working set are automatically opened.</p>	
<b>Share a working set</b>	<p>Right-click on the working set name and select the <b>Share Working Set</b> context menu option.</p> <p>The folder icon on the left of the working set name in the list now contains an icon depicting the globe, and the working set is available in the Working Sets tab of every other user of the model.</p> <p>This setting is not reversible; to make the global list into a personal list, make a copy of it (see below).</p>	
<b>Add an open view or diagram to a working set</b>	<p>If, in your work, you open a diagram or view that you decide to add to a working set:</p> <p>Click on the Working Sets tab, right-click on the required set name and select the <b>Add Active View</b> context menu option.</p> <p>The open view is added to the working set.</p>	
<b>Duplicate a working set</b>	<p>If you want to use a variation of a working set without losing the original, you can create a copy and amend that copy as required.</p> <p>Right-click on the source working set and select the <b>Copy as New</b> context menu option.</p> <p>The Create Working Set dialog displays.</p> <p>In the <b>Name</b> field, type a name for the working set.</p> <p>Each of the listed diagrams and views is already selected; make any changes you require, and click on the <b>OK</b> button to save the list as a new working set.</p>	
<b>Edit a working set</b>	<p>To change the diagrams and views in a working set, or change the set name, click on the set name and either:</p> <ul style="list-style-type: none"> <li>Click on the second icon in the toolbar or</li> <li>Right-click and select the <b>Edit</b> context menu option</li> </ul> <p>The Create Working Set dialog displays.</p> <p>Make any changes you require, and click on the <b>OK</b> button to save the changes.</p>	
<b>Delete a working set, or a view in working set</b>	<p>Expand the working set and select a specific item (to remove that item) or the set name (to move the whole set)</p> <p>Either:</p>	

Option	Action	See also
	<ul style="list-style-type: none"> <li>Click on the fifth icon in the toolbar or</li> <li>Right-click on the set and select the <b>Delete Working Set Item</b> context menu option</li> </ul> <p>A prompt displays to confirm the deletion; click on the <b>Yes</b> button.</p> <p>The item or set is removed.</p> <p>If you delete a working set that you have opened, the diagrams and views from that set remain open until you specifically close each one or close Enterprise Architect.</p> <p><b>Note:</b></p> <ul style="list-style-type: none"> <li>When you create a Working Set on a diagram, using the icon from the Common page of the Diagram Toolbar, the Working Set persists in the Personal Tasks view if you delete the element from the <b>diagram</b>, but <b>not</b> if you delete the element from the <b>Project Browser/model</b></li> </ul>	
<b>Save floating window layout in working set</b>	<p>If any of the selected diagrams or views in your working set are floating windows, the Create Working Set or Edit Working Set dialog includes the item <b>Save Workspace Layout</b>.</p> <p>Select this checkbox; the working set captures the complete current workspace layout, including the locations of the floating or docked views.</p> <p>Next time you load the working set, the captured workspace layout is reapplied.</p>	<a href="#">Window Menu</a> [121]  <a href="#">Manage Workspace Layout</a> [163]
<b>Track working set/views last opened</b>	<p>You might use several working sets in a work session, and close some views and open others.</p> <p>To capture the diagrams and views you actually have open when you close down Enterprise Architect, right click on the body of the tab and select the <b>Store Main Tab View History</b> context menu option. A new, empty working set folder displays at the top of the list - <i>Windows open when Model was last closed</i>.</p> <p>When you reopen the model in a new work session, this new folder is automatically populated with a list of the diagrams and views that were open when you last closed the model; this is a read-only set that cannot be edited.</p> <p>You can expand the set and double-click on specific windows to re-open them, or right-click on the set and use the <b>Open Working Set</b> option to reopen all the diagrams and views you had open before.</p> <p>However, it is recommended that you use the <b>Apply when model opens</b> context menu option on the set; thereafter, whenever you open the model, all the diagrams and views you had open when you closed the model are automatically re-opened.</p>	
<b>Locate a diagram from the working set</b>	<p>Either:</p> <ul style="list-style-type: none"> <li>Right-click on the diagram name and select the <b>Find in Project Browser</b> context menu option, or</li> </ul>	

Option	Action	See also
<b>in the Project Browser</b>	<ul style="list-style-type: none"> <li>Click on the diagram name and click on the <b>Locate in Project Browser</b> toolbar icon ()</li> </ul> <p>The section of the Project Browser containing the diagram is expanded, and the diagram name is highlighted.</p>	
<b>Locate a Working Set Artifact element in the model</b>	<p>Either:</p> <ul style="list-style-type: none"> <li>Right-click on the element name and select the <b>Find in Project Browser</b> context menu option, or</li> <li>Click on the element name and click on the <b>Locate in Project Browser</b> toolbar icon ()</li> </ul> <p>The section of the Project Browser containing the element is expanded, and the element name is highlighted.</p>	

#### Learn more

- [Create Project Shortcut](#)<sup>[206]</sup>
- [Capture Current Work Environment](#)<sup>[207]</sup>

### 3.8.8 Model Mail

The **Model Mail** tab shows all emails that you have received from project team members within the project, under your User Security ID. You can open and respond to these emails and create new ones.

The **Sent Items** tab shows all the emails you have sent to project team members under your User Security ID. You can forward these emails and make further responses (see the final table in this topic).

This facility is available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, with User Security enabled; the mail system uses the individual and group users defined in User Security.

**Access** [View | Model Mail](#)

#### Use to



- Read messages from other project team members
- Read and respond to messages sent to you as a member of a group that has a shared mail inbox
- Organize and sort both sent and received messages
- Create new messages, and reply to and forward received messages
- Delete messages from your Inbox and Sent Items mailboxes

#### Review your Model Mail messages

Field/Option/ Button	Action	See also
<b>Open a message</b>	Double-click on the message.  If the message contains a hyperlink, click on the link to open or display the target file or object.	
<b>Flag</b>	Check the flag color to establish the significance of the message (the meanings of the flag colors should be defined within the project team).	
<b>Status</b>	Check for the read or unread icon; unread message items are anyway displayed in bold.	
<b>Sender</b>	The name of the project team member who sent the message.	
<b>Subject</b>	The topic of the message.	
<b>Date</b>	Indicates the age of the message, whether it was sent: <ul style="list-style-type: none"> <li>• Today</li> <li>• Yesterday</li> <li>• This week</li> <li>• This Month</li> <li>• Last Month</li> <li>• Older</li> </ul>	
<b>Sent</b>	The exact date and time the message was sent.	
<b>Select Columns</b>	<p>Right-click on the column headings and select the <b>Field Chooser</b> menu option, which enables you to add or remove specific columns from the display.</p> <p>You can also click on the column headings and drag them across the header bar to reposition the columns in a different sequence.</p> <p>Adding the <b>To</b> column will display the user account or group name that this mail item was sent to; this column is useful to identify whether the message was sent to you as an individual or as a member of a <b>Group</b> with a <b>Shared Mail</b> inbox - if you received the message as a member of a shared mail group, remember that your actions on this message will also affect all other members of this group.</p>	<a href="#">List Header</a> [677]
<b>Filter Columns</b>	<p>You can filter the mail items listed by displaying the Filter Bar (right-click on the column headings and select the <b>Toggle Filter Bar</b> option) and typing in the value (or partial value) to filter on.</p> <p>For example, if you type your group mail account name in the filter field for the <b>To</b> column, you will list only those mail items sent to the group</p>	<a href="#">Maintain Groups</a> [320]

Field/Option/ Button	Action	See also
	<p>mail address, not those sent to you as an individual.</p> <p>To clear the value of a filter field, click on the blue cross against that field.</p>	
<b>Reorganize messages in the list</b>	<p>Either:</p> <ul style="list-style-type: none"> <li>Click on a column heading to toggle sorting of list items by this column in ascending or descending order, or</li> <li>Right-click on the column heading and select the <b>Enable Group Box</b> option to organize the messages into groups</li> </ul>	

#### Model Mail Toolbar Options

Option	Action	See also
<b>Compose Message</b>	Open the Model Message dialog, through which you create and send a mail message.	<a href="#">Create a Message</a> 
<b>Delete Messages</b>	Delete a selected message or messages; you are prompted to confirm the deletion.	
<b>Reply</b>	Open the Model Message dialog, through which you create a response to the sender of the message, which includes a copy of the message and any earlier messages in the thread.	<a href="#">Create a Message</a> 
<b>Reply to All</b>	Open the Model Message dialog, through which you create a response to the sender of the message and the other recipients, which includes a copy of the message and any earlier messages in the thread.	
<b>Forward</b>	Open the Model Message dialog, through which you forward the message to other project team members with, if necessary, your own comments.	
<b>Mark as Unread</b>	Mark the selected messages as unread.	
<b>Mark as Read</b>	Mark the selected messages as read.	
<b>Viewing Panel</b>	Adds a panel to the Model Mail window showing the contents of the currently selected message. The Viewing Panel can be displayed to the <b>Right</b> or <b>Bottom</b> of the message list. Select <b>Hide</b> to show only the	

Option	Action	See also
	message list.	
<b>Help Contents</b>	Display this Help topic.	

#### Model Mail Context menu options

Option	Action	Shortcut	See also
<b>Compose New Message</b>	Click to open the Model Message dialog, through which you create and send a mail message.		<a href="#">Create a Message</a> <sup>[570]</sup>
<b>Reply</b>	Create a response to the sender of the message, which includes a copy of the message and any earlier messages in the thread.	<b>Ctrl+R</b>	
<b>Reply to All</b>	Create a response to the sender of the message and the other recipients, which includes a copy of the message and any earlier messages in the thread.	<b>Ctrl+Shift+R</b>	
<b>Forward</b>	Forward the message to other project team members with, if necessary, your own comments.	<b>Ctrl+F</b>	
<b>Set Message Flag</b>	Select the appropriate flag color to establish the significance of the message (the meanings of the flag colors should be defined within the project team).		
<b>Mark as Unread</b>	Mark the selected messages as unread.	<b>Ctrl+U</b>	
<b>Mark as Read</b>	Mark the selected messages as read.	<b>Ctrl+Q</b>	
<b>Delete</b>	Delete a selected message or messages; you are prompted to confirm the deletion.		

#### Manage your sent messages on the Sent Items tab



Field/Option/ Button	Action	See also
<b>Open a message</b>	<p>Double-click on the message.</p> <p>From the open message you can reply to the sender, reply to the sender and all other recipients, or forward the message to other team members.</p>	<a href="#">Create a Message</a> <sup>670</sup>
<b>Flag</b>	Check the flag color to establish the significance of the message (the meanings of the flag colors should be defined within the project team).	
<b>To</b>	Check the name(s) of the project team member(s) to whom the message was sent.	
<b>Subject</b>	The topic of the message.	
<b>Date</b>	<p>Indicates the age of the message, whether it was sent:</p> <ul style="list-style-type: none"> <li>• Today</li> <li>• Yesterday</li> <li>• This week</li> <li>• Last Week</li> <li>• This Month</li> <li>• Last Month</li> <li>• Older</li> </ul>	
<b>Sent</b>	The exact date and time the message was sent.	
<b>Select Columns</b>	<p>Right-click on the column headings and select the <b>Field Chooser</b> menu option, which enables you to add or remove specific columns from the display.</p> <p>You can also click on the column headings and drag them across the header bar to reposition the columns in a different sequence.</p> <p>Adding the <b>Sender</b> column is useful to identify whether you have sent the message as an individual, or as a member of a shared mail group.</p>	<a href="#">List Header</a> <sup>671</sup>
<b>Reorganize messages in the list</b>	<p>Either:</p> <ul style="list-style-type: none"> <li>• Click on a column heading to toggle sorting of list items by this column in ascending or descending order, or</li> <li>• Right-click on the column heading and select the <b>Enable Group Box</b> option to organize the messages into groups</li> </ul>	
<b>Delete messages</b>	Right-click on the message and select the <b>Delete</b> context menu option.	

Field/Option/ Button	Action	See also
	You are prompted to confirm the deletion.	
<b>Reply to All</b>	Right-click on the message and select the <b>Reply to All</b> context menu option. The Model Message dialog displays.	<a href="#">Create a Message</a> <sup>570</sup>
<b>Forward</b>	Right-click on the message and select the <b>Forward</b> context menu option. The Model Message dialog displays.	

### 3.8.8.1 Create a Message

The **Model Message** dialog enables you to compose messages to project team members within the project, under your Author ID.

Access **View | Model Mail**



#### Use to

- Create messages to other project team members
- Reply to messages from other team members
- Forward messages to other team members
- Link model components to the message, or add hyperlinks to files or other objects of relevance

#### How to

To create a message to another project team member as a new message, a reply to a message, or a forwarded message

Step	Action	See also
<b>1</b>	If your message or response concerns a diagram or model object, you could click on that object in the Project Browser now so that it is immediately available to be linked to your message.	
<b>2</b>	Select the appropriate toolbar icon, context menu option or message button to create, reply to or forward a message. The <b>Model Message</b> dialog displays.	<a href="#">Model Mail</a> <sup>565</sup>
<b>3</b>	If you are creating or forwarding a message, or you want to send a reply to a wider audience, for <i>each</i> person you intend to send the message to: <ul style="list-style-type: none"> <li>• Click on the <b>To:</b> button</li> </ul>	

Step	Action	See also
	<p>A list of project team member and group IDs displays</p> <ul style="list-style-type: none"> <li>• Scroll to the required name and double-click on it The name is added to the <b>To:</b> field</li> </ul>	
4	In the <b>Subject</b> field, type or edit the subject of the message as necessary.	
5	In the <b>Flag</b> field click on the drop-down arrow and on the appropriate flag color or option for your message.	
6	<p>In the text panel, write the text of your message.</p> <p>You can format the text using the facilities of the Notes toolbar at the top of the field.</p>	<a href="#">Notes Toolbar</a> 
7	<p>If you intend to link a diagram or model object to this message, place the cursor at the appropriate position in your message text and click on the <b>Insert Quick Link</b> button.</p> <p>Select the object you highlighted in Step 1; a hyperlink to that object is inserted in the message text.</p>	
8	<p>If you did not select a model object and now intend to link to one, or you want to link to another target such as a file, Model Search or Team Review message, click on either:</p> <ul style="list-style-type: none"> <li>• The <b>Hyperlink</b> icon in the toolbar or</li> <li>• The <b>Insert Quick Link</b> button and select the <b>Other</b> option</li> </ul> <p>The Hyperlink Details dialog displays; create the required link.</p>	<a href="#">Hyperlinks</a> 
9	Click on the <b>Send</b> button to send the message to the recipients.	

### Notes


- This facility is available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, with User Security enabled

## 3.8.9 Project Calendar

The Project Calendar provides a project management overview in calendar format of the deployment of resources, timeframes for tasks, and upcoming project events such as meetings and milestones. The calendar displays in the Diagram View workspace, consisting of a:

- Calendar panel, displaying months of the year
- Diary panel sectioned into days or weeks
- Toolbar at the top of the window, which enables you to define what information is displayed and how it is formatted

**Access** **Project | Calendar**

Screen area	Description	See also
<b>Calendar Panel</b>	<p>The calendar panel, when viewed normally on the screen, displays a calendar of six months that by default includes the current month.</p> <p>By closing or reducing other panels on the screen, and depending on the size of your screen, you can show the months for a period of up to 3 years and six months.</p> <p>You can review data for up to several decades into the past, if such data has been recorded, and plan ahead for up to several decades into the future; to display future or past months, either:</p> <ul style="list-style-type: none"> <li>Click on the arrows in the names of the months on the left and right of the top row of months, or</li> <li>Click and hold the mouse on the name of a month so that a list of months and years displays, then move the mouse forwards to scroll up the list or backwards to scroll down the list; when you reach the required month, ensure that it is highlighted and release the mouse button</li> </ul> <p>On the calendar, today's date has a red border; if you have scrolled the calendar so that today's date is not shown, click on the second icon in the toolbar () to position the current month in the top left of the calendar with today's date highlighted.</p>	
<b>Diary Panel</b>	<p>The diary panel displays a set of day fields, each of which can contain one or more markers for an event or task.</p> <p>The content of the display is determined by the selections you make in the toolbar, and principally by the object type you select from the first drop-down field:</p> <ul style="list-style-type: none"> <li>Project Calendar</li> <li>Allocated Resources</li> <li>Project Tasks</li> </ul> <p>The structure of the display depends on the period you want to review, specified using the icons on the toolbar:</p> <ul style="list-style-type: none"> <li>1 day (Project Calendar only)</li> <li>5 days (Mon - Fri) (Project Calendar only)</li> <li>7 days (Mon - Sun)</li> <li>31 days</li> </ul> <p>However, in the Calendar panel you can highlight specific periods of between 1 and 7 days (Project Calendar only) and 14, 21, 28, 35 and 42 days, to display just those periods in the Diary panel.</p> <p>If you select a period of between 1 and 7 days, each day is divided into one hour time intervals; you can adjust the time intervals for these periods, using the toolbar (see below).</p>	<a href="#">Calendar</a> <sup>[574]</sup> <a href="#">Allocated Resources</a> <sup>[578]</sup> <a href="#">Project Tasks</a> <sup>[526]</sup>

Screen area	Description	See also
	<p>For the current day, the current time is also highlighted.</p> <p>Also, if you click on today's date in the Calendar panel the Diary panel changes to show the shortest period (7 days or 1 day) for the object type, again with adjustable time intervals for the times of day.</p> <p>For displays of 7 or more days, you can scroll up or down to display the information for earlier or later dates; if you scroll away from today's date, you can return to it by right-clicking anywhere on the display and selecting the <b>Show Today</b> context menu option (not Project Calendar).</p>	
<b>Toolbar</b>	<p>The toolbar options modify what is displayed on the Diary panel, and enable you to add certain kinds of information.</p> <p>The icons and fields in the toolbar, from left to right, have the following functions:</p> <ul style="list-style-type: none"> <li>• <b>Add New &lt;object&gt;</b> - enables you to create a new record for an event or task</li> <li>• <b>Assign Resource</b> - for Allocated Resources, enables you to assign a new resource to the task</li> <li>• <b>Show Today</b> - re-focuses the Diary panel and Calendar panel on today's date, for any object type</li> <li>• <b>1, 5, 7, 31</b> - sets the number of days displayed in the Diary panel</li> <li>• First drop-down - identifies the type of information to display: Calendar events, allocated Resources, or Tasks</li> <li>• Second drop-down - identifies the type or characteristic of object to filter for (and depends on the value selected in the first field)</li> <li>• Third drop-down - for Allocated Resources, identifies the element type to filter for</li> <li>• Options: <ul style="list-style-type: none"> <li>• <b>Configure Event Subtypes</b> - Displays the Configure Subtype dialog, which you use to define categories of event to record on the Calendar</li> <li>• <b>Time Scale</b> - where the display includes times of day, enables you to reset the time interval</li> <li>• <b>Show End Time</b> - for the Project Calendar, where the times of day are NOT shown and an event occurs within one day, this shows or hides the time at which the event finishes (the start time displays automatically)</li> <li>• <b>Show Time As Clocks</b> - toggle between showing start and end times in digital format and as a clock face</li> <li>• <b>Compress Weekend Days</b> - in 31-day format, toggle between showing Saturday and Sunday as separate fields and a pair of half-fields</li> <li>• <b>Show ToolTips</b> - toggle between showing and hiding mouse roll-over tooltips</li> </ul> </li> </ul>	<a href="#">Configure Event Subtypes</a> <sup>577</sup>

**Notes**

- The Project Calendar is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect
- In the Lite edition of Enterprise Architect, you can view Calendar entries but not change them

### 3.8.9.1 Calendar

**Project Calendar** is a display mode of the project Calendar facility. In Project Calendar mode, the Diary panel shows the selected period of time (see the *Project Calendar* topic) containing flags for each *event* that takes place in or through that period. These events can be, for example, meetings, staff absences or commitments, product releases or public holidays.

You can customize the range of events that you record, using the Configure Subtype dialog.

Access **Project | Calendar > Project Calendar**

#### Record an event in the Project Calendar

Step	Action	See also
1	Double click on the required day, or the first day in a longer period. The Project Event dialog displays.	
2	In the <b>Subject</b> field, type the name of the event.	
3	If the event is to take place at a specific place, in the <b>Location</b> field type the name of the place or room.	
4	In the <b>Category</b> field, click on the drop-down arrow and select the appropriate categorization: <ul style="list-style-type: none"> <li>• <b>Event</b> (such as an external trade show, or internal presentation)</li> <li>• <b>Meeting</b> (whether internal or external)</li> <li>• <b>Milestone/Objective</b> (such as a product test or release)</li> <li>• <b>People</b> (an absence or commitment of a specific staff member)</li> </ul>	
5	If the event is likely to occupy one complete day, select the <b>All day event</b> checkbox; this: <ul style="list-style-type: none"> <li>• Places the event icon at the top of the day field in the Diary, above any time sections</li> <li>• Hides the time sections of the <b>Start time</b> and <b>End time</b> fields</li> </ul>	
6	In the <b>Start time</b> and <b>End time</b> fields, specify the start and end dates and/or times of the event.  In the date section of each field, either: <ul style="list-style-type: none"> <li>• Click on the drop-down arrow to display the current month and select the month and date, or</li> </ul>	

Step	Action	See also
	<ul style="list-style-type: none"> <li>Click on the 'spin' arrows to roll the date forwards or backwards by one day at a time</li> </ul> <p>In the time section of each field, click on the hour or minute components (which are separate) and either:</p> <ul style="list-style-type: none"> <li>Type in the required time or</li> <li>Click on the 'spin' arrows to roll the time backwards or forwards</li> </ul>	
7	<p>In the <b>Event Type</b> field, click on the drop-down arrow and select an appropriate type label for the event; the options change depending on the value you selected for the <b>Category</b> field.</p> <p>This provides the fill color for the event icon, as indicated by the fill box next to each option.</p>	
8	In the <b>Defined as</b> field, click on the drop-down arrow and select the indicator for whether the event is internal or external to your organization.	
9	If the event consists of a telephone call, select the <b>Phone Call</b> checkbox.	
10	In the description field, type any notes required on the event; these display in the mouse roll-over tool tip for the event on the Diary panel.	
11	If this event repeats at regular intervals, click on the <b>Recurrence</b> button and complete the Event Recurrence dialog (see below).	
12	Click on the <b>OK</b> button to save the event and display it on the Calendar.	

#### Define a recurring event

Step	Action	See also
1	<p>In the <b>Start</b> and <b>End</b> fields, specify the start and end times of the event; click on the hour or minute components (which are separate) and either:</p> <ul style="list-style-type: none"> <li>Type in the required time or</li> <li>Click on the 'spin' arrows to roll the time backwards or forwards</li> </ul>	
2	In the <b>Duration</b> field, click on each of the day(s), hours and minutes portions and either type or 'spin' to the required value.	

Step	Action	See also
	<p>The hours and minutes portions are linked to the <b>Start</b> and <b>End</b> fields, so that:</p> <ul style="list-style-type: none"> <li>As you edit the <b>Start</b> field or the <b>Duration</b> field, the <b>End</b> field adjusts to maintain the duration relative to the start time</li> <li>As you edit the <b>End</b> field, the <b>Duration</b> field changes to match the difference between the start and end times</li> </ul>	
3	<p>In the Recurrence pattern panel, select the appropriate radio button for the frequency of the event.</p> <p>As you select the radio button, the fields on the right of the panel change to further define when the event recurs.</p>	
4	<p>Select the appropriate radio buttons and field values, with the following guidance:</p> <ul style="list-style-type: none"> <li><i>Daily</i> can be every 1, 2, 3, 4, 5 or 6 days, or every day of the working week</li> <li><i>Weekly</i> is on one or more specific days of the week, the events being separated by a period of between 1 and 51 complete weeks</li> <li><i>Monthly</i> can be on a specific date or a specific day of the month, the events being separated by a period of between 1 and 11 complete months</li> <li><i>Yearly</i> can be on a specific date or a specific day of a specific month</li> </ul>	
5	<p>In the Range of recurrence panel, in the <b>Start</b> field, specify the date on which the event cycle begins.</p> <p>Click on the day, month and year components (which are separate) and either:</p> <ul style="list-style-type: none"> <li>Type in the required value or</li> <li>Click on the 'spin' arrows to roll the date backwards or forwards</li> </ul>	
6	<p>Select the appropriate radio buttons to indicate that the recurrence cycle:</p> <ul style="list-style-type: none"> <li>Has no defined end point</li> <li>Ends after a specific number of occurrences of the event (type in the number)</li> <li>Ends by a specific date (enter the date)</li> </ul>	
7	<p>Click on the <b>OK</b> button, and again on the Project Event dialog.</p> <p>The event icon displays at all points in the Calendar where it is scheduled to occur, as defined in the Event Recurrence dialog; the recurrence is indicated by a graphic depicting two recirculating arrows.</p>	
8	<p>If the recurrence is no longer required:</p> <ul style="list-style-type: none"> <li>Double-click on the icon for any occurrence of the event, to display the Project Event dialog</li> <li>Click on the <b>Recurrence</b> button to display the Event Recurrence dialog</li> <li>Click on the <b>Remove Recurrence</b> button; the Event Recurrence dialog closes</li> </ul>	



Step	Action	See also
	<ul style="list-style-type: none"> <li>Click on the <b>OK</b> button; the Project Event dialog closes</li> </ul> <p>Only the first occurrence of the event remains in the Calendar.</p>	

### Notes

- The Project Calendar is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect
- In the above editions, if security is enabled you must have **Manage Project Calendar** permission in order to create, update or delete Project Calendar events; if security is not enabled, you can change data without this permission
- In the Lite edition of Enterprise Architect, you can view Calendar entries but not change them
- You can transport a calendar of defined events between projects, using the **Export Reference Data** and **Import Reference Data** options

### Learn more

- [Project Calendar](#)<sup>[57]</sup>
- [Configure Event Subtypes](#)<sup>[57]</sup>
- [List of Available Permissions](#)<sup>[329]</sup>
- [Export Reference Data](#)<sup>[376]</sup>
- [Import Reference Data](#)<sup>[380]</sup>

#### 3.8.9.1.1 Configure Event Subtypes

The Configure Subtype dialog enables you to define additional types of event to record on the Project Calendar, such as:

- Define a new event type within a select event category
- Change an existing event type
- Delete an existing event type

**Access** **Project | Calendar:**  **| Configure Event Subtypes**

### Manage the event types for your project

Step	Action	See also
1	On the Configure Subtype dialog, in the <b>Category</b> field, click on the drop-down arrow and select the appropriate category for the event.	
2	Click on the <b>Type</b> field and type a suitable name for the event type (or select an existing	

Step	Action	See also
	event from the list below the <b>Category</b> field).	
3	In the <b>Color</b> field, click on the drop-down arrow and select or define a color for the event icon.	
4	Click on the <b>Save</b> button to save the new or edited event type to the list for the specified category.  Click on the <b>New</b> button if you are going to create another event type.	
5	If the event type is no longer required, click on the <b>Delete</b> button.	
6	If required, you can change the sequence of the event types in the displayed list, using the ' <b>Up Hand</b> ' and ' <b>Down Hand</b> ' buttons at the bottom of the dialog.	
7	Click on the <b>OK</b> button to close the dialog; any new event types are available for use in the Project Calendar.	

### Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Project Calendar** permission in order to create or delete Project Calendar event subtypes; if security is not enabled, you can change data without this permission
- You can transport your defined event types between projects, using the **Export Reference Data** and **Import Reference Data** options

### Learn more

- [Calendar](#)<sup>[574]</sup>
- [Export Reference Data](#)<sup>[376]</sup>
- [Import Reference Data](#)<sup>[380]</sup>

### 3.8.9.2 Allocated Resources

**Allocated Resources** is a display mode of the project Calendar facility, that you can use to:

- Review the allocation of resources across the project over a period of time
- Review the status of the work assigned to those resources
- Display and edit the details of the task assigned to a resource
- Allocate further resources to an element
- Delete a resource allocation from an element
- Create and send an email to the resource allocated to an element

In Allocated Resources mode, the Diary panel shows the selected period of time (see the *Project Calendar* topic), with at least one icon representing each element in the project to which a resource has been allocated during that period. If an element has more than one allocated resource, each element:resource combination is represented separately.

Each icon shows the element name and resource name, and indicates the status of the assigned work with one of the following symbols:

- a green square, indicating that the resource has been assigned the work
- a green tick, indicating that the resource has completed the assigned work
- a red square, indicating that the resource has not completed the assigned work
- a square divided diagonally with green and red halves, indicating that the assigned work is to be completed within one working day

Typically, while the work is in progress a resource is represented by:

- An icon with a green square on the day the work is planned to start, and
- An icon with a red square on the day the work is planned to finish

When the work is complete, the icons on both days display the green tick.

Resources are initially allocated to elements through the:

- Resource Allocation tab of the Project Management window, or
- Allocated Work tab of the Personal Tasks window

You can then *edit* these allocations through either of those windows or through the Allocated Resources Calendar itself.

**Access**   **Project | Calendar > Allocated Resources**

### **Allocate Resources**

Option	Usage	See also
<b>Display/edit details of an allocated resource</b>	<p>Either:</p> <ul style="list-style-type: none"> <li>• Double-click on any icon for that element:resource combination, or</li> <li>• Right-click on the icon and select the <b>Properties</b> context menu option</li> </ul> <p>The Assigned Resources dialog displays, showing the details of the assignment of the resource to a task on the element.</p> <p>This dialog has the same content, format and functions as the Resource Allocation tab of the Project Manager window, in Item mode.</p>	<p><a href="#">Resource Allocation</a> <sup>[512]</sup></p> <p><a href="#">The Project Management Window</a> <sup>[510]</sup></p>

Option	Usage	See also
<b>Add resource to element</b>	<p>Right-click on any icon for the element and select the <b>Assign Resource to &lt;element name&gt;</b> context menu option.</p> <p>The Assigned Resources dialog displays, showing the start and end dates both set to the date from which you selected the icon, and the <b>Allocated Time</b> field set to <b>1</b>; all other fields are blank.</p> <p>This dialog has the same content, format and functions as the Resource Allocation tab of the Project Manager window, in Item mode.</p> <p>Specify the resource and the role or task that resource is performing, and define the period for which the resource is allocated to the element.</p>	<p><a href="#">Resource Allocation</a> [512]</p> <p><a href="#">The Project Management Window</a> [510]</p>
<b>Message Resource</b>	<p>Right-click on any icon for the element:resource combination and select the <b>Message Resource &lt;resource name&gt;</b> context menu option.</p> <p>The Model Message dialog displays, on which you create and send your message to the resource allocated to the element.</p>	<a href="#">Create a Message</a> [570]
<b>Delete resource from element</b>	<p>Right-click on any icon for the element:resource combination, and select the <b>Delete Resource from &lt;element name&gt;</b> context menu option.</p> <p>A prompt displays to confirm the deletion; click on the <b>Yes</b> button.</p> <p>The icon and any corresponding icons for that element:resource combination are deleted from the calendar, and the resource is no longer allocated to that element.</p>	
<b>Locate element in diagrams in which it is used</b>	<p>Right-click on any icon for the element, and select the <b>Find in all Diagrams</b> context menu option.</p> <p>If the element is used in only one diagram, that diagram displays.</p> <p>If the element is used in more than one diagram the Element Usage dialog displays, listing the diagrams in which the element occurs.</p> <p>Select the required diagram and click on the <b>Open</b> button to display that diagram.</p> <p>This option also operates on Port and Part Property Type Classifiers.</p>	<a href="#">Show Element Use</a> [910]
<b>Locate element in Project Browser</b>	<p>Right-click on any icon for the element, and select the <b>Find in Project Browser</b> context menu option.</p> <p>The area of the Project Browser containing the element is brought into focus and expanded, and the element is highlighted.</p>	

Option	Usage	See also
<b>Locate resource allocation start date corresponding to allocation end date</b>	<p>Firstly, click on the end date icon for the element:resource combination to highlight it; the corresponding start date icon is also highlighted.</p> <p>If the start date icon is not in view, right-click on the end date icon and select the <b>Show Start/End</b> context menu option; the display scrolls to show the start date and the start date cell is highlighted.</p>	
<b>Locate resource allocation end date corresponding to allocation start date</b>	<p>Firstly, click on the start date icon for the element:resource combination to highlight it; the corresponding end date icon is also highlighted.</p> <p>If the end date icon is not in view, right-click on the start date icon and select the <b>Show Start/End</b> context menu option; the display scrolls to show the end date and the end date cell is highlighted.</p>	
<b>Refocus display on today's date</b>	<p>Right-click anywhere on the display and select the <b>Show Today</b> context menu option.</p> <p>The display returns to today's date, which is highlighted and outlined.</p>	

#### Notes

- The Project Calendar is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect
- In the above editions, if security is enabled you must have **Manage Project Settings** permission in order to allocate resources to or remove resources from the Project Calendar; if security is not enabled, you can change data without this permission
- In the Lite edition of Enterprise Architect, you can view Calendar entries but not change them
- As the start and end icons for a *completed* task are identical, and one-day tasks have only one icon, the methods to show the corresponding start date or end date for an icon are also useful to show whether the icon *is* for a start or end date, or for a one-day task; alternatively, double-click on the icon and review the allocation dates

#### Learn more

- [Project Calendar](#)<sup>[57]</sup>
- [List of Available Permissions](#)<sup>[329]</sup>

### 3.8.9.3 Project Tasks

**Project Tasks** is a display mode of the project Calendar facility, that you can use to:

- Review the tasks across the project over a period of time
- Review the progress of those tasks
- Create new tasks
- Display and edit the details of the task
- Allocate a resource to a task

- Delete a task
- Create and send an email to the resource assigned to a task
- Create and send an email to the owner of the task

In Project Tasks mode, the Diary panel shows the selected period of time (see the *Project Calendar* topic), with at least one icon representing each *project task* scheduled during that period. The icon represents an independent record of the task - there are no comparisons or validations of the tasks. Therefore you can have separate icons for the same task, with different resources allocated or no resources allocated; you can even have identical task records, if these serve a purpose for you.

Each icon shows the task name and resource name, and indicates the status of the task with one of the following symbols:

- a green square, indicating that the task has been scheduled
- a green tick, indicating that the task is complete
- a red square, indicating that the task is incomplete
- a square divided diagonally with green and red halves, indicating that the task is to be completed within one working day

Typically, while a task is in progress it is represented by:

- An icon with a green square on the day the work is planned to start, and
- An icon with a red square on the day the work is planned to finish

When the work is complete, the icons on both days display the green tick.

Tasks can be created and managed through this view of the Project Calendar, or the:

- Tasks tab of the Project Status window, or
- Project Tasks tab of the Personal Tasks window

**Access** **Project | Calendar > Project Tasks**

### **Manage Project Tasks**

Option	Usage	See also
<b>Create a task</b>	<p>Either:</p> <ul style="list-style-type: none"> <li>• Double-click on the date cell in which the task is to start</li> <li>• Right-click on the cell and select the <b>Add New Task</b> context menu option, or</li> <li>• Click on the down-arrow next to the first icon in the Calendar toolbar, and select the <b>New Task</b> option</li> </ul> <p>The Task Details dialog displays, through which you create the task.</p>	<a href="#">Add, Modify and Delete Tasks</a> <sup>527</sup>

Option	Usage	See also
<b>Display and edit a task</b>	<p>Either:</p> <ul style="list-style-type: none"> <li>• Double-click on the task</li> <li>• Right-click on the cell and select the <b>Properties</b> context menu option, or</li> <li>• Click on the down-arrow next to the first icon in the Calendar toolbar, and select the <b>Properties</b> option</li> </ul> <p>The Task Details dialog displays; if necessary, edit the information.</p>	<a href="#">Add, Modify and Delete Tasks</a> <sup>[527]</sup>
<b>Allocate a resource to a task</b>	<p>Certain tasks might not initially be assigned to a resource, but might later require specific assignment to finish them off.</p> <p>Open the task to edit it, click on the drop-down arrow in the <b>Assigned</b> field and select the required resource.</p>	
<b>Delete a task</b>	<p>Right-click on any icon for the required task and select the <b>Delete '&lt;task-name&gt;'</b> context menu option.</p> <p>You are prompted to confirm the deletion; click on the <b>Yes</b> button.</p>	
<b>Create and send a message to the task owner</b>	<p>Right-click on any icon for the task and select the <b>Message Owner</b> context menu option.</p> <p>The Model Message dialog displays, on which you create and send your message to the user who owns the task (as identified in the Task Details dialog).</p> <p>If the task does not have a defined owner, this option does not operate.</p>	<a href="#">Create a Message</a> <sup>[570]</sup>
<b>Create and send a message to the resource assigned to the task</b>	<p>Right-click on any icon for the task and select the <b>Message Assigned &lt;resource name&gt;</b> context menu option.</p> <p>The Model Message dialog displays, on which you create and send your message to the resource assigned to complete the task (as identified in the Task Details dialog).</p> <p>If the task does not have a defined resource, this option does not operate.</p>	
<b>Locate task start date corresponding to task end date</b>	<p>Firstly, click on the end date icon for the task to highlight it; the corresponding start date icon is also highlighted.</p> <p>If the start date icon is not in view, right-click on the end date icon and select the <b>Show Start/End</b> context menu option; the display scrolls to show the start date and the start date cell is highlighted.</p>	
<b>Locate task end date corresponding</b>	<p>Firstly, click on the start date icon for the task to highlight it; the corresponding end date icon is also highlighted.</p> <p>If the end date icon is not in view, right-click on the start date icon</p>	

Option	Usage	See also
<b>to task start date</b>	and select the <b>Show Start/End</b> context menu option; the display scrolls to show the end date and the end date cell is highlighted.	
<b>Refocus display on today's date</b>	Right-click anywhere on the display and select the <b>Show Today</b> context menu option.  The display returns to today's date, which is highlighted and outlined.	

### Notes

- The Project Calendar is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect
- In the Lite edition of Enterprise Architect, you can view Calendar entries but not change them

### Learn more

- [Project Calendar](#) <sup>[57]</sup>
- [Project Tasks](#) <sup>[526]</sup>
- [Monitor Your Tasks](#) <sup>[559]</sup>

## 3.8.10 Use Case Estimation

Project estimation is the task of working out how much time and effort is required to build and deploy a solution.

The Use Case metrics facility in Enterprise Architect provides a starting point for estimating project effort; using this facility you can get a rough measure of the complexity of a system and some indication of the effort required to implement the model. Like all estimation techniques, Use Case metrics requires some experience with previous projects to 'calibrate' the process.

There is additional information available on Use Case metrics on the Sparx Systems website.

### Topics

Topic	Detail	See also
<b>Calibrating</b>	<p>The following values must be carefully calibrated in order to gain the best possible estimates:</p> <ul style="list-style-type: none"> <li>• Technical Complexity Factors, which are values that attempt to quantify the difficulty and complexity of the work in hand</li> <li>• Environment Complexity Factors, which are values that attempt to quantify non-technical complexities such as team experience and knowledge</li> <li>• Default Hour Rate, which sets the number of hours per Use Case point</li> </ul>	<p><a href="#">Technical Complexity Factors</a> <sup>[585]</sup></p> <p><a href="#">Environment Complexity Factors</a> <sup>[586]</sup></p> <p><a href="#">Default Hours</a> <sup>[588]</sup></p>



Topic	Detail	See also
<b>Estimating</b>	<p>Once you have entered all the calibration values, you can estimate the project timescale through the Use Case Metrics tab of the QA Reports view.</p> <p>The estimation process also draws on information on Use Cases recorded in the Project management window, in particular on the Resource Allocation tab.</p>	<p><a href="#">Estimating Project Size</a> <sup>588</sup></p> <p><a href="#">Resource Allocation</a> <sup>512</sup></p>

#### Learn more

- [Project Estimation Using Use Case Metrics](#)

### 3.8.10.1 Technical Complexity Factors

**Technical Complexity Factors** (TCFs) are used in the Use Case Metrics estimation technique.

The EABase.eap model contains a default set of TCFs, which you can add to or modify using the Estimation Factors dialog. This set of factors should include all factors that could affect the technical complexity of the project environment.

**Access** [Settings](#) | [Project Types](#) | [Estimation Factors](#) > [Technical Complexity Factors](#)

#### Maintain Technical Complexity Factors

Step	Action	See also
<b>1</b>	<p>On the Technical Complexity Factors tab, either:</p> <ul style="list-style-type: none"> <li>• Click on the <b>New</b> button to add another TCF, or</li> <li>• Click on the required factor in the <b>Defined Technical Types</b> list to edit it (go to step 3)</li> </ul>	
<b>2</b>	In the <b>Factor Number</b> field, type the appropriate TCF number.	
<b>3</b>	In the <b>Description</b> field, type or edit the TCF description.	
<b>4</b>	<p>In the <b>Weight</b> field, type or edit the technical complexity weighting.</p> <p>This indicates how much technical complexity you assign to a factor; for example, <i>the system is to be developed in ADA</i> might warrant a higher weight than <i>the system is to be a shell script</i>.</p>	
<b>5</b>	<p>In the <b>Value</b> field, type or edit a value representing the degree of influence the factor has on the project.</p> <p>As a suggested gauge:</p>	

Step	Action	See also
	<ul style="list-style-type: none"> <li>• <b>0</b> indicates no influence</li> <li>• <b>3</b> indicates average influence</li> <li>• <b>5</b> indicates strong influence</li> </ul>	
<b>6</b>	Click on the <b>Save</b> button.	
<b>7</b>	<p>Examine the <b>Defined Technical Types</b> list, and scroll across it to show the <b>Ex Value</b> column (<b>Weight x Value</b>).</p> <p>The summed <b>Ex Values</b> yield the <b>Unadjusted TCF</b> value (at the bottom of the dialog).</p> <p>The <b>Unadjusted TCF</b> value is combined with the Environment Complexity Factors to skew the overall complexity up or down, depending on the level of technical complexity and the corresponding level of environmental support.</p>	<a href="#">Environment Complexity Factors</a> <sup>[586]</sup>

#### Notes

- The TCF Weight evaluates its respective factor, but is irrelevant to a project; the **Value** field assesses each factor's role within a project and, for most purposes, is the only field requiring adjustment
- The supplied factors and their associated weights are defined by the *Use Case Points Method*, although they can be adjusted to suit a project's specific requirements
- You can transport the Technical Complexity Factors between models, using Export Reference Data and Import Reference Data

#### Learn more

- [Export Reference Data](#) <sup>[376]</sup>
- [Import Reference Data](#) <sup>[380]</sup>

### 3.8.10.2 Environment Complexity Factors

**Environment Complexity Factors** (ECFs) are used in the Use Case Metrics estimation technique.

The EABase.eap model contains a default set of ECFs, which you can add to or modify using the Estimation Factors dialog. This set of factors should include all factors that could affect the general design and development environment, including team experience and knowledge, team size, expertise and other non-functional environmental factors.

**Access** [Settings](#) | [Project Types](#) | [Estimation Factors](#) > [Environment Complexity Factors](#)

#### Maintain Environment Complexity Factors

Step	Action	See also
<b>1</b>	On the Environment Complexity Factors tab, either:	

Step	Action	See also
	<ul style="list-style-type: none"> <li>Click on the <b>New</b> button to add another ECF, or</li> <li>Click on the required factor in the <b>Defined Environment Types</b> list to edit it (go to step 3)</li> </ul>	
2	In the <b>Factor Number</b> field, type the appropriate ECF number.	
3	In the <b>Description</b> field, type or edit the ECF description.	
4	In the <b>Weight</b> field, type or edit the environment complexity weighting. This indicates how much complexity you assign to a factor.	
5	In the <b>Value</b> field, type or edit a value representing the degree of influence the factor has on the project. As a suggested gauge: <ul style="list-style-type: none"> <li>0 indicates no influence</li> <li>3 indicates average influence</li> <li>5 indicates strong influence</li> </ul>	
6	Click on the <b>Save</b> button.	
7	Examine the <b>Defined Environment Types</b> list, and scroll across it to show the <b>Ex Value</b> column ( <b>Weight x Value</b> ).  The summed <b>Ex Values</b> yield the <b>Unadjusted ECF value</b> (at the bottom of the dialog).  The <b>Unadjusted ECF</b> value is combined with the Technical Complexity Factors to skew the overall complexity up or down, depending on the level of technical complexity and the corresponding level of environmental support.	<a href="#">Technical Complexity Factors</a> <sup>[585]</sup>

### Notes

- The ECF Weight evaluates its respective factor, but is irrelevant to a project; the **Value** field assesses each factor's role within a project and, for most purposes, is the only field requiring adjustment
- The supplied factors and their associated weights are defined by the *Use Case Points Method*, although they can be adjusted to suit a project's specific requirements
- You can transport the Environment Complexity Factors between models, using Export Reference Data and Import Reference Data

### Learn more

- [Export Reference Data](#) <sup>[376]</sup>

- [Import Reference Data](#)<sup>[380]</sup>

### 3.8.10.3 Default Hours

Setting an hourly rate is the most difficult factor in an accurate estimation. Typical ranges can vary from 10 to 30 hours per Use Case point.

Studying the **Use Case Points Method**, from which this variable is defined, can help you to understand its role in the estimation and facilitate selection of a suitable initial value.

The best way to estimate this value is through analysis of previous completed projects. By calculating the project estimation on a completed project for which the Use Cases and environment are configured within Enterprise Architect, you can adjust the hour rate to render an appropriate value for your unique work environment.

**Access**   **Settings | Project Types | Estimation Factors > Default Hour Rate**

#### Set the default hour rate per adjusted Use Case point

Step	Action	See also
1	In the <b>Duration</b> field, type the number of hours per Use Case Point.	
2	In the <b>Hourly Rate</b> field, type the cost per hour.	
3	Click on the <b>Close</b> button.	

#### Notes

- The values you enter are stored as local settings on your computer only
- This option is also active in the 'Lite', read-only version of Enterprise Architect

#### Learn more

- [Project Estimation Using Use Case Metrics](#)
- [The Read-only 'Lite' Edition](#)<sup>[23]</sup>

### 3.8.10.4 Estimating Project Size

Enterprise Architect uses a simple estimation technique based on the established:

- Number of Use Cases to be built
- Difficulty level of those Use Cases
- Project environment factors and
- Build parameters

This technique is of value only once you have developed a couple of known projects to use as a baseline.

Please DO NOT use the provided 'guesstimates' as a real world measure until you have some real world base lines to measure against.

**Access** Select package in Project Browser:

**Project | QA Reports & Metrics > Use Case Metrics**, or  
**Right-click | Documentation | Package Metrics > Use Case Metrics**

#### Complete a Use Case Metrics Estimation

Field	Usage	See also
<b>Root Package</b>	Confirm the root package in the hierarchy.  All Use Cases under this package could potentially be included in the report.	
<b>Reload</b>	Re-run the load from the selected package, usually after you change the filter criteria.	
<b>Phase like</b>	Include Use Cases with a phase that matches the wildcard value in the field.  Use * to match any characters, for example 1.* for 1.1 and 1.2.	
<b>Keyword like</b>	Include Use Cases with a keyword that matches the wildcard value in the field.  Use * to match any characters.	
<b>Bookmarked</b>	Include all Use Cases, or only those that are tagged, or those that are not tagged.	
<b>Use Cases</b>	Check the total count of Use Cases in the estimate.  The Use Cases and their parameters are listed in the panel underneath this field.	
<b>Include Actors</b>	Select to include Actors in the estimate.	
<b>Technical Complexity Factor</b>	Review the parameters that describe the degree of technical complexity of the project.  While the <b>Unadjusted TCF Value</b> comes from the Technical Complexity Factors tab of the Estimation Factors dialog, the other values compose the Use Case Points Method formula.  Modify these fields with caution.  The final project estimate is directly proportional to the TCF.	<a href="#">Technical Complexity Factors</a> <sup>[585]</sup>

Field	Usage	See also
<b>Environment Complexity Factor</b>	<p>Review the parameters that calculate the degree of environmental complexity of the project, from factors such as programmer motivation or experience.</p> <p>The listed parameters compose the formula calculating the ECF, defined by the Use Case Points Method; the only parameter affected by the project is the <b>Unadjusted ECF Value</b>, derived from the Environment Complexity Factors tab of the Estimation Factors dialog.</p> <p>The final project estimate is directly proportional to the ECF.</p>	<a href="#">Environment Complexity Factors</a>
<b>Unadjusted Use Case Points (UUCP)</b>	Check the sum of the Use Case complexity numbers.	
<b>Ave Hours per Use Case</b>	<p>Check the average number of hours assigned to easy, medium and difficult Use Cases.</p> <p>You cannot change these figures.</p>	
<b>Total Estimate</b>	<p>Review the detailed breakdown of the final figure.</p> <p>You must tailor the hours per Use Case Point figure to the level that matches your type of project and capability based on known previous project outcomes.</p>	
<b>Default Rate</b>	Set the default hours to be fed into the final calculation.	<a href="#">Default hours</a>
<b>Re-Calculate</b>	Re-run the estimate, usually after you change the hours or Use Case point number.	
<b>Report</b>	Produce a rich text formatted report from the current estimate.	
<b>View Report</b>	Display the last-generated report.	

### 3.8.11 Update Package Status

Often a complete Package structure moves from one status to another (such as for release) in one operation. To help facilitate this, Enterprise Architect supports a 'bulk' update of Package and element **Status**, **Phase** and **Version**, which also provides the option of defining the scope of the update.

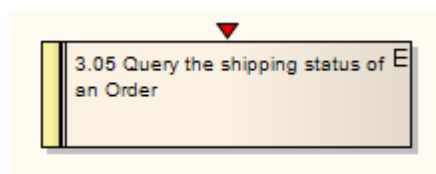
**Access** [Project Browser package context menu | Advanced | Update Package Status](#)

[Update Status across a Package](#)

Field	Action	See also
<b>New Status</b>	Type in or select the new status of the Package.	
<b>New Phase</b>	Type the new phase.	
<b>New Version</b>	Type the new version.	
<b>Modified Date</b>	Defaults to today's date; if the change takes effect on a different date, use the 'spin' arrows to select that date.	
<b>Set Date</b>	Defaults to selected to apply the Modified Date; if necessary, deselect to ignore the date stamp on the change.	
<b>Recursively update all child packages</b>	Select to apply the changes through all child Packages of the selected Package; deselect to apply the changes to only the selected Package.	
<b>Include Elements</b>	Defaults to selected; if the changes are to apply only to Package elements, deselect the checkbox.	
<b>Include Element Requirements</b>	Defaults to selected, to update the element requirements' <b>Status</b> field; deselect to ignore this field.	
<b>Include Element Constraints</b>	Defaults to selected, to update the constraints' <b>Status</b> field; deselect to ignore this field.	
<b>OK</b>	Click on this button to update all required elements to the new status.	

### 3.8.12 Manage Bookmarks

A bookmark is a visual clue that something is different about an element; it is represented by a small red triangle that displays above the element in the diagram.



You can assign any meaning to the bookmark as is appropriate to your model.

You can bookmark:

- Single elements manually, or
- All elements in a package automatically when they assume a defined status

### Guide

To work with bookmarks, select from the options described below:

Option	Usage	See also
<b>Bookmark a selected element in a diagram</b>	Either: <ul style="list-style-type: none"> <li>• Press <b>(Shift+Space)</b>, or</li> <li>• Select the <b>Edit   Bookmark Selected</b> menu option</li> </ul>	
<b>Clear a bookmark on a selected element</b>	Use either of the above two options again.	
<b>Bookmark Multiple Elements in a Package</b>	Select <b>Project Browser package context menu   Bookmarks</b> The Manage Bookmarks dialog displays, enabling you to automatically bookmark any elements (and, if required, their children) in the package that have: <ul style="list-style-type: none"> <li>• New changes defined in the Maintenance window</li> <li>• New defects defined in the Maintenance window, and/or</li> <li>• Test scripts defined in the Testing window</li> </ul> This is useful to highlight elements that have additional project information.	
<b>Clear all elements of bookmarks</b>	To clear all bookmarks: <ul style="list-style-type: none"> <li>• Throughout the current package, click on the <b>Clear All</b> button on the Manage Bookmarks dialog</li> <li>• In the current diagram, select the <b>Edit   Clear All Bookmarks</b> menu option</li> </ul>	

### Notes

- You should reload the project to update it with the new or cleared bookmarks
- The Model Search window enables searching based on bookmarked elements

### Learn more

- [Working on Maintenance Items](#)  <sup>2623</sup>
- [Working on Test Records](#)  <sup>2605</sup>
- [Refresh View of Shared Project](#)  <sup>308</sup>



- [Create & Modify Searches](#) <sup>[709]</sup>

### 3.8.13 Monitor Change Events

You can automatically monitor work events in your project, using the Model Views facility.

#### Guide

Topic	Detail	See also
<b>Model Views</b>	<p>This facility enables you to:</p> <ul style="list-style-type: none"> <li>• Automatically refresh a search in a View at an interval that you define</li> <li>• Notify you if the results of the search change between two consecutive searches</li> </ul> <p>You can therefore use Model Views to monitor various events in the development project, depending on how you set up the search in a View.</p>	<a href="#">Model Views Operations</a> <sup>[693]</sup>
<b>Example</b>	<p>You could set up a search to detect:</p> <ul style="list-style-type: none"> <li>• Change items, or Issue items, so that Enterprise Architect would notify you as new items were created</li> <li>• Element Status, Type, Phase, Version, Priority and/or date of last update, so that Enterprise Architect would notify you as items were progressed to: <ul style="list-style-type: none"> <li>• Fall in to the level of work represented by the search categories or</li> <li>• Move out of the categories into the next level of work</li> </ul> </li> <li>• Tagged Values, so that - again - as items were changed to satisfy the criteria of a sequence of searches, the progression of items through a set of stages could be checked and managed</li> </ul>	
<b>Personalization</b>	<p>People responsible for different stages in a process could have their own Model View searches so that:</p> <ul style="list-style-type: none"> <li>• As a development, validation or authorization task falls due the responsible person is automatically notified, and</li> <li>• When the work is complete both the next person in line and the overseeing manager are notified</li> </ul>	

#### Notes

- This facility is available in the Enterprise Architect Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions

### 3.8.14 The Gantt View

Using the Gantt View, you can visualize elements and assigned resources in a Gantt Chart format, to review the breakdown of work for a specific project. The Gantt View illustrates a project schedule by showing the start and finish dates of assigned resources, so that a Project Manager can quickly see the current project status using the percent-complete bar shading and the percent-complete, resource name and status columns.

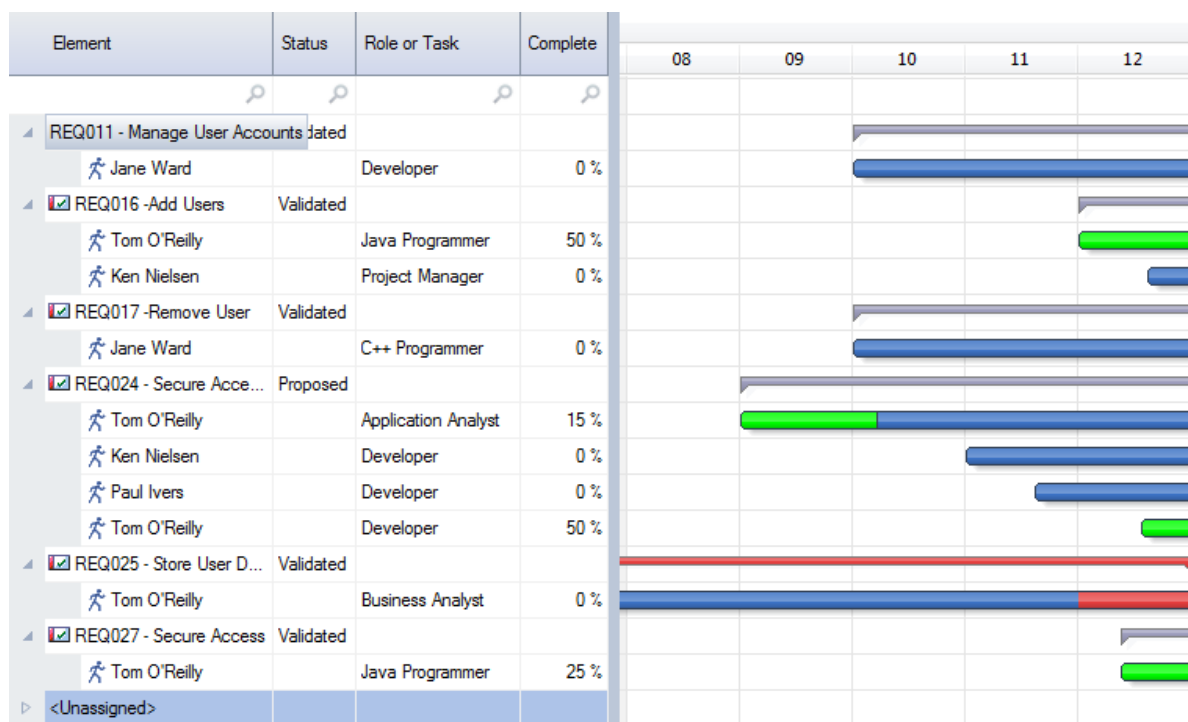
Overdue items can quickly be identified in red by selecting the **Toggle Highlight of Overdue Items** option from the Gantt view's context menu.

There are four different Gantt Views available within Enterprise Architect:

- **Diagram Gantt View** - Show all the elements on a diagram in a Gantt Chart - view or assign resources to them
- **Personal Tasks** - Show all the elements in the model that you have been assigned as a resource to work on
- **Project Gantt View** - Show all the resources in the model and what elements they have been assigned to
- **Package Browser Gantt View** - Show all the elements in a Package and its sub-Packages and see what resources have been assigned to them

**Access** **Right-click on a diagram | Switch to Gantt View**, or  
**View | Personal Tasks**, or  
**Project | Project Gantt View**, or  
**Right-click on a Package in the Project Browser | Package Browser | Gantt View**

#### Gantt View



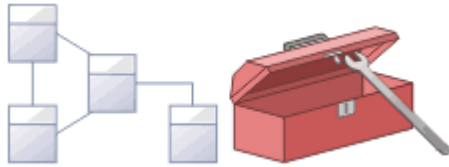
**Notes**

- Items in the list pane can be filtered; right-click on the column headings in the list pane to toggle the Filter Bar between hidden or shown, or to edit the filter

**Learn more**

- [Project Task Allocation](#) <sup>[538]</sup>
- [Resource View](#) <sup>[538]</sup>
- [Element View](#) <sup>[542]</sup>
- [Report View](#) <sup>[546]</sup>
- [Filter Bar](#) <sup>[678]</sup>

### 3.9 Project Maintenance



This topic highlights some administrative functions you might have to carry out to maintain your model.

#### Topics

Topic	Detail	See also
<b>Check the integrity of the data in a project</b>	If you have a failed XML import, network crash or other unforeseen event that could disrupt the integrity of information in the model, it is recommended that you run a Project Integrity Check.	<a href="#">Check Project Data Integrity</a> <sup>[597]</sup>
<b>Reset Auto Increment Columns in Tables</b>	XML Import/Export can affect the table auto increment column values and push them towards the maximum datatype value; you can re-sequence the columns to avoid this problem.	<a href="#">Reset Table Auto Increment or Identity Columns</a> <sup>[598]</sup>
<b>Upgrade an old project to enable use of new features</b>	The structure of Enterprise Architect project files is occasionally changed to support more features; when this happens, existing project files must be upgraded to the new format to ensure correct operation and to take advantage of all the new features.	<a href="#">Upgrade a Project</a> <sup>[599]</sup>
<b>Rename a project</b>	If you want to rename an Enterprise Architect project, you can do so through Windows Explorer.	<a href="#">Rename a Project</a> <sup>[601]</sup>
<b>Compact a project</b>	After some time, a project .EAP file might benefit from compacting to conserve space.	<a href="#">Compact a Project</a> <sup>[601]</sup>
<b>Repair a project if it did not close properly</b>	If a project has not been closed properly, such as during system or network outages, on rare occasions the .EAP file does not re-open correctly.	<a href="#">Repair a Project</a> <sup>[602]</sup>

#### Notes

- You only rename, compact and repair models created as .EAP files; these processes are not required for models stored in a DBMS

### 3.9.1 Check Project Data Integrity

If you have a failed XML import, network crash or other unforeseen event that could disrupt the integrity of information in the model, it is recommended that you run the *Project Integrity Check* function to check that your project data is structurally complete.

You can select a variety of items to check. The integrity check examines all database records and ensures there are no 'orphaned' records or inaccurate or unset identifiers. This function does NOT check UML conformance, only the data relationships and repository structure.

You can run the integrity checker first in report mode to discover if anything should be corrected, and then run it again in 'recover/clean' mode.

When Enterprise Architect 'cleans' the model, it attempts to recover lost packages and elements, and generates a new package at the model root level called *\_recovered\_*. Check through any elements that are found and, if required, drag them into the model proper. If they are not required, delete them.

**Access** [Tools | Data Management | Project Integrity Check](#)

#### Check the data integrity of your project

Step	Action	See also
1	Open the project.	
2	Select the <b>Project Integrity Check</b> menu option. The Project Integrity Check dialog displays.	
3	Select the checkbox for each check to run: <ul style="list-style-type: none"> <li>Package Structure</li> <li>Object Structure</li> <li>Object Features</li> <li>All GUIDs</li> <li>Cross References</li> <li>Connectors</li> <li>UML 2.0 Migration (see below)</li> </ul>	
4	Select either: <ul style="list-style-type: none"> <li>the <b>Report Only</b> option to just view a report on the state of your model, or</li> <li>the <b>Recover/Clean</b> option to recover and clean your project; before selecting this option back up your project file first</li> </ul>	
5	To write a log of the integrity check, click on the <b>Save Results</b> button and select a log file.	

Step	Action	See also
6	<p>Click on the <b>Go</b> button to run the check.</p> <p>If you want to display the resulting information in a more readable layout, you can resize the dialog and its columns.</p>	

### UML 2.0 Migration

The UML 2.0 Migration check enables you to migrate the project from UML 1.3 semantics to UML 2.0 semantics. The migration process currently converts activities that are invocations of operations into called operation actions as per the UML 2.0 specification.

The UML 2.0 Migration option is an exclusive process that does not enable any of the other checks to be selected. When you click on the **Go** button to perform the migration, a prompt displays for you to confirm the operation.

### Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Check Data Integrity** permission to perform a data integrity check

### Learn more

- [List of Permissions](#)<sup>[329]</sup>
- [Run SQL Patches](#)<sup>[601]</sup>

## 3.9.2 Reset Table Auto Increment or Identity Columns

This topic explains possible impacts of XML Export/Import on table auto increment columns, and how to re-sequence the columns whose value approaches the maximum datatype value.

### Topics

Topic	Detail	See also
<b>XML Export/Import</b>	<p>XML Export/Import can cause gaps in the numbering sequence of auto increment columns.</p> <p>Each XML Import deletes rows from several tables; the import then adds rows starting from the maximum value of the auto increment column.</p> <p>Repeated XML imports can result in the value of the auto increment approaching the maximum value of the database datatype; for example, SQL Server's <i>int</i> datatype has a maximum value of <b>2,147,483,647</b>.</p>	
<b>Replication</b>	<p>Large auto increment values can also arise where the project originated as an EAP replica or design master.</p> <p>The Jet engine assigns random values for auto increment columns with each XML Import into the project.</p>	<a href="#">Replication</a> <sup>[310]</sup>

Topic	Detail	See also
	These random values can approach the maximum range of the repository data type, which could present a problem when the EAP project is transferred to a repository.	

**Access** [Tools](#) | [Data Management](#) | [Reset IDs](#)

### How to

To re-sequence auto increment columns

Step	Action
1	Open the project.
2	<p>Select the <b>Reset IDs</b> menu option.</p> <p>A dialog displays listing all non-empty tables that contain an auto increment or identity column.</p> <ul style="list-style-type: none"> <li>The <b>Rows</b> column shows the number of rows in the table</li> <li>The <b>Maximum ID</b> column shows the current maximum value of the auto increment column</li> <li>The <b>Action</b> column shows either <b>No Action</b> or <b>Reset</b>, depending on how close the column value is to the datatype's maximum</li> </ul> <p>Tables requiring a reset are automatically selected in the list.</p>
3	Click on the <b>Go</b> button to reset the auto increment column values.

### Notes

- Project Auditing must to be disabled before auto incremented IDs can be reset.

### Learn more

- [Auditing](#) <sup>446</sup>

## 3.9.3 Upgrade a Project

The structure of Enterprise Architect project files is occasionally changed to support more features. In such cases, the project might have to be upgraded.

Topic	Detail	See also
<b>Upgrading</b>	Upgrading to the new file structure is a simple and quick process that brings your project to the current level to:	

Topic	Detail	See also
	<ul style="list-style-type: none"> <li>• Ensure correct operation and</li> <li>• Support all the latest Enterprise Architect features</li> </ul>	
<b>Initial Check</b>	When you load a project that was created in an early release of Enterprise Architect (for example, an archived project) using a recent release of Enterprise Architect, the system determines whether the project should be upgraded and, if the upgrade is necessary, displays the Upgrade Wizard.	
<b>The Upgrade Project Wizard</b>	<p>The Upgrade Project Wizard takes you through the upgrade process and:</p> <ul style="list-style-type: none"> <li>• Advises you of the necessity to upgrade</li> <li>• Advises you to back up the current project; it is essential to back up before any changes are made</li> <li>• Checks which upgrade path is required</li> <li>• Guides you through the steps to perform the upgrade</li> <li>• Opens the newly converted project</li> </ul>	

#### Notes

- If you are using replication in your models, and the upgrade wizard detects that the project you are opening is a replica and not a Design Master, a different upgrade path is required
- Once upgraded, the project cannot be opened with the version of Enterprise Architect in which it was created

#### Learn more

- [Replication](#)<sup>[310]</sup>
- [Upgrade Replicas](#)<sup>[600]</sup>

### **3.9.3.1 Upgrade Replicas**

Models that have replication features added might have to be upgraded differently from regular projects.

- If the model is a Design Master (the root model of all other replicas) then you can upgrade the model to suit the current version of Enterprise Architect; after upgrading a Design Master you should re-create the replicas, rather than synchronizing
- If the model is not a Design Master, you must first remove the replication features, then upgrade the project in the normal manner; the Upgrade Wizard guides you through the steps

#### Learn more

- [Upgrade a Project](#)<sup>[599]</sup>
- [Replication](#)<sup>[310]</sup>
- [Design Masters](#)<sup>[311]</sup>



### 3.9.4 Run SQL Patches

Occasionally, Sparx Systems might release a patch to correct a model fault.

**Access** [Tools](#) | [Data Management](#) | [Run Patch](#)

The patch generally checks how many records are to be updated, and reports on what is to be done.

### 3.9.5 Rename a Project

If you want to rename a project, you can only do so at the Windows file system level, using Windows Explorer.

#### Rename an Enterprise Architect project file

Step	Action
1	If you have the project open, close it.
2	Ensure no other users have the file open.
3	Open Windows Explorer and navigate to the project.
4	Rename the project file using Windows Explorer.
5	You should keep the file extension to preserve compatibility with the default project type, as installed in the registry at installation time.

### 3.9.6 Compact a Project

After a long period of use and development, a project .EAP file might occupy more disc space than necessary. You can move the project to a local drive and compact the file to recover some of this space.

**Access** [Tools](#) | [Data Management](#) | [Manage .EAP File](#) | [Compact .EAP File](#)

#### Compact a project

Step	Action
1	If you have the project open, shut it down.
2	Ensure no other users have the file open.

Step	Action
3	Select the <b>Compact .EAP File</b> menu option.
4	Follow the on-screen instructions to complete the process.

#### Notes

- Always compact projects on a **local** drive, never on a **network** drive
- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Administer Database** permission to compact a project

#### Learn more

- [List of Available Permissions](#) <sup>329</sup>

### 3.9.7 Repair a Project

If a project has not been closed properly, such as during system or network outages or on poor network connections, on rare occasions the .EAP file does not re-open correctly. A message displays informing you the project is of an unknown database format or is not a database file. In such cases, you can move the project file to a local drive and repair it.

**Access** **Tools | Data Management | Manage .EAP File | Repair .EAP File**

#### Repair a project that was not closed properly

Step	Action	See also
1	Ensure that all users are logged off the project you are attempting to repair.	<a href="#">General Options</a> <sup>605</sup> (page of the Options dialog)
2	Copy the project file to a <b>local drive</b> on your workstation.	
3	In Enterprise Architect, select <b>Tools   Options</b> to display the Options dialog. On the General page, deselect the <b>Use Jet 4.0 - requires restart</b> checkbox.	
4	Close and restart Enterprise Architect - you do not need to open any model, including the one you are repairing.	

Step	Action	See also
5	Select the <b>Repair .EAP File</b> menu option, and follow the on-screen instructions.	
6	Once you have repaired the project, it is recommended that you perform a data integrity check.	<a href="#">Check Project Data Integrity</a> <sup>[597]</sup>

### Notes

- Always repair projects on a **local** drive, never on a **network** drive
- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Administer Database** permission to repair a project

### Learn more

- [List of Available Permissions](#)<sup>[329]</sup>

### 3.10 Local Options




















Enterprise Architect has a wide range of default settings for controlling how the system **displays and works** with models and model elements. You can **customize** these settings to define your preferred behaviour of the system. Most of the settings are stored in your registry so they are set for **your** use only. For a networked workplace, registry settings can be copied down to any network workstation you log in to. Otherwise, the settings control operations under your login on the current **machine** only.

The settings are grouped according to their actions on specific functional areas, on a number of pages of the Options dialog.

**Access**   **Tools | Options (Ctrl+F9)**

#### Option Groups

Group/Page	Action	See also
<b>General Options</b>	Control the behavior of the <b>system</b> as a whole.	<a href="#">General Options</a> <sup>[605]</sup>
<b>Diagram (New Diagram Defaults)</b>	Set the general properties of all diagrams you create.	<a href="#">Diagram Options</a> <sup>[608]</sup>
<b>Diagram - Themes</b>	Create and apply display themes for all diagrams that you open.	<a href="#">Diagram Theme Options</a> <sup>[611]</sup>
<b>Diagram - Standard Colors</b>	Set the display colors and background colors of objects and properties in diagrams across the system.	<a href="#">Standard Colors</a> <sup>[616]</sup>
<b>Diagram - Appearance</b>	Control how elements are displayed on a diagram.	<a href="#">Diagram Appearance Options</a> <sup>[621]</sup>
<b>Diagram - Behavior</b>	Control how a diagram responds to actions taken on it.	<a href="#">Diagram Behavior Options</a> <sup>[625]</sup>
<b>Diagram - Sequence (diagram display)</b>	Control the appearance and behaviour of Sequence diagrams specifically.	<a href="#">Sequence Diagram Options</a> <sup>[629]</sup>
<b>Objects (display)</b>	Configure the properties and behaviors of elements created on your diagrams.	<a href="#">Object Display Options</a> <sup>[631]</sup>

Group/Page	Action	See also
<b>Links (Connector display)</b>	Define the creation, behavior and notation of connectors on your diagrams.	<a href="#">Connector Display Options</a> 
<b>Communication Colors (of Communication Messages)</b>	Set the colors of Messages you create in Communication diagrams.	<a href="#">Communication Message Colors</a> 
<b>XML Specifications</b>	Configure the settings for working with XML.	<a href="#">XML Specifications</a> 
<b>Source Code Engineering</b>	Set general options that apply to all languages when generating code from models, including the default language in which to generate code.	<a href="#">Source Code Engineering</a> 
<b>Code Editors</b>	Configure options for the system internal editor, and for the default editor for DDL scripts.	<a href="#">Code Editors</a> 
<b>Object Lifetimes</b>	Enable the creation of constructors and destructors for Object Lifetimes.	<a href="#">Object Lifetimes</a> 
<b>Attribute/Operations</b>	Set the system actions on attributes and operations.	<a href="#">Attribute/Operations</a> 
<b>Code language specifications</b>	Define how the system handles a particular language when generating code, including the default source directory and the default editor for a specific code language.	<a href="#">ActionScript</a>  <a href="#">C</a>  <a href="#">C#</a>  <a href="#">C++</a>  <a href="#">Delphi</a>  <a href="#">Java</a>  <a href="#">PHP</a>  <a href="#">Python</a>  <a href="#">Visual Basic</a>  <a href="#">VB.Net</a> 

### 3.10.1 General Options

You can control how the system behaves when you open a project, including setting the home directories and web pages the project refers to and how the Project Browser responds when items are selected in it.

**Access**   **Tools | Options (Ctrl+F9) | General**

Control system behavior

Field/Button	Action	See also
<b>Author</b>	Defaults to your user name as the name to which the <b>Author</b> field on the diagram or element Properties dialog defaults when objects are created and modifications are made.  Optionally, click on the drop-down arrow and select another name for the properties <b>Author</b> field to display.	<a href="#">General Tab</a> <sup>[824]</sup> <a href="#">General Settings</a> <sup>[958]</sup>
<b>Clipboard</b>	Click on the drop-down arrow and select the graphic format in which to save images to the Microsoft Windows clipboard; <b>Metafile</b> has the best detail.	<a href="#">Copy Image to Clipboard</a> <sup>[842]</sup> <a href="#">Edit Menu</a> <sup>[80]</sup>
<b>Double click on browser</b>	Defaults to <b>Shows Properties</b> (dialog) as the action the Project Browser takes when you double-click on an item.  Optionally, select the radio button for a different response that you want the Project Browser to make.	<a href="#">Set Default (Project Browser) behavior</a> <sup>[672]</sup>
<b>Use Jet 4.0 - requires restart</b>	Select the checkbox to set JET 4.0 as the database engine; this ensures compatibility with .EAP files that are in turn compatible with versions of MS Access later than Access 97, and that support unicode character sets.  If your project is not in a Jet 4.0 database, you should also download a copy of the <b>Jet 4.0 EABase Model</b> from the Sparx Systems website, and do a <b>.EAP to .EAP transfer</b> of your model into the Jet 4.0 file.  If the checkbox is not selected, the database engine is <b>Jet 3.5</b> .	<a href="#">Link to Jet 4.0 EABase Model</a> <a href="#">Perform a Project Data Transfer</a> <sup>[504]</sup>
<b>Use extended « and » characters</b>	The checkbox defaults to selected to apply the guillemet characters to stereotype names; recommended for double byte character sets.  Deselect the checkbox to hide the guillemet characters.	<a href="#">Stereotype Visibility</a> <sup>[1456]</sup>
<b>Allow custom RTF tags in reports</b>	Select the checkbox to use customized rich text format code in report templates when you generate reports <b>with the Legacy RTF Report Generator</b> .  In releases of Enterprise Architect from <b>7.0</b> onwards, with the notes-formatting facility, this option is not really necessary.	<a href="#">The Legacy RTF Report Generator</a> <sup>[2720]</sup> <a href="#">Notes Toolbar</a> <sup>[1143]</sup>
<b>Project Directory</b>	Type in the default directory location for the Enterprise Architect project files you create.	
<b>File Directory</b>	Type in the default directory name for the shared files you define on the Files tab of an element's Properties dialog (maximum 50 characters).	<a href="#">Associated Files</a> <sup>[988]</sup>

Field/Button	Action	See also
<b>Web Home</b>	Type in the URL of the default home page to open when you click on the <b>Home</b> button in the internal web browser.	<a href="#">The Web Browser</a> <sup>[170]</sup>
<b>Web Search</b>	Type in the URL of the default web page to open when you click on the <b>Web Search</b> button in the internal web browser.	
<b>Web Mail</b>	Type in the email server address ( <b>http://xxxxx/exchange/</b> ) for accessing email within Enterprise Architect, through the web browser.	
<b>Confirm Deletes</b>	The checkbox defaults to selected to display a confirmation prompt whenever you delete a model component.  Only clear this checkbox to bypass the prompt if you are an experienced user and have taken other measures to identify and recover from accidental deletes.	<a href="#">Delete Elements from Diagram and Model</a> <sup>[92]</sup> <a href="#">Delete Diagram</a> <sup>[839]</sup>
<b>Allow Free Sorting</b>	Select the checkbox to allow you to manually reorganize elements within a Package regardless of element type, in the Project Browser.	<a href="#">Order Package Contents</a> <sup>[672]</sup>
<b>Show Stereotypes</b>	The checkbox defaults to selected to display element and feature stereotype names as part of the object names in the Project Browser.  Deselect the checkbox to hide the stereotype names.	<a href="#">Set Default Behavior</a> <sup>[672]</sup>
<b>Always open maximized</b>	The checkbox defaults to selected to set Enterprise Architect to open in a full-screen (maximized) window on start-up.  Deselect the checkbox to set Enterprise Architect to open in a smaller window.	
<b>Hide Properties Info Section</b>	The checkbox defaults to selected to hide the properties information bar at the bottom of the element Properties <b>window</b> .  Deselect the checkbox to expose the information bar.	<a href="#">Properties Window</a> <sup>[992]</sup>
<b>Docked Windows Tabs on Bottom (requires restart)</b>	The checkbox defaults to selected to display any tabs on a <b>docked window</b> at the bottom of the window.  Clear the checkbox to show the window tabs at the top of the window.	
<b>Main Diagram Tabs at Bottom</b>	The checkbox defaults to selected to display the <b>diagram tabs</b> at the bottom of the Diagram View.	

Field/Button	Action	See also
<b>(requires restart)</b>	Clear the checkbox to show the tabs at the top of the Diagram View.	
<b>File Loading Strategy</b>	<p>Click on the drop-down arrow and select the model loading behavior for project files; choose either:</p> <ul style="list-style-type: none"> <li>• <b>Load on Demand (Lazy Load)</b></li> <li>• <b>Preload Entire Model</b></li> </ul> <p><b>Load on Demand</b> does not load the full project view when you load a model; instead, only the parts that are necessary to display the visible portion of the tree are loaded.</p> <p>This means that the model loads faster and you can begin work sooner, but at the expense of later small delays as the system loads specific portions of the model.</p> <p>For server connections, the <b>Load on Demand</b> option is automatically set when the connection is created.</p>	

#### Learn more

- [Local Options](#) <sup>604</sup>

### 3.10.2 Diagram Options

In order to maintain consistency in the format, layout and visibility of objects in your diagrams, you can define a number of default settings to apply to all diagrams that you display, whether you are creating them or reviewing them. You can change or reverse many of these default settings for your view of individual diagrams (whether you created them or another user created them), on the Diagram Properties dialog.



Access   **Tools | Options**   (Ctrl+F9)   | **Diagram**

#### Control diagram set up

Field/Button	Action	See also
<b>Default Page Size</b>	<p>Show (underneath the field name) the default page size for diagrams.</p> <p>You set the default page size using the <b>Page Setup</b> button (below).</p>	
<b>Page Setup</b>	<p>Click on this button to define the page dimensions for diagrams that you create.</p> <p>The Page Setup dialog displays, on which you set the page size, orientation, printer paper tray, and page margins.</p>	<a href="#">Set Up Diagram Page</a> <sup>870</sup>



Field/Button	Action	See also
<b>Print with Border</b>	Select the checkbox to set the default to printing a border on all diagrams you have created.	
<b>Landscape</b>	This checkbox is controlled from the Page Setup dialog. When selected, any diagrams that you create will default to being printed in landscape orientation.	
<b>Show Public Features</b> <b>Show Protected Features</b> <b>Show Private Features</b>	Select any or all of the checkboxes to automatically show Class features with the scope of public, protected and/or private on the diagrams you create.	
<b>Show Diagram Notes</b>	Select the checkbox to display, by default, the diagram details (name, parent package, version and author) in the top left corner of any new diagram that you create.  Selecting this checkbox sets the <b>Show Diagram Details</b> checkbox to selected on the Diagram Properties dialog for the new diagram.	<a href="#">Configure Diagram Display</a> <sup>[825]</sup>
<b>Use Opaque Diagram Labels</b>	Select any or all of the checkboxes to specify which renditions of your diagrams should have opaque diagram labels; <b>Screen</b> and <b>Printing</b> are most often applicable.  An opaque label is a connector or element label that, when the label overlaps the connector line or border, hides the line rather than allowing it to show through the label text.	
<b>Disable fully scoped object names</b>	Select this checkbox to disable the use of fully scoped object names for elements in a diagram; deselect the checkbox when the elements are in their home package.  A scoped object name has the format <i>MyClasses::foo</i> , where the :: character indicates that the Class is within another namespace.	
<b>Allow change of Created Date</b>	Select this checkbox to allow you to change the diagram creation date on any diagram, through the Diagram Properties dialog.	
<b>Zoom to best scale</b>	Select this checkbox to resize all diagrams that you open to neatly fit the monitor screen.	
<b>Auto-pan with middle mouse button</b>	Select this checkbox to turn on auto-panning using the middle mouse button. In this case, when you hold down the middle mouse button a <b>black</b> compass icon displays, and as you move the cursor away from this icon the diagram scrolls in that direction.	

Field/Button	Action	See also
	If you deselect the checkbox, when you hold down the middle mouse button a <b>white</b> compass icon displays, and as you move this icon the whole diagram moves with it across or up and down the screen.	
<b>Enable Connector line jumps</b>	Defaults to selected, so that solid-line connectors show 'hoop' icons or 'jumps' when they cross, on all diagrams that you open.  Deselect to disable line jumps so that connectors simply cross with no 'hoop' icon at the intersection.	<a href="#">Connector Styles</a> 
<b>Scale Saved Bitmaps to</b>	When you save a diagram as a bitmap, you can leave it scaled to 100% or click on the drop-down arrow and select to save the bitmap at a higher <i>resolution</i> of 200% or 400%, suitable for using in published works.	
<b>Image Memory Limit</b>	When you generate images for document or web reports or save images to file, the memory limit for storing this image defaults to 48 MB. If the image is beyond this limit, the system starts to scale down the image.  You can click on the drop-down arrow and select a lower or higher memory limit for the image files of your diagrams. A low memory setting means the system scales the image sooner.	
<b>Diagram Frames</b>	Click on any or all of the checkboxes to specify which saved images of your diagrams will automatically include a diagram frame around them - those saved to disk, those printed out, and/or those saved to the default system clipboard.  A diagram frame is a labeled outline around the diagram image, providing both a border and a reference.  These diagram frames are not the same as the Diagram Frame <b>element</b> in which you can enclose a diagram image or reference when you drag the diagram from the Project Browser onto another diagram.	<a href="#">Diagram Frame</a> 
<b>Show Tooltips</b>	Select the checkbox to display a tooltip when the cursor is 'rolled over' an element on a diagram you have opened.  If you have selected the <b>Show Tooltips</b> checkbox, select the checkbox against each item of information to be displayed as part of a tooltip.  If you select the <b>Notes</b> checkbox, consider whether the <b>Max Characters</b> value is appropriate to represent the text of your element <b>Notes</b> fields. This field defines the maximum number of characters of notes text included in the tooltip; if you want to show less text or more, reset the character limit accordingly.	

Field/Button	Action	See also
<b>Close</b>	Click on this button to save the changes you have made, and to close the Options dialog.	

### Learn more

- [Local Options](#) <sup>[604]</sup>
- [Set Diagram Properties](#) <sup>[823]</sup>
- [Diagram Theme Options](#) <sup>[611]</sup>
- [Diagram Gradients and Backgrounds](#) <sup>[613]</sup>
- [Standard Colors](#) <sup>[616]</sup>
- [Diagram Appearance Options](#) <sup>[622]</sup>
- [Diagram Behavior Options](#) <sup>[625]</sup>
- [Sequence Diagram Options](#) <sup>[629]</sup>

### 3.10.2.1 Diagram Theme Options

It is possible to apply a **theme** to all diagrams in the project that you display on your workstation; that is, how the diagrams and their contents appear when you display them in the Diagram View. By applying a theme you can change the color, font, gradient, line thickness and background image (**tile**). Each design helps bring your diagrams to life with brighter colors, better contrast and stylish new designs. As you select a theme, the appearance settings are immediately applied to any diagram that you have open. Enterprise Architect includes a number of basic built-in themes, which are useful to a wide range of users and are good examples of how a theme can enhance the diagrams that you have open.

You can also create your own themes, capturing the set of colors, text font, line thicknesses and/or gradient properties that you have **currently** defined in the Options dialog, on the:

- Gradients and Background page
- Standard Colors page
- Appearance page, and
- Links page

You select or deselect a checkbox for each set of properties you want to include or exclude, from the Options dialog pages.

**Access** **Tools | Options (Ctrl+F9) | Diagram | Themes**

### Configure themes for diagrams and elements

Field/Option/ Button	Action	See also
<b>Name</b>	Click on the drop-down arrow and select the theme you want to either: <ul style="list-style-type: none"> <li>• Apply as the default appearance of your diagrams</li> </ul>	

Field/Option/ Button	Action	See also
	<ul style="list-style-type: none"> <li>Edit (user-defined theme)</li> <li>Use as the base for a new user-defined theme, or</li> <li>Delete (user-defined theme)</li> </ul>	
<b>Delete</b>	<p>Click on this button to delete the selected user-defined theme.</p> <p>A prompt displays for you to confirm the deletion. Click on the <b>Yes</b> button.</p> <p>The <b>Name</b> field defaults to <b>High Contrast White</b>; click on the drop-down arrow and select another theme to apply instead of the deleted theme.</p>	
<b>Font</b>	<p>Select the check box to include the User Font <b>Font Face</b> and <b>Font Size</b> settings from the Appearance page of the Options dialog (click on the <b>Configure Default Element Fonts</b> button on the Appearance page to set these fields).</p> <p>Deselect the check box to retain the previous font applied <b>to the diagram</b>.</p>	<a href="#">Diagram Appearance Options</a> <sup>[622]</sup>
<b>Color</b>	<p>Select the check box to include the element, connector and diagram colors defined on the Gradients and Background page and the Standard Colors page.</p> <p>Deselect the checkbox to retain the previous set of colors applied <b>to the diagram</b>.</p>	<a href="#">Diagram Gradients and Backgrounds</a> <sup>[613]</sup> <a href="#">Standard Colors</a> <sup>[616]</sup>
<b>Line Thickness</b>	<p>Select the check box to include the <b>Default Element Border Width</b> from the Appearance page, and the <b>Pen Width</b> (for connector line thickness) from the Link page of the Options dialog.</p> <p>Deselect the checkbox to retain the previous set of line widths applied <b>to the diagram</b>.</p>	<a href="#">Diagram Appearance Options</a> <sup>[622]</sup> <a href="#">Links</a> <sup>[635]</sup>
<b>Gradient</b>	<p>Select the check box to apply the setting of the <b>Show Gradient Fill for Paper Color</b> checkbox (which applies a color gradient to the background color of the diagrams you display) and the <b>Gradient Fill Direction for an Element</b> field on the Gradients and Background page of the Options dialog.</p> <p>Deselect the checkbox to retain the previous color gradient settings applied <b>to the diagram</b>.</p>	<a href="#">Diagram Gradients and Backgrounds</a> <sup>[613]</sup>
<b>Background Image</b>	<p>Select the checkbox to incorporate the currently-set diagram background image in the theme.</p>	<a href="#">Diagram Gradients and Backgrounds</a> <sup>[613]</sup>
<b>Save</b>	<p>Click on this button to either:</p>	

Field/Option/ Button	Action	See also
	<ul style="list-style-type: none"> <li>Create a new user defined theme, or</li> <li>Save changes to an existing theme</li> </ul> <p>The Diagram Theme dialog displays for the theme name. Either:</p> <ul style="list-style-type: none"> <li>Type in a name for the theme and click on the <b>OK</b> button to create the new theme, or</li> <li>Simply click on the <b>OK</b> button and then click on the <b>Yes</b> button to save the updates to the theme under the same theme name</li> </ul>	
<b>Close</b>	<p>Click on this button to close the Options dialog.</p> <p>If you have <b>edited</b> an existing theme, this applies all changes to the theme to the diagrams you have open.</p>	

**Notes**

- You cannot edit or delete a **built-in** theme; you can use a built-in theme as the base for a user-defined theme
- A theme draws its properties from other pages of the Options dialog, therefore if you select the **Gradient** checkbox the theme will apply the status of the **Show Gradient Fill for Paper Color** checkbox on the Gradients and Background page, when that status might be **disabled** because you have applied a background tile image

**3.10.2.2 Diagram Gradients and Backgrounds**


When reviewing the diagrams in your model, you have a range of options for managing the display and appearance of the diagrams and their contents. These options include setting the **color** and **shading** of the diagram and elements, and the style of the page border of the diagram. You can also, instead of using the color wash, select a **tiled** background image for the diagrams you display, from a set of system-supplied images or from images that you have created or downloaded to a local system folder yourself.

**Access**    **Tools | Options (Ctrl+F9) | Diagram | Gradients and Background**

**Set diagram backgrounds and shading**

Field/Option/ Button	Action	See also
<b>Diagram Background</b>		
<b>Tile</b>	<p>Either:</p> <ul style="list-style-type: none"> <li>Type in the file path for a .jpg, .bmp or .png graphics file to use as a background for all your diagrams, or</li> <li>Click on the <b>Select Tile</b> button to the right of the field and select a file (by default from the Enterprise Architect supplied tiles folder, but you can browse to any other file location for the required file)</li> </ul>	

Field/Option/ Button	Action	See also
	<p>The selected background is immediately applied to any diagrams you have open, and the field displays the location of the background graphic file, such as:</p> <p style="padding-left: 40px;">C:\Program Files\Sparx Systems\EA\BackgroundTiles\Aqua_03.jpg</p> <p>If you no longer want to apply a background tile to your diagrams, click on the <b>Clear File</b> button to clear the field.</p>	
<b>Paper</b>	<p>Click on the drop-down arrow to change the <b>background</b> color of the <b>diagrams</b> you display.</p> <p>This field is disabled if you have selected a background tile (above). The field is re-enabled when the file path is deleted from the <b>Tile</b> field.</p>	
<b>Page Border Style</b>	<p>Click on the drop-down arrow and select the option for the type of page border you want to use when displaying diagrams on your monitor. You can choose from:</p> <ul style="list-style-type: none"> <li>• <b>Solid</b> (the default) - pages are framed by a solid dark gray line on the screen</li> <li>• <b>Dash</b> - pages are framed by a broken dark gray line</li> <li>• <b>Dash (with white line)</b> - pages are framed by a broken dark gray line on a solid pale background</li> <li>• <b>Dot</b> - pages are framed by a dotted dark grey line</li> <li>• <b>Dot (with white line)</b> - pages are framed by a dotted dark gray line on a solid pale background line</li> <li>• <b>Chiselled</b> - pages are framed by a thin black line bordered by a thicker dark gray line to the right or below</li> </ul> <p>If you cannot see the page border on a diagram, right-click on the diagram and select the <b>Properties</b> option, then check that the <b>Hide Page Border (All Diagrams)</b> option is <i>not</i> selected and that the <b>Show Page Border (Current Diagram)</b> is selected.</p>	<a href="#">Configure Diagram Display</a> <sup>825</sup>
<b>Show Gradient Fill for Paper Color</b>	<p>Select the checkbox to apply a color <b>gradient</b> to the background color of the diagrams you display. You set the background color itself in the <b>Paper</b> field (above).</p> <p>Deselect the checkbox to apply the background color uniformly to the diagram background.</p> <p>This checkbox is disabled if you have selected a background tile (above). The checkbox is re-enabled when the file path is deleted from the <b>Tile</b> field.</p>	
<b>Element Fill</b>		

Field/Option/ Button	Action	See also
<b>Gradient Fill Direction For an Element</b>	<p>Click on the drop-down arrow and select the <b>direction</b> of the color gradient applied to the <b>element</b> fill color.</p> <p>Select <b>&lt;none&gt;</b> to apply the element fill color uniformly to the element, with no color gradient.</p>	
<b>Gradient Style for Element</b>	<p>Click on the drop-down arrow and select the <b>style</b> to apply to the element fill color gradient. The options apply:</p> <ul style="list-style-type: none"> <li>On the left or upper edge of each element, a band of the gradient color that fades in the fill direction defined above, and that has an angled line or curve on the inner boundary</li> <li>On the right or lower side of each element, a band of the gradient color that fades at 90 degrees to the gradient direction defined above</li> </ul> <p>Select <b>&lt;none&gt;</b> to have a single band of color fading across the element in the fill direction (above).</p>	
<b>Fill</b>	<p>Click on the drop-down arrow and select the <b>fill</b> color of all <b>elements</b> (except Screen elements and Notes) on the diagrams you display.</p> <p>You can also change the element fill color on the Standard Colors page, where you can define the fill color of Screen and Notes elements as well.</p> <p>You can quickly make fill color changes using the <b>Hue</b>, <b>Saturation</b> and <b>Luminosity</b> fields, below, in combination. The <b>Fill</b> field reflects the color changes as you make them, but having a diagram open with large elements also shows the changes immediately and very clearly.</p>	<a href="#">Standard Colors</a> 
<b>Revert</b>	Click on this button to reset the fill color to the previously saved color. If you have been testing new colors but have not closed and re-opened the Options dialog, you can return to the color you originally changed from.	
<b>Hue</b>	Click on the slider icon and drag it along the line to set the basic primary color or combination of two primary colors. The numerical value of the color (the <b>H</b> value in the HSL code) displays in the field to the right of the slider.	
<b>Saturation</b>	Click on the slider icon and drag it along the line to set the color saturation (the degree of grayness in the color). The numerical value of the saturation (the <b>S</b> value in the HSL code) displays in the field to the right of the slider.	
<b>Luminosity</b>	Click on the slider icon and drag it along the line to set the color luminosity (the degree of light in the color, from none (black) to full (white)). The numerical value of the luminosity (the <b>L</b> value in the HSL code) displays in the field to the right of the slider.	

Field/Option/ Button	Action	See also
<b>Gradient Options</b> (ineffective if the <b>Gradient Fill Direction For an Element</b> field is set to <none>)		
<b>Tint Amount</b>	<p>Click on the slider icon and drag it along the line to set the degree of tint in the color gradient. Moving from the left end the slider sets completely black at the top of the gradient, through no tint to completely white at the top of the gradient.</p> <p>The percentage tint displays in the field to the right of the slider, showing <b>-50%</b> for the left end of the slider, <b>0%</b> (the default) in the center and <b>50%</b> at the right end.</p> <p>To reset the <b>Tint Amount</b> to the system default (which is a slight, dusky gradient), click on the <b>Default</b> button.</p>	
<b>Fill Amount</b>	<p>Click on the slider icon and drag it along the line to set the degree of fill of the gradient. From the left end the slider sets, effectively, no gradient (<b>0%</b>) through to starting the gradient immediately at one edge of the element and building right across the element (<b>100%</b>).</p> <p>The percentage gradient fill displays in the field to the right of the slider.</p> <p>To reset the <b>Fill Amount</b> to the system default (<b>80%</b>), click on the <b>Default</b> button.</p>	
<b>Close</b>	Click on this button to save the changes you have made, and to close the Options dialog.	

#### Learn more

- [Create Custom Diagram Background](#) 

### 3.10.2.3 Standard Colors

On a model diagram, the objects and texts are rendered in the system default colors. You can **change** any of these default colors to either one of a range of standard colors, or a color that you define yourself. Another user viewing the same model diagrams will still see the diagrams in either the standard default colors or colors **that** user has selected for their own use.

If you define and select a 'non-standard' color for an object type, that color is shown in the color pallet whilst it is the selected color for that object type. If you then select a different color, the non-standard color is no longer available in the pallet. You can, however, define up to 16 **custom** colors that are **permanently available** to use in changing the appearance of individual elements, through the Default Appearance dialog and diagram Format Toolbar.

**Access**   **Tools | Options (Ctrl+F9) | Diagram | Standard Colors**

#### Set the Color of an Object Type

The top 14 fields on the Standard Colors page each shows a color strip displaying the currently-set color for the corresponding type of object or feature, and a down-arrow.



Step	Action
1	<p>On the selected field, click on the down arrow.</p> <p>A color pallet displays, showing a small range of standard colors and, if the object is currently set to a <b>non-standard</b> color, an additional square to the right of the <b>Other</b> button to show that color.</p>
2	<p>If the color you want to use is shown on the pallet, click on it to select it and to return to the Standard Colors page.</p>
3	<p>If the color you want to use is not shown, click on the <b>Other</b> button.</p> <p>The Color dialog displays showing the Standard tab, offering a larger range of standard colors.</p>
4	<p>If the color you want to use is shown on the pallet, click on it to show that color in the New panel and (optionally) compare it with the currently-set color in the Current panel.</p> <p>Click on the <b>OK</b> button to set the selected color and to return to the Standard Colors page.</p>
5	<p>If a suitable color is not available in the extended pallet, click on the Custom tab. This contains:</p> <ul style="list-style-type: none"> <li>• A spectrum panel, with a 'sampler' icon; move the icon around the panel to change the color shown in the New panel</li> <li>• A Luminosity bar; move the arrow on the right up and down to change the intensity of the color shown in the New panel</li> <li>• <b>Hue</b>, <b>Sat</b>(uration) and <b>Lum</b>(inosity) (HSL) fields, and <b>Red</b>, <b>Green</b> and <b>Blue</b> (RGB) fields; type and increment or decrement the decimal values to either: <ul style="list-style-type: none"> <li>• make slight modifications to a color or</li> <li>• exactly define a color for which you know the HSL and/or RGB values</li> </ul> </li> </ul> <p>As you define the color it changes in the New panel; (optionally) compare it with the currently-set color in the Current panel and adjust it as necessary.</p> <p>When you have defined your color, click on the <b>OK</b> button to select it and to return to the Standard Colors page.</p>
6	<p>As an alternative to selecting a standard color or defining a new one, you can capture a color that is used elsewhere on your <b>current</b> monitor screen.</p> <p>Click on the <b>Select</b> button to redisplay the cursor as an 'eye-dropper' icon; move this around the screen to the color you want to sample.</p> <p>Position the cursor so that the color is under the tip of the 'eye-dropper' and shown in the New panel.</p> <p>Click the left mouse button to capture the color, then click on the <b>OK</b> button to select it and to return to the Standard Colors page.</p>
7	<p>Click on the <b>Close</b> button to save your changes and close the Options dialog.</p>

**Object Types for which to set Colors on all diagrams you display**

Field	Effect	See also
<b>Element</b>		
<b>Fill</b>	<p>Define the <b>fill</b> color of all <b>elements</b> (except Screen elements and Notes - see below) on the diagrams you display.</p> <p>This color can also be <b>changed</b> on the Gradients and Backgrounds page, and <b>overridden</b> by any other fill color definition in your model, such as the Format Toolbar fill setting, element default appearance, Template Element package or stereotype definition.</p>	<a href="#">Diagram Gradients and Backgrounds</a> <sup>[613]</sup>
<b>Line</b>	Define the <b>line</b> color of all element <b>borders</b> on the diagrams you display.	
<b>Text</b>	Define the <b>font</b> color of <b>element text</b> and labels on the diagrams you display (including Text elements).	
<b>Shadows</b>	Define the color of element <b>outline shadows</b> on the diagrams you display (including Notes).	
<b>Attributes</b>	Define the <b>font</b> color of <b>attribute text</b> in the attribute <b>compartment</b> of all elements on the diagrams you display.	
<b>Methods</b>	Define the <b>font</b> color of method ( <b>operation</b> ) <b>text</b> in the operation <b>compartment</b> of all elements on the diagrams you display.	
<b>Behaviors</b>	Define the color for <b>behaviors</b> in <b>Activity</b> diagrams you display.	
<b>Notes</b>	Define the color of <b>text</b> in the element <b>notes compartment</b> of all elements on the diagrams you display.	
<b>Screen</b>	Define the <b>fill</b> color of <b>Screen</b> elements on <b>User Interface</b> diagrams you display.	
<b>Note &amp; Constraint Elements</b>		
<b>Fill</b>	Define the <b>fill</b> color of <b>Note</b> and <b>Constraint</b> elements on the diagrams you display.	
<b>Text</b>	Define the <b>font</b> color of Note and Constraint <b>text</b> on the diagrams you display.	

Field	Effect	See also
<b>Note Line</b>	Define the <b>line color</b> of the borders on <b>Note</b> elements on the diagrams you display.	
<b>Constraint Line</b>	Define the <b>line color</b> of the borders on <b>Constraint</b> elements on the diagrams you display.	
<b>Others</b>		
<b>Connector Lines</b>	Define the <b>line color</b> of all <b>connectors</b> on the diagrams you display.	

### Notes

- To override the default appearance of a **specific** element on **all diagrams** on which it is found, right-click on the element and select the **Appearance | Default Appearance** context menu option; the Default Appearance dialog displays
- To change the appearance of a specific element on the **current** diagram only, use the Format Toolbar; which automatically displays at the top of every diagram.

### Learn more

- [Create Custom Colors](#)<sup>[619]</sup>
- [Format Toolbar](#)<sup>[785]</sup>

#### 3.10.2.3.1 Create Custom Colors

If the standard colors available through the color palettes do not meet your requirements, you can define up to 16 custom colors that you apply to individual elements and connectors through the diagram Format Toolbar, Connector Appearance dialog and element Default Appearance dialog.

By applying such custom colors within a template diagram, you can render all elements of the same type with the same custom color, for all users of the project.

If another user has a set of custom colors that you want to use, or a project manager has defined colors specifically for the project, they can export these colors to the project file and you can then **import** them directly as your custom color pallet. If you have your own colors in the pallet, they are overwritten by the imported set.

**Access** **Tools | Options (Ctrl+F9) | Standard Colors: Define Colors**

### Define your custom colors

Step	Action	See also
<b>1</b>	On the Color dialog, click on the <b>Define Custom Colors »</b> button.	

Step	Action	See also
	The Color dialog is extended to provide color selection fields on the right of the dialog.	
2	<p>Click on an empty block in the <b>Custom colors</b> pallet (or, if you are <i>replacing</i> a color, click on that color block in the pallet) and on the:</p> <ul style="list-style-type: none"> <li>• Spectrum panel, move the 'sampler' icon around the panel to change the color shown in the <b>Color Solid</b> box</li> <li>• Luminosity bar, move the arrow on the right up and down to change the intensity of the color shown in the <b>Color Solid</b> box</li> <li>• <b>Hue</b>, <b>Sat</b>(uration) and <b>Lum</b>(inosity) (HSL) fields, and <b>Red</b>, <b>Green</b> and <b>Blue</b> (RGB) fields; type decimal values to either: <ul style="list-style-type: none"> <li>• make slight modifications to a color or</li> <li>• exactly define a color for which you know the HSL and/or RGB values</li> </ul> </li> </ul> <p>As you define the color it changes in the <b>Color Solid</b> box.</p>	
3	When you have finished creating the color, click on the <b>Add to Custom Colors</b> button to add the color to the selected <b>Custom colors</b> block.	
4	Repeat steps 2 and 3 for any other custom colors you want to define.	
5	Click on the <b>OK</b> button to close the Color dialog.	
6	<p>To make the custom colors available for your use, select the <b>Show Project Custom Colors in Color Combo Boxes</b> checkbox.</p> <p>The 16 squares dedicated to custom colors are displayed at the end of the color selection pallets on the diagram Format Toolbar, Connector Appearance dialog and element Default Appearance dialog.</p>	<a href="#">Format Toolbar</a> <a href="#">Set an Element's Default Appearance</a> <a href="#">Connector Appearance Options</a>
7	Click on the <b>Close</b> button to close the Options dialog.	

### Notes

- For changes to the custom colors to be reflected in the Format Toolbar pallets, you might need to close and re-open the model and reset the Toolbar (**Tools | Customize > Toolbars**, select the **Format Tool** checkbox and click on the **Reset** button)
- If you want to export your colors to the project file to make them available for others to import, use the **Settings | Colors | Set Project Colors** menu option
- If you want to import someone else's colors from the project file as your custom colors, use the

**Settings | Colors | Get Project Colors** menu option

[Learn more](#)

- [Get/Set Custom Project Colors](#)<sup>[62]</sup>

### 3.10.2.3.2 Get/Set Custom Project Colors

If more than one person is working on a project, you might want to share a set of custom colors with other team members, or possibly provide special colors for use across the whole project. You would then:

- Capture the custom or project colors you have defined, in the project file, or
- Retrieve a set of colors someone else has defined, from the project file



**Access**    **Settings | Colors | Set Project Custom Colors**  
**Settings | Colors | Get Project Custom Colors**

#### Set custom project colors in the project file

Step	Action	See also
1	Define the custom colors on the Standard Colors page of the Options dialog.	<a href="#">Create Custom Colors</a> <sup>[61]</sup>
2	Select the <b>Set Project Custom Colors</b> option. The following message displays: <i>Project Custom Colors export successful. 16 custom colors saved to the project file.</i>	
3	Click on the <b>OK</b> button. Notify the appropriate users that the colors are available for importing.	

#### Get project custom colors from the project file

Step	Action	See also
1	Select the <b>Get Project Custom Colors</b> option. The following message displays: <i>Project Custom Colors import successful. 16 custom colors imported from the project file.</i>	
2	The custom colors are now available at the end of the color pallets of the Default Appearance dialog, of the Connector Appearance dialog and of the Font Color, Fill Color and Line Color drop-downs in the diagram Format Toolbar.	<a href="#">Set an Element's Default Appearance</a>


Step	Action	See also
	If the colors do not show, select <b>Tools   Options   Standard Colors</b> and select the <b>Show Project Custom Colors in Color Combo Boxes</b> checkbox.	 <a href="#">Format Toolbar</a> 

### 3.10.2.4 Diagram Appearance Options

It is possible to configure the appearance of diagrams and elements in your models; that is, how diagrams and their contents appear when displayed in the Diagram View. There is a wide range of options you can set to create this display configuration. As you select an option, it has immediate effect on any diagram that you have open, so you can see what the change in appearance is.

**Access** **Tools | Options (Ctrl+F9) | Diagram | Appearance**

#### Configure the appearance of diagrams and elements

Field/Button	Action	See also
<b>Configure Default Element Fonts</b>	Click on this button to set the default text fonts to use on diagrams either across the model or that you have created yourself. A font defined across the model overrides the font for diagrams you create yourself.	<a href="#">Set Default Fonts</a> 
<b>Renderer</b>	<p>Render smooth curves and diagonal lines in diagrams, so that staggered vertical or horizontal pixels are less obvious. Click on the drop-down arrow and select the type of renderer you prefer to use:</p> <ul style="list-style-type: none"> <li><b>Basic</b> is GDI32; it does not provide anti-aliasing and gradient fills</li> <li><b>Enhanced-1</b> is parallel to Windows GDI+ but internal to Enterprise Architect; it provides anti-aliasing and gradient fills, and operates well across different platforms</li> <li><b>Enhanced-2</b> is Windows GDI+; this can vary across different platforms, performing better than <b>Enhanced-1</b> in some environments, and less well in others</li> </ul> <p>Experiment with these options and see which works best for your system and requirements.</p>	
<b>GDI Plus Metafiles</b>	<p>This checkbox defaults to selected, to include the use of GDI+ metafiles in your diagrams.</p> <p>Deselect the checkbox if you do not want to use GDI+ metafiles.</p>	
<b>Default Element Border Width</b>	Click on the drop-down arrow and select the default element border width (1, 2 or 3 pixels).	

Field/Button	Action	See also
<b>Scale view by</b>	<p>Click on the drop-down arrow and select the percentage (from 0 to 50, in increments of 10) by which you can automatically increase the size of all objects on a diagram you are viewing, without affecting other users reading that diagram.</p> <p>You can perform the same function with the <b>Zoom Slider</b> on the Status bar; changes in the 'zoomed' display scale of a diagram update this field and affect any other diagrams that you open.</p> <p>This has no impact any other diagram Zoom facility in the system.</p>	<a href="#">Status bar</a> <sup>1501</sup>
<b>Print in Color</b>	<p>Select the checkbox to print your diagrams in color.</p> <p>Deselect the checkbox to print the diagrams in black and white.</p>	
<b>Anti-aliased text</b>	<p>Select the checkbox to force text anti-aliasing in diagrams.</p> <p>If you deselect the checkbox, the system applies the MS Windows default setting. Therefore, if you do not want to use anti-aliasing, ensure that the Windows anti-aliasing default is <b>also</b> set to <b>OFF</b>.</p>	
<b>Compress text to fit within Element</b>	<p>Select the checkbox to compress element text within the current element boundaries, when text at <i>zoom</i> levels other than 100% would not fit inside those boundaries.</p> <p>Deselect the checkbox to expand the element boundaries under those conditions.</p>	
<b>Italic Note Element Text</b>	Select the checkbox to display the text of Note elements in italics.	
<b>Use Element Group Style</b>	<p>Select the checkbox to apply a different fill color to each object type, with objects derived from a common element having the same color (for example, Object, Boundary, Control and Lifeline elements are the same color).</p> <p>Overrides all other settings for element fill colors.</p>	
<b>Bold Element Names</b>	Select the checkbox to display element names in bold text.	
<b>Element Shadows On</b>	Select the checkbox to display a shadow around the bottom and right edges of each element in a diagram.	
<b>Invert rotated text for metafiles</b>	Select the checkbox to use a different text format when external metafile readers are causing issues.	
<b>Round edges for elements</b>	Select the checkbox to soften the corners of all elements that have a rectangular appearance, so that they have rounded corners.	

Field/Button	Action	See also
<b>Hand Draw Mode Font</b>	Click on the drop-down arrow and select the font to be used in any diagram you display that has the <b>Hand Drawn</b> diagram option selected.	<a href="#">Configure Diagram Display</a> <sup>825</sup>
<b>Default</b>	Click on this button to re-set the font you see used in hand-drawn diagram mode to the system default (Segoe Print).	
<b>Italic</b>	For diagrams displayed in Hand Draw mode, select this checkbox to display text in italics.	
<b>Use Watermark</b>	Select the checkbox to add a watermark to any diagrams that you print.	
<b>Text</b>	Type in the text to use as watermark text. If the <b>Use Watermark</b> checkbox is selected, this text is displayed on any diagrams that you print.	
<b>Close</b>	Click on this button to save the changes you have made, and to close the Options dialog.	

#### 3.10.2.4.1 Set Default Fonts

A diagram in your model uses a standard font that is applied either to all diagrams every user displays across the model, or to any diagrams you display that you created yourself.

You can define both Model and User fonts, but the Model font overrides the User font, to ensure that all members of the project team have a consistent and coherent view of the model. This avoids the problem of one user creating a diagram in a small font, and another user trying to view it in a larger font, which distorts the diagram.

It is recommended that a project authority sets the Model font, and all project members abide by it and do not change it without project approval. In editions where security is enabled, changes can be restricted by allocating permission only to the project authority.

**Access** **Tools | Options (Ctrl+F9) | Diagram | Appearance: Configure Default Element Fonts**

#### Control default diagram fonts

Field/Button	Action	See also
<b>Model Font : Font Face</b>	Click on the drop-down arrow and select the <b>font typeface</b> to apply across the model.	



Field/Button	Action	See also
<b>Model Font : Font Size</b>	Click on the drop-down arrow and select the <b>font size</b> to apply across the model.	
<b>Clear</b>	Click on this button to clear the model <b>Font Face</b> and <b>Font Size</b> fields.	
<b>User Font: Font Face</b>	Click on the drop-down arrow and select the <b>font typeface</b> to apply to <b>your</b> diagrams if no model font is specified.	
<b>User Font: Font Size</b>	Click on the drop-down arrow and select the <b>font size</b> to apply to <b>your</b> diagrams if no model font size is specified.	
<b>Restore Defaults</b>	Click on this button to restore the system defaults to the user <b>Font Face</b> and <b>Font Size</b> fields.	

#### Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Project Settings** permission to set, change or clear the **Model Font** fields
- Both model and user fonts are overridden by specifically-defined element fonts, so that the element is viewed as designed regardless of the model or user defaults; to define the font for a specific element, right-click on the element in a diagram and select the **Appearance | Set Font** context menu option
- If you cannot read the diagrams because the default font makes the objects and text too small, you can scale up all objects (that is, all diagram displays) to a more readable size for you only; the objects are not scaled up for other users

Everything on the diagram is enlarged to the same extent, so it remains in proportion and readable; to do this, return to the Diagram Appearance page of the Options dialog and enter a suitable percentage value in the **Scale view by** field

#### Learn more

- [Set Element Font](#)<sup>[949]</sup>
- [Diagram Appearance Options](#)<sup>[622]</sup>
- [List of Available Permissions](#)<sup>[329]</sup>

### 3.10.2.5 Diagram Behavior Options

As you create and edit diagrams in your model, the system can respond in different ways to the types of object you are adding to a diagram. You can configure a range of options to control what response the system makes to the object you are adding or the action you are performing.

**Access**   **Tools | Options (Ctrl+F9) | Diagram | Behaviour**

Control diagram behavior

Field/Button	Action	See also
<b>Auto Instance</b>	<p>Select the checkbox to automatically create <b>object instances</b> when dragging certain element types - such as Class and Component - from the Project Browser onto the diagram, with the dragged element as the classifier.</p> <p>Deselect the checkbox to paste the element as a link.</p>	<a href="#">Paste from Project Browser</a> 833
<b>Instance has Classifier style</b>	<p>Select the checkbox to automatically apply the <b>classifier style</b> of the element from which an instance is instantiated, when the instance is created either through the Paste Element dialog or when <b>Auto Instance</b> is selected.</p> <p>The classifier style is the default appearance of the classifier, such as fill color and text font.</p>	
<b>Show Life Lines</b>	<p>Select the checkbox to adjust the alignment of life lines for Sequence elements.</p>	
<b>Layout uses all relations</b>	<p>Select the checkbox to automatically take connectors of all types into account when you lay out the diagram; deselect the checkbox to ignore Association connectors in automatically laying out the elements.</p> <p>The elements with Association connectors are moved as if they do not have those relationships, although the connectors themselves are still depicted on the diagram.</p> <p>This option acts through the <b>Diagram   Layout Diagram</b> command, and through the <b>Digraph</b> option of the Layout Tools window (<b>Diagram   Diagram Layout</b>).</p>	<a href="#">Layout Diagrams</a> 874
<b>Auto Save Changes</b>	<p>Select the checkbox to automatically save your changes as you work. No prompts are displayed asking you to save the changes.</p> <p>Deselect the checkbox to display prompts to save your changes when you attempt to close the diagram without having saved it.</p>	
<b>Show 'Hidden' Parents</b>	<p>Select the checkbox to list each element's parent elements that are not already part of the diagram, in the top right corner of the element in the diagram.</p> <p>Deselect the checkbox to not list the parent elements.</p>	<a href="#">Set Element Parent</a> 908
<b>Auto Group Elements</b>	<p>Select the checkbox to also move contained elements when you are moving their container elements. Deselect the checkbox to move the container elements independently.</p>	

Field/Button	Action	See also
	You can toggle to the <b>opposite</b> setting of this option when moving an element on a diagram, by pressing and holding ( <b>Alt</b> ).	
<b>Show Linked Items in Package</b>	<p>Select the checkbox to include external elements in the package contents list in a package element on a diagram. For this to take effect, the <b>Package Contents</b> checkbox must be selected on the Elements tab of the Diagram Properties dialog for the diagram.</p> <p>Deselect the checkbox to omit external elements from the list.</p>	<a href="#">Define Element Characteristics</a> <small>[828]</small>
<b>Show Package Item Scope</b>	<p>Select the checkbox to display the scope indicators (such as <b>+</b>, <b>#</b> and <b>-</b>) against the elements in the contents list within a package element. For this to take effect, the <b>Package Contents</b> checkbox must be selected on the Elements tab of the Diagram Properties dialog for the diagram.</p> <p>Deselect the checkbox to hide these indicators in the list.</p>	<a href="#">Modeling Conventions</a> <small>[2080]</small> <a href="#">Define Element Characteristics</a> <small>[828]</small>
<b>Shift-Mouse Autoscroll</b>	Select the checkbox to enable you to press and hold ( <b>Shift</b> ) and press the middle mouse button while you move the mouse, to autoscroll around diagrams.	
<b>Use Automatic SubActivities</b>	Select the checkbox to automatically create a new Structured Activity element when you drag one Activity diagram from the Project Browser onto another Activity diagram, and select the <b>Hyperlink</b> option.	<a href="#">Structured Activity</a> <small>[1338]</small> <a href="#">Diagram Frame</a> <small>[1296]</small>
<b>Always Highlight Context Element</b>	<p>Select the checkbox to show a hatched border around any element when you select it, along with the tracking handles (see <b>Tracking Handle Size</b> below).</p> <p>Deselect the checkbox to show only the tracking handles when you select an element.</p>	<a href="#">Highlight Context Element</a> <small>[930]</small>
<b>Objects Snap to Grid</b>	<p>Select the checkbox to make all elements on a diagram always align with the nearest grid point.</p> <p>Deselect the checkbox to position elements at a precise point regardless of the grid.</p>	
<b>Show Grid</b>	<p>Select the checkbox to display the grid points as pale gray dots. If you specify a large grid size (below) the grid points can be hard to see.</p> <p>Deselect the checkbox to hide the grid points.</p>	

Field/Button	Action	See also
<b>Grid Size</b>	<p>If you have selected the <b>Objects Snap to Grid</b> checkbox, type in the separation of the grid points, in pixels.</p> <p>If you deselect the <b>Objects Snap to Grid</b> checkbox the value is grayed out, but it remains in the field and is applied again if you re-select the checkbox.</p>	
<b>Auto Tidy</b>	Select the check box to automatically tidy line angles for custom connectors; this 'nudges' the custom line into horizontal and vertical increments.	<a href="#">Connector Styles</a> [1117]
<b>Tidy line gap</b>	Type in the maximum distance, in pixels, that the <b>Auto Tidy</b> option (above) can adjust the end point of a non-orthogonal line to make it orthogonal, for custom connectors.	<a href="#">Connector Styles</a> [1117]
<b>Alias only</b>	<p>Select the radio button to display, on elements with aliases, the alias instead of the element name.</p> <p>This option and the next two options take action on diagrams for which the <b>Use Alias if Available</b> checkbox is selected on the Diagram Properties dialog.</p>	<a href="#">Configure Diagram Display</a> [825]
<b>Alias and Name</b>	Select the radio button to display, on elements that have aliases, both the element name and the Alias in the format <i>(Alias) name</i> .	
<b>Use alias field for role</b>	Select the checkbox to use the <b>Alias</b> property of instances of elements to show the <b>Role</b> property.	
<b>Enable Presentation Mode</b>	<p>Select the checkbox to display your diagrams in <b>Presentation Mode</b>.</p> <p>In Presentation Mode, you can display a diagram in a presentation and, when you click on an element on the diagram, the element is highlighted with a solid colored border. This provides a more visually appealing highlight of the element you are discussing.</p> <p>In Presentation Mode, the 'tracking handles' (below) are hidden.</p>	
<b>Highlight</b>	<p>Click on the drop-down arrow and select the color in which to highlight elements in Presentation mode.</p> <p>You can select either:</p> <ul style="list-style-type: none"> <li>• One of the available colors</li> <li>• The <b>Other</b> button, or</li> <li>• (If you have previously changed the color) the <b>Default</b> color button (Aqua)</li> </ul>	<a href="#">Standard Colors</a> [616]

Field/Button	Action	See also
	<p>The <b>Other</b> button displays the Colors dialog, from which you can select a preset color from the Standard tab or define a custom color on the Custom tab.</p> <p>Click on the <b>OK</b> button to apply the new color as the highlight in diagram presentations.</p>	
<b>Tracking Handle Size</b>	<p>Click on the drop-down arrow and select the size of the 'size change handles' or 'tracking handles' on an element, connector or label in a diagram. These display when you click on the object to select it.</p> <p>You can select:</p> <ul style="list-style-type: none"> <li>• <b>Small</b> (the default, 8 pixels)</li> <li>• <b>Medium</b> (12 pixels), or</li> <li>• <b>Large</b> (16 pixels)</li> </ul> <p>If <b>Enable Presentation Mode</b> (above) is selected, the 'tracking handles' are hidden. You can still expand or contract the element by dragging on the corners, ends or mid-points of the object.</p>	
<b>Color</b>	<p>Click on the drop-down arrow and select the fill color of the 'tracking handles' (above) from the list:</p> <ul style="list-style-type: none"> <li>• <b>White</b></li> <li>• <b>Light Gray</b></li> <li>• <b>Dark Gray</b> or</li> <li>• <b>Black</b></li> </ul>	
<b>Close</b>	Click on this button to save your changes and close the Options dialog.	

#### Notes



- If you have a diagram open when you reset any of these diagram behavior options, you might need to close the diagram and open it again to see the effect

### 3.10.2.6 Sequence Diagram Options

When you are modeling using Sequence diagrams, there are a number of rendering features that are specific to that diagram type, such as the focus of the control indicator. You can define default settings for these features, in addition to the more-general diagram settings that you define on the other diagram pages of the Options dialog.

Access   **Tools | Options (Ctrl+F9) | Diagram | Sequence**

**Control display of Sequence diagrams**

Field/Button	Action	See also
<b>Message Spacing</b>	Type in the vertical gap (in points, between 15 and 45) to maintain between Sequence Messages.  This spacing can be overridden manually by dragging a Message up or down.	
<b>Default page layout is Landscape</b>	Select the checkbox to set the default orientation of your Sequence diagrams to landscape.	
<b>Show Sequence Numbering</b>	Select the checkbox to display sequence numbers against the Sequence Messages on the diagrams. (This option has immediate effect.)	
<b>Assume right to left messages are Return</b>	Select the checkbox to automatically render <b>new</b> Messages passing from right to left as Return Messages.	
<b>Default concurrency is Asynchronous</b>	Select the checkbox to set the default concurrency for Sequence Messages to <b>Asynchronous</b> .  Deselect the checkbox to set the default concurrency to <b>Synchronous</b> .	<a href="#">Message (Sequence Diagram)</a> 
<b>Enable tooltips when re-ordering messages</b>	Select the checkbox to display a tooltip when you attempt to move one Message above or below another. The tooltip is to remind you that to swap the Message positions, and thus re-order the Messages, you must hold down the <b>Alt</b> key.  If you do not hold down the <b>Alt</b> key the selected Message will move close to the next Message but will not go past it.  To avoid displaying the tooltip, deselect the checkbox. This is only advisable if you are familiar with the requirement to use the <b>Alt</b> key.	<a href="#">Layout of Sequence Diagrams</a> 
<b>Width</b>	Click on the drop-down arrow and select the line width (in points, between <b>0</b> and <b>20</b> ) of the 'focus of control' rectangle (the body at the start and end of a Message, on the Lifeline).  (This option takes effect when a new element is added, an existing element is repositioned, or the diagram is closed and re-opened.)	
<b>Assume message returns</b>	Select the checkbox to set implicit Return Messages when none are explicitly drawn (recommended).  Deselect the checkbox to use Return Messages only when they are explicitly created.	

Field/Button	Action	See also
<b>GarbageCollect</b>	Select the checkbox to automatically truncate the Lifelines for elements that have a defined lifecycle, by drawing an <b>X</b> after the last Message (that is, assume garbage collect rather than explicit delete).	<a href="#">Denote Lifecycle of an Element</a> <small>[1251]</small>
<b>Name</b>	Click on the ( ... ) button to display the MS Windows Font dialog, and define how the caption bar text (above the Sequence diagram) is rendered. The caption bar shows each element name immediately above its element in the diagram.  This feature is particularly useful for non-English character sets.  Changes take effect immediately after the Options dialog is closed.	
<b>Size</b>	Click on the drop-down arrow and select the font size of the caption bar text.  This overrides the font size set in the MS Windows Font dialog, above.	
<b>Use System</b>	Select this check box to override (but not overwrite) the <b>Name</b> and <b>Size</b> settings with the system default heading font definition.	
<b>Close</b>	Click on this button to save the changes you have made, and to close the Options dialog.	

### 3.10.3 Object Display Options

In modeling with elements, you can configure the standard contents of the elements you create and view and, in diagrams, how they are rendered and how the system responds to actions you take on them.


Access   **Tools | Options (Ctrl+F9) | Objects**

#### Control element appearance and behavior



Field/Button	Action	See also
<b>Version</b>	Type in the default version number or code for new elements.	
<b>Phase</b>	Type in the default phase number or code for new elements.	
<b>Highlight References</b>	Select the checkbox to highlight parameters in operations that are passed by reference rather than value.	<a href="#">Operation Parameters by Reference</a> <small>[1031]</small>

Field/Button	Action	See also
<b>Reference Char(s)</b>	Type in a character or character string to act as the reference.	
<b>Prefix/Suffix</b>	Select the appropriate radio button to display the <b>Reference Char(s)</b> value as either a prefix or a suffix on the operation parameter.	
<b>Warn about spaces in class names</b>	Select the checkbox to display a warning message when a Class, operation or attribute name has embedded spaces (which can cause coding problems).  Deselect the checkbox to hide such messages.	
<b>Classes honor analysis stereotypes</b>	Select the checkbox to show Classes as their stereotype; for example, if a Class is stereotyped as a Boundary, it appears as a Boundary rather than a Class.  Deselect the checkbox to show the element as its base type; that is, as a Class.	
<b>Show stereotype icon for requirements</b>	Select the check box to show a code letter in the top right corner of Requirement (E, for external), Change (C) and Issue (I) elements.  Deselect the checkbox to hide the indicator letter.	<a href="#">Requirements</a> <sup>1763</sup> <a href="#">Changes</a> <sup>2633</sup> <a href="#">Issues (Defects)</a> <sup>2631</sup>
<b>Support for Composite Objects</b>	Select the checkbox to: <ul style="list-style-type: none"> <li>• Display a hashed border within a valid parent element when you drag a potential child element onto it in a diagram</li> <li>• Automatically embed the child element in their parent element in the model, when you drag and drop the child element onto the parent element in a diagram</li> <li>• Automatically disengage the child element from the parent (break the child-parent relationship) in the model when you drag the child element out of its parent element</li> </ul> Deselect the checkbox to not display the hashed border and to not embed the child element in the parent in the model.	<a href="#">Move Elements Within Diagrams</a> <sup>917</sup>
<b>Auto-resize marks diagram 'dirty'</b>	Select the checkbox to set auto-resizing of elements as an unsaved change to the diagram (displaying an asterisk on the diagram tab), so that you must manually save the diagram to protect the change.  Deselect the checkbox to automatically save autosizing changes.	



Field/Button	Action	See also
<b>Highlight {abstract} elements</b>	Select the check box to indicate, on a diagram, Classes set as <b>abstract</b> by displaying the tag <code>{abstract}</code> below and to the right of the Class name.	<a href="#">Element Details</a> <sup>[959]</sup>
<b>Allow elongated Use Cases</b>	Select the checkbox to allow Use Cases or Use Case extension points to elongate to accommodate long names.  If you deselect the checkbox, Use Case re-sizing is proportional and the 'make same height/width' menu options are blocked.	<a href="#">Operations on Multiple Elements</a> <sup>[951]</sup>
<b>Show status colors on diagrams</b>	Select the checkbox to display color coding in Requirement, Change and Issue elements.  Deselect the checkbox to display the status bands and element shadows in the <b>standard fill and shadow colors</b> .  You can increase the range of element types to use status color coding on, through the Status page of the General Types dialog ( <b>Settings   Project Types   General Types</b> ).	<a href="#">Color Code External Requirements</a> <sup>[1776]</sup>  <a href="#">Status Types</a> <sup>[1159]</sup>
<b>Copy inheritance links on duplicate</b>	Select the checkbox to duplicate Inheritance and Realization connectors when you copy their elements.	
<b>Port and Part type visible by default</b>	Select the checkbox to show Port and Part types on the diagram by default.	
<b>Show buttons for selected Object on diagram</b>	Select the checkbox to enable the display of the floating toolbar buttons to the right of an object when you select it.    Deselect the checkbox to not display these toolbar buttons or element icons.	<a href="#">Element Icons</a> <sup>[954]</sup>
<b>Edit Object on New</b>	Select the checkbox to automatically display the element Properties dialog when a new element is added to a diagram.  Deselect the checkbox to simply create the element without displaying the Properties dialog.	
<b>Show «column» stereotype</b>	Select the checkbox to show the «column» stereotype as a heading in Table elements, when data modeling.  Deselect the checkbox to just display the column names without the «column» heading.	<a href="#">Example Data Model Diagram</a> <sup>[2337]</sup>
<b>Extend Complexity</b>	Select the checkbox to extend levels of complexity to five levels in the <b>Complexity</b> field in the element Properties dialog:	<a href="#">General Settings</a> <sup>[958]</sup> (of element

Field/Button	Action	See also
	<ul style="list-style-type: none"> <li>• <b>Easy</b></li> <li>• <b>Medium</b></li> <li>• <b>Difficult</b></li> <li>• <b>Extreme</b></li> <li>• <b>Unknown</b></li> </ul> <p>Deselect the checkbox to restrict the field to three options (omitting <b>Extreme</b> and <b>Unknown</b>).</p>	properties)
<b>UML 1.5 Components</b>	Select the checkbox to perform your modeling using UML 1.5 components (Enterprise Architect versions from 4.0 onwards default to using UML 2.x).	
<b>Show State Compartment</b>	Select the checkbox to show, in a State element, the State compartment divider under the State name.	<a href="#">Example State Machine</a> <sup>[1205]</sup>
<b>Show Duplicate Tags</b>	<p>Select the checkbox to enable duplicate tag names to be shown in the element Tagged Values compartment and in the Tagged Values window for the element.</p> <p>The system defaults to hiding duplicate tag names, to avoid displaying inherited and overridden Tagged Values. It is useful to display the duplicate tags when you are revising them; for example, when you are adding namespaces and labels in ODM.</p>	<a href="#">ODM Commands</a> <sup>[1980]</sup>
<b>Group Operations by Stereotype</b>	<p>Select the checkbox to group an element's operations by their stereotype, in the element's operations compartment on the diagram.</p> <p>The order in which the stereotypes are listed is determined by the order of the operations in the Operations dialog for the element - if operation AA with stereotype Z is the first in the operation list, stereotype ZZ is first in the stereotype list.</p> <p>Deselect the checkbox to list the operations ungrouped, in the order in which they appear in the Operations dialog for the element..</p>	<a href="#">General Properties of Operations</a> <sup>[1015]</sup>
<b>Group Attributes by Stereotype</b>	<p>Select the checkbox to group an element's attributes by their stereotype, in the element's attributes compartment on the diagram.</p> <p>The order in which the stereotypes are listed is determined by the order of the attributes in the Attributes dialog for the element - if attribute AA with stereotype Z is the first in the attribute list, stereotype ZZ is first in the stereotype list.</p> <p>Deselect the checkbox to list the attributes ungrouped, in the order in which they appear in the Attributes dialog for the element.</p>	<a href="#">General Properties of Attributes</a> <sup>[1001]</sup>



Field/Button	Action	See also
<b>Sort Features Alphabetically</b>	<p>Select the checkbox to list element features of each type alphabetically. Features include Attributes, Operations, Tags, Constraints and Test Cases.</p> <p>Deselect the checkbox to list the features in the order in which they are defined (if not alphabetical) in the corresponding dialog or window for the feature.</p> <p>To observe the effects of selecting or deselecting the checkbox, you must close the diagram and reload it.</p>	
<b>Disable spelling</b>	<p>Select the checkbox to turn off automatic spell checking.</p> <p>Deselect the checkbox to resume automatic spell checking.</p> <p>The spelling checker operates on the view, dialog or window where you initially enter the text strings.</p>	<a href="#">Spell Checking</a> 
<b>Advanced</b>	<p>Click on this button to specify which types of element should be listed in document reports and in diagram packages.</p> <p>Some types of element do not appear in packages and in document output by default. You use this facility to include such types.</p>	<a href="#">Customize Visibility of Elements</a> 
<b>Close</b>	Click on this button to save the changes you have made, and to close the Options dialog.	










### 3.10.4 Connector Display Options

It is possible to configure a number of settings that control the creation, behavior and notation for the connectors that you define in your diagrams.

Access   **Tools | Options (Ctrl+F9) | Links**

#### Control connector appearance and behavior

Field/Button	Action	See also
<b>Edit Connector on New</b>	<p>Select the checkbox to automatically open the connector Properties dialog each time you create a new connector on a diagram.</p> <p>Deselect the checkbox to simply create connectors without displaying the Properties dialog to assign a name or properties.</p>	<a href="#">Connect Elements</a> 
<b>Association default = source</b>	Select the checkbox to automatically set the direction of <b>new</b> Associations, flowing from source to target element (that is, with	<a href="#">Association</a> 

Field/Button	Action	See also
<b>--&gt; target</b>	<p>an arrow head at the target end).</p> <p>Deselect the checkbox to create Associations with <b>no</b> direction (no arrowhead on the connector line.)</p>	
<b>Generalization link style Default = Tree</b>	<p>Select the checkbox to show Generalizations in vertical tree-style, forming a hierarchy where multiple links to elements exist.</p> <p>Deselect the checkbox to create Generalizations in Auto Route style.</p>	<a href="#">Generalization</a> 
<b>Shade Qualifier boxes</b>	<p>Select the checkbox to lightly shade all Qualifier boxes on Associations in the diagrams you open (see <b>Suppress Qualifier Boxes</b>, below.)</p> <p>Deselect the checkbox to prevent shading of the Qualifier boxes.</p>	
<b>Draw Aggregations Reversed</b>	<p>Select the checkbox to draw Aggregate and Composite connectors from target element to source element (the diamond on the source end of the line).</p> <p>Deselect the checkbox (the default), to draw these connectors from source to target (the diamond on the target end of the line).</p>	<a href="#">Aggregation</a>  <a href="#">Composition</a> 
<b>Prompt on connector deletes</b>	<p>Select the checkbox to display a prompt before deleting connectors, offering the choice of hiding the connector on the diagram or deleting it completely.</p> <p>If you deselect this option, the delete operation defaults to the last-set option on the dialog.</p>	<a href="#">Delete Connectors</a> 
<b>Suppress Link Constraints</b>	<p>Select the checkbox to hide any constraints set on connectors in the diagrams you open.</p> <p>Deselect the checkbox to allow the constraints to display on the connector lines.</p>	<a href="#">Connector Constraints</a> 
<b>Suppress Qualifier boxes</b>	<p>Select the checkbox to hide the qualifiers (in their boxes) on Association connectors.</p> <p>Deselect the checkbox to show the qualifiers on the diagrams you open.</p>	<a href="#">Qualifiers</a> 
<b>Show Uses arrowheads</b>	<p>Select the checkbox to show an arrowhead on the target end of each Use connector in a Use case diagram.</p> <p>Deselect the checkbox to show the connector as a simple line.</p>	<a href="#">Use</a>  <a href="#">Show Uses Arrow Head</a> 
<b>Show Override Operation dialog</b>	<p>Select the checkbox to show the Override Operation dialog automatically when adding Generalization and Realization</p>	<a href="#">Override Parent Operations</a> 

Field/Button	Action	See also
<b>on new connector</b>	connectors between Classes and Interfaces, if the target element has features that can be overridden.	
<b>Suppress ' + ' Role Scope</b>	<p>Select the checkbox to hide the 'Public' indicator (+, when <b>Access</b> = <b>Public</b> in the connector role properties) of the source and/or target role on a connector.</p> <p>Deselect the connector to allow the + symbol to display.</p> <p>When changing this setting, you might need to reload the diagram to see the change take effect.</p>	<a href="#">Source Role</a> <sup>[1130]</sup>
<b>Strict Connector Syntax</b>	Select this checkbox to check that each new connector complies with UML, BPMN 2.0 and SysML 1.2 & 1.3 syntax as you add it, and to enforce compliance.	
<b>Pen Width</b>	Click on the drop-down arrow and select the default connector line width (between <b>1</b> and <b>4</b> pixels).	
<b>Routing</b>	Click on the drop-down arrow and select the default connector style for new connectors ( <b>Direct</b> , <b>Routed</b> or <b>Custom</b> ).	<a href="#">Connector Styles</a> <sup>[1114]</sup>
<b>Enable</b>	<p>Select the checkbox to enable the Quick Linker, showing the Quick Linker arrow when you click on an element on the diagram.</p> <p>Deselect the checkbox to hide the Quick Linker arrow and make the Quick Linker unavailable.</p>	<a href="#">The Quick Linker</a> <sup>[896]</sup>
<b>Show Help</b>	Select the checkbox to add a <b>Help</b> option at the end of the Quick Linker menu, to display the Quick Linker Help topic.	
<b>Center to center</b>	<p>Select this radio button to set the orientation of the dashed <b>guide line</b> for new connectors to anchor on the center of the source element and, as you approach the target element, the center of that element.</p> <p>The guide line displays both when you are dragging the Quick Linker arrow and when you have dragged a connector icon from the Diagram Toolbox.</p>	
<b>Exact placement</b>	<p>Select this radio button to set the orientation of the dashed <b>guide line</b> for new connectors to the exact point on the element you drag from and the exact position of the cursor as you drag to the target element.</p> <p>This option enables the <b>Force perpendicular line</b> checkbox.</p>	
<b>Force</b>	Select this checkbox to set the orientation of the dashed <b>guide line</b>	

Field/Button	Action	See also
<b>perpendicular line</b>	for new connectors to a perpendicular line between the source and target element edges, wherever this is possible.  You can toggle between this effect and the <b>Exact Placement</b> effect by holding <b>[Ctrl]</b> as you move the cursor between the source and target elements.	
<b>Close</b>	Click on this button to save the changes you have made, and to close the Options dialog.	

#### Notes

- For the **Draw Aggregations Reversed** option, all tools have the parent as the target and the child as the source of the connector, which is a requirement of UML; only the direction in which you drag the mouse to draw the connector is changed

### 3.10.5 Communication Message Colors

Communication diagrams illustrate how components interact and exchange information in a process. They have a large number of messages that indicate the timing or sequence of events in the communication process. You can define the sequence of these messages using sequence **numbers**, which also help you to **group** messages within a stage in the sequence. To help make the sequence easier to see at a glance, you can also:

- Select to display the messages in color
- Set a color for each group of messages

**Access** **Tools | Options (Ctrl+F9) | Communication Colors**

#### Notes

- Select the **Use Communication Message Coloring** checkbox to turn on message coloring in the Communication diagrams you open; when you select this option, Communication messages display in different colors depending on the sequence group they belong to on the diagram - for example, messages 1.n are black, 2.n are red and 3.n are green
- Click on the down arrow in the color field for each numerical message group and click on the color to set for that group; if you have more than eight groups in the sequence the color pattern repeats, so group 9 messages are the same color as group 1 messages, and group 10 messages are the same color as group 2 messages
- Click on the **Close** button to save the changes you have made, and to close the Options dialog

#### Learn more

- [Message \(Communication Diagram\)](#)<sup>[1428]</sup>
- [Communication Diagrams in Color](#)<sup>[1261]</sup>

### 3.10.6 XML Specifications

Many of the model development and management tasks you perform operate either directly or indirectly with XML. You can set defaults on a number of settings to standardize the way in which you use XML in your own modeling activities. You can override several of these settings on any specific XML import or export, especially when publishing a model package in XML to another modeling language or tool.

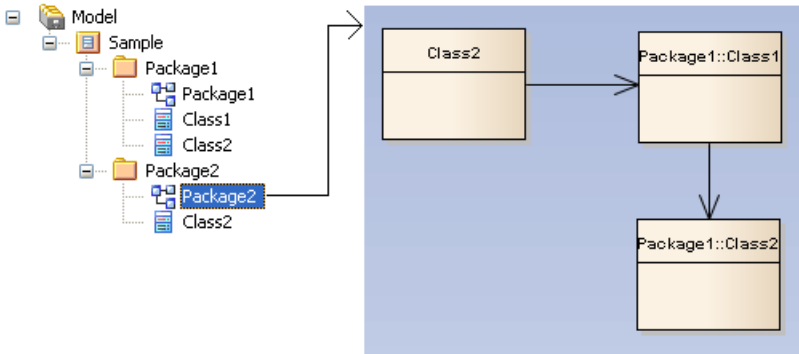
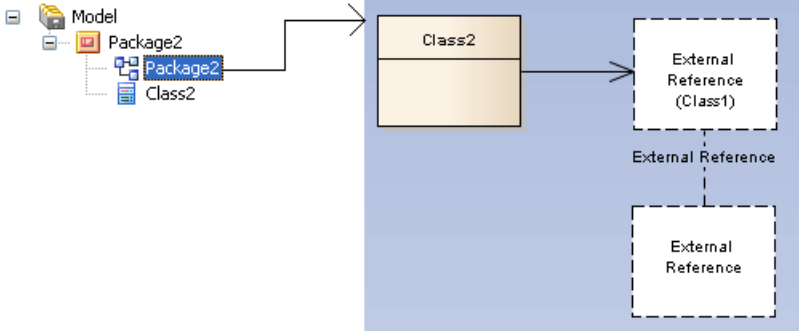
**Access**   **Tools | Options (Ctrl+F9) | XML Specifications**

#### Set XML Specification Options

Field/Button	Action	See also
<b>Editor</b>	Specify the default editor for reviewing any XML documents you open within the system. Either: <ul style="list-style-type: none"> <li>Type in the location of the editor <b>.exe</b> file, or</li> <li>Click on the ( ... ) button and browse for the location</li> </ul>	
<b>Format XML Output</b>	Select the checkbox to set <b>formatting</b> of your XML output into human-readable form as the default.  Deselect the checkbox to show the XML as unformatted code.	<a href="#">Publish Model Package</a> <sup>[476]</sup>
<b>Write Log</b>	Select the checkbox to write the execution activity to a <b>log file</b> when you import or export XML.  Deselect the checkbox to not record the activity of the import or export to a log file.	
<b>Use DTD</b>	Select the checkbox to default to using a <b>Data Type Definition (DTD)</b> in your XML imports and exports. A DTD validates the correctness of the model and checks that no syntactical errors have occurred.  Deselect the checkbox to set the default to not using a DTD.	<a href="#">The UML DTD</a> <sup>[483]</sup>
<b>Export Diagrams</b>	Select the checkbox to default to including diagrams when you export XML to another format or tool.	<a href="#">Publish Model Package</a> <sup>[476]</sup>
<b>Export Alternate Images</b>	Select the checkbox to export any alternative images used in the model when you export to XML.	<a href="#">Using the Image Manager</a> <sup>[860]</sup>  <a href="#">Sharing Reference Data</a> <sup>[374]</sup>
<b>Code Page</b>	Defaults to <b>windows-1252</b> ; if you do not want to use this Code Page, type in your preferred code page reference.  Setting a <b>null</b> encoding string results in the encoding tag being entirely	

Field/Button	Action	See also
	<p>omitted from the XML output.</p> <p>Click on the <b>Default</b> button to restore the setting to the default Code Page.</p>	
<b>Import using single transaction</b>	<p>Select the checkbox to make all XML <b>import</b> operations and all <b>Paste Package from Clipboard</b> operations default to being performed within a single transaction.</p> <p>Deselect this checkbox to make all XML imports and paste package operations default to being performed <b>data item by data item</b>, so that one failure does not block the whole operation (recommended for large imports/copies).</p> <p>In either case, in any specific XML import operation the setting can be overridden.</p>	<a href="#">Import from XML</a> <sup>[478]</sup> <a href="#">Copy a Package</a> <sup>[775]</sup>
<b>Validate XML against Schema/DTD</b>	Select the checkbox to validate the XML format against the schema or Data Type Definition when loading an imported XML file.	
<b>Export Diagram Images</b>	Select the checkbox to export diagrams as <b>images</b> when you export to XML.	
<b>Image Type</b>	If you select the <b>Export Diagram Images</b> option (above), click on the drop-down arrow and select the image format to export to. Select from <b>metafile, bitmap, .gif, .png</b> and <b>.jpeg</b> .	
<b>Create placeholders for missing External References during XML 1.1/2.1 Import</b>	<p>Select the checkbox to show, during the import of an Enterprise Architect-exported XML 1.1/2.1 file, any missing:</p> <ul style="list-style-type: none"> <li>Diagram elements, using a Boundary element as a placeholder</li> <li>Diagram connectors, using a Dependency connector as a placeholder</li> </ul> <p>When importing an Enterprise Architect-exported XML 1.1/2.1 file with cross-package references, use this option to visually show the diagram items that are not yet in the model.</p> <p>A sample scenario would be, as shown below, exporting <i>Package2</i> to XML 1.1 from <i>Master.EAP</i> and importing it into <i>New.EAP</i> - the Classes <i>Class1</i> and <i>Class2</i> and the connector between them are shown using a placeholder on the diagram <i>Package2</i>.</p>	



Field/Button	Action	See also
	<p>Master.EAP :</p>  <p>New.EAP :</p>  <p>If the file is for a package under package control or version control, and this checkbox is <b>not</b> selected, it is overridden if the <b>For all packages, create placeholders for external references</b> checkbox is selected on the Version Control Settings dialog and/or Package Control Options dialog.</p>	<p><a href="#">Version Control Settings</a> <sup>[415]</sup></p> <p><a href="#">Configure Packages</a> <sup>[488]</sup></p>
<b>Prefix EA tagged values in XMI 1.0 with ea\$</b>	Select the checkbox to prefix any Enterprise Architect Tagged Values within any <b>XMI 1.0</b> you create, with <b>ea\$</b> .	
<b>Update Last Save Time</b>	Select the checkbox to update the timestamp of the last time each controlled package was saved.	
<b>Report Cross Package Reference Deletions (XMI 1.1)</b>	Select the checkbox to report any Package cross-references that would be deleted when a Controlled Package from a completed model is exported to XMI 1.1.	<a href="#">Report Deletion of Cross-Package References</a> <sup>[496]</sup>
<b>Default XML Directory</b>	Type in the location of the default XML directory to use when importing and exporting XML, or click on the [ ... ] (Browse) button and browse for the location.	

Field/Button	Action	See also
<b>Close</b>	Click on this button to save the changes you have made, and to close the Options dialog.	

#### Notes

- If **Export Alternate Images** is set, and you have packages in your model under version control, then any alternative images used in those packages are also exported to the version control repository when you check in the packages; in this case, you would only select the checkbox if the alternative images are subject to frequent change

Otherwise, do not select this option and instead use the **Export Reference Data** and **Import Reference Data** options to manage alternative images

#### Learn more

- [Export MOF Model to XML](#)<sup>[2460]</sup>
- [Export Reference Data](#)<sup>[376]</sup>
- [Import Reference Data](#)<sup>[380]</sup>

**Part**

---

**IV**

## 4 Navigate, Search and Trace



This section explains how you **navigate** through the model structures in Enterprise Architect, locate and display specific information, and trace the origins, development and fate of the model elements. While the **Project Browser** provides an effective hierarchical view of the model, large models can present particular challenges, and tracing **dependencies** and **relationships** is not always a simple task. Fortunately, Enterprise Architect has a number of tools designed to meet just these requirements.

### Topics

Topic	Link
The use of the Project Browser and Package Browser to navigate and explore your model	<a href="#">The Project Browser</a> <sup>[646]</sup> <a href="#">Package Browser</a> <sup>[673]</sup>
How Model Views provide different perspectives and 'entry points' into a model	<a href="#">Model Views</a> <sup>[686]</sup> <a href="#">Diagram Slide Show</a> <sup>[695]</sup>
Finding elements and relationships using the Model Search tools and custom queries	<a href="#">Model Search</a> <sup>[700]</sup> <a href="#">Pre-defined Searches</a> <sup>[706]</sup>
Traceability tools to help track completeness, dependencies and other behavioral and structural relationships	<a href="#">Traceability Tools</a> <sup>[724]</sup>
Use of the Diagram List or Package Browser to view model elements in a conventional list based manner	<a href="#">Diagram List</a> <sup>[684]</sup> <a href="#">Package Browser</a> <sup>[673]</sup>
Use of the Relationship Matrix to understand and track relationships between model elements in a spreadsheet view	<a href="#">Relationship Matrix</a> <sup>[727]</sup>
Using Diagram Filters to highlight information in diagrams	<a href="#">Diagram Filters</a> <sup>[718]</sup>
Using the Traceability window to explore model dependencies	<a href="#">The Traceability</a>

Topic	Link
	<a href="#">Window</a> <sup>[725]</sup>
Finding and working with Element Relationships	<a href="#">The Relationships Window</a> <sup>[742]</sup>

## 4.1 Navigate: Exploring Your Model



Navigating is the process of systematically exploring the structure of your model using the hierarchical and list based views, such as the Project Browser. It is particularly useful when you are familiar with the structure of a model and its packaged contents. Whereas Searching provides a flat and random set of results, the Project and Package Browsers provide a highly structured viewpoint, reflecting the exact structure of your model.

### Topics

Topic	Link
Use the Project Browser, the main interface element for exploring and navigating your model	<a href="#">The Project Browser</a> <sup>[646]</sup> <a href="#">Project Browser Context Menus</a> <sup>[648]</sup> <a href="#">Project Browser Toolbar</a> <sup>[669]</sup>
The Package Browser - a tabular, editable view of elements in a selected package, which can be displayed in the main workspace	<a href="#">Package Browser</a> <sup>[673]</sup>
The Diagram List - a tabular, editable view of elements in a selected diagram, which can be displayed in the main workspace	<a href="#">Diagram List</a> <sup>[684]</sup>
Use Model Views to provide different perspectives and 'entry points' into your model	<a href="#">Model Views</a> <sup>[686]</sup>
Set up a Diagram Slide show	<a href="#">Diagram Slide Show</a> <sup>[695]</sup>
Use the Pan and Zoom window to navigate around very large diagrams	<a href="#">The Pan &amp; Zoom Window</a> <sup>[698]</sup>

### 4.1.1 The Project Browser

Using the Project Browser, you can navigate through the Enterprise Architect project space. It displays Packages, diagrams, elements and element features in a tree-like structure, reflecting the arrangement of elements and Packages within your model. The Project Browser is the primary mechanism for browsing and exploring your model and is the jumping off point for many of the most important features in Enterprise Architect. You can use it to:

- Review the structure, content and organization of your model
- Drill down to specific elements

- Drag and drop elements and Packages within the model
- Copy (duplicate) whole Packages
- Import/export model information
- Set up version control, locking, execution analysis and many other aspects of the model tied to a particular Package
- Document models
- Import code, xmi and csv files, database schema and other external data sources

If you right-click on an item in the Project Browser to display the context menus, you can perform additional actions such as adding new Packages, creating diagrams, renaming items, creating documentation and other reports, and deleting model elements. You can also edit the name of any item in the Project Browser by selecting the item and pressing **F2**.

**Access** **View | Project Browser ( Alt+0 )**

### Views

The Project Browser can be divided into Views, each of which contains diagrams, Packages and other elements. A default View hierarchy is described here, but you can create different Views to suit your requirements:

View	Description
Use Case View	The functional and early analysis View, containing Business Process and Use Case models.
Dynamic View	The dynamics of your system, containing State Charts, Activity and Interaction diagrams.
Logical View	The Class Model and Domain Model View.
Component View	A View for your system components. The high level view of what software is to be built (such as executables, DLLs and components).
Deployment View	The physical model; what hardware is to be deployed and what software is to run on it.
Custom View	A work area for other Views, such as formal requirements, recycle bin, interview notes and non-functional requirements.

### Selective Collapse to the parent element or Package

When you are working on an expanded project in the Project Browser, you might want to locate the parent element or Package of an item, and/or collapse the structure under that parent element or Package.

Step	Action	See also
1	Position the cursor on an item within the element or Package.	
2	Press ( ← ) on the keyboard to highlight the parent.	
3	Press the key again to collapse the structure under that parent element or Package.	

#### Learn more

- [Views](#) <sup>[769]</sup>
- [Project Browser Icon Overlays](#) <sup>[670]</sup>

### 4.1.1.1 Project Browser Context Menus

The Project Browser provides a wide range of options to work on the objects it contains. These options are presented through a number of **context-sensitive menus**.

#### Learn more

- [Model \(Root Node\) context menu](#) <sup>[648]</sup>
- [Package Context Menu](#) <sup>[651]</sup>
- [Element Options in the Project Browser](#) <sup>[660]</sup>
- [Diagram Context Menu](#) <sup>[665]</sup>
- [Operation Context Menu](#) <sup>[667]</sup>

#### 4.1.1.1.1 Model (Root Node) Context Menu

The **Root Node** in the Project Browser is the **Model** element. You can have more than one model element. The first level Packages beneath the Model node are sometimes referred to as **Views** as they commonly divide a model into categories such as Use Case Model and Logical Model.

**Access**    **Right-click on the Root Node**

#### Options



Option	Action	Shortcut	See also
<b>Extensions</b>	<p>Select this option to access a submenu of enabled Technologies on the system (for example, ArcGIS or TOGAF), each of which provides options for performing technology-specific operations on the Package.</p> <p>This option is available in the Professional, Corporate and Extended editions of Enterprise Architect.</p>		<a href="#">Geodatabase Design for ArcGIS</a> <sup>[1944]</sup>



Option	Action	Shortcut	See also
<b>Scripts</b>	<p>Select this option to list the scripts enabled for execution directly from the Project Browser.</p> <p>(The option is not available if no Project Browser scripts exist.)</p> <p>Click on a script name to run that script.</p>		<a href="#">Script Group Properties</a> <sup>[2794]</sup>
<b>Add</b>	Select this option to access the <b>Add</b> submenu; see below.		
<b>Rename Model</b>	Select this option to display a simple prompt for a new name to apply to the current model.		
<b>Package Control</b>	Select this option to access the <b>Package Control</b> submenu.		<a href="#">Package Control</a> <sup>[485]</sup>
<b>Update Package Status</b>	Select this option to change the status, phase and/or version of all elements and child Packages in the selected Package.		<a href="#">Update Package Status</a> <sup>[590]</sup>
<b>Apply/Release User Lock</b>	<p>Select this option to apply or release a user lock on the selected Package.</p> <p>(This option is only displayed when user security is enabled in <b>Require User Lock to Edit</b> locking mode.)</p>		<a href="#">Apply a User Lock</a> <sup>[339]</sup>
<b>Lock Package</b>	<p>Select this option to apply a lock on the selected Package.</p> <p>(This option is only displayed when user security is enabled in <b>User/Group</b> locking mode.)</p>		<a href="#">Lock Package</a> <sup>[337]</sup>
<b>Apply Auto Naming to Elements</b>	Select this option to apply configured auto naming conventions to existing elements in the selected Package.		<a href="#">Apply Auto Naming to Existing Elements</a> <sup>[906]</sup>
<b>Copy Package to Clipboard</b>	Select this option to copy the selected Package to the clipboard, to be copied into another Package in either the same project file or a different project file.		<a href="#">Copy the selected Package</a> <sup>[775]</sup>
<b>Paste Package from Clipboard</b>	Select this option to paste a Package from the clipboard into the selected Package.		<a href="#">Paste a Package</a> <sup>[775]</sup>

Option	Action	Shortcut	See also
<b>Find in Project Browser</b>	Select this option to find a specified term in the Project Browser.	<b>Ctrl+Shift+F</b>	
<b>Expand Branch</b>	Select this option to expand all items in this branch of the Project Browser.		
<b>Collapse Branch</b>	Select this option to collapse all items in this branch of the Project Browser.		
<b>Import Model from XMI</b>	Select this option to import a model from its XMI file to under this root node.	<b>Ctrl+Alt+I</b>	<a href="#">Import a model</a> <sup>[473]</sup>
<b>Export Model to XMI</b>	Select this option to export the model under this root node to an XMI file.	<b>Ctrl+Alt+E</b>	<a href="#">Export a model</a> <sup>[473]</sup>
<b>Generate Documentation</b>	Select this option to generate documentation on the contents of the model.	<b>F8</b>	<a href="#">Document Reports</a> <sup>[2640]</sup>
<b>HTML Report</b>	Select this option to generate web documentation on the contents of the model.	<b>Shift+F8</b>	<a href="#">Web Reports</a> <sup>[2744]</sup>
<b>Diagrams Only Report</b>	Select this option to generate a Diagrams Only report for the model.	<b>Ctrl+Shift+F8</b>	<a href="#">Diagrams Only Report</a> <sup>[2741]</sup>
<b>Copy Reference</b>	Select this option to copy a reference to the root node to the Enterprise Architect clipboard. Select the appropriate sub-option to copy the: <ul style="list-style-type: none"> <li>Selected Package hierarchy structure (Node path) or</li> <li>Node GUID</li> </ul>		
<b>Delete Project Root</b>	Select this option to delete the Model root node, <b>after</b> you have already deleted all of its subordinate Views and Packages.  A prompt displays for you to confirm the deletion.		
<b>Help</b>	Select this option to display the Help topic for the Project Browser.		

**Add Submenu**

Option	Action	Shortcut	See also
<b>Add Model (root node)</b>	Select this option to create a new model root, without any subordinate structure or content.  A prompt for the model name displays.		
<b>Add View</b>	Select this option to create a new View (Package) underneath the model root node.  The Create New View dialog displays.		<a href="#">Add Views</a> 
<b>Add a Model using Wizard</b>	Select this option to add further models using the Model Wizard.  The list of available Model Pattern Technologies includes the MDG Technology Builder, which creates a Model Package containing templates that you can use to generate a new MDG Technology containing a set of Profiles.	<b>Ctrl+Shift +M</b>	<a href="#">Model Wizard</a> 



[Learn more](#)

[Projects Defined](#) 

#### 4.1.1.1.2 Package Options in the Project Browser

**Access** **Project Browser | Right-click on Package or View**

##### Options

Option	Action	Shortcut	See also
<b>Extensions</b>	Select this option to access a submenu of enabled Technologies on the system (for example, ArcGIS or TOGAF), each of which provides options for performing technology-specific operations on the Package.  This option is available in the Professional, Corporate and Extended editions of Enterprise Architect.		<a href="#">Geodatabase Design for ArcGIS</a> 
<b>Scripts</b>	Select this option to list the scripts enabled for execution directly from the Project Browser.  (This option does not display if no Project Browser scripts exist.)  Click on a script name to run that script.		<a href="#">Script Group Properties</a> 

<b>Properties</b>	Select this option to display the Properties dialog for the Package element.		<a href="#">Properties Dialog</a> <sup>[956]</sup>
<b>Linked Document</b>	Select this option to create or display a linked document for the Package or View.	<b>Ctrl+Alt+D</b>	<a href="#">Linked documents</a> <sup>[1044]</sup>
<b>Delete Linked Document</b>	Select this option to delete the linked document attached to the Package.  The system prompts you to confirm the deletion.		
<b>Advanced</b>	Select this option to access a sub-menu of options for performing more-specialized operations on the Package.		<a href="#">Advanced Sub-Menu</a> <sup>[654]</sup>
<b>List View</b>	Select this option to display the Package contents in the <b>Package Browser</b> in <b>List View</b> mode.		<a href="#">Package Browser</a> <sup>[673]</sup>
<b>Gantt View</b>	Select this option to display the Package contents in the <b>Package Browser</b> in <b>Gantt View</b> mode		<a href="#">The Gantt View</a> <sup>[594]</sup>
<b>Specification Manager</b>	Select this option to display the Package contents in the <b>Specification Manager</b> .  You can display the Specification Manager at the same time as either the Package Browser or Gantt View.		<a href="#">Specification Manager</a> <sup>[1728]</sup>
<b>Add a Model using Wizard</b>	Select this option to add another Model Package to the model or View, using the Model Wizard.	<b>Ctrl+Shift+M</b>	<a href="#">Model Wizard</a> <sup>[753]</sup>
<b>Add a Package</b>	Select this option to add another Package to the selected Package.	<b>Ctrl+W</b>	<a href="#">Add a Package</a> <sup>[772]</sup>
<b>Add Diagram</b>	Select this option to add a new diagram to the selected Package.		<a href="#">Add New Diagrams</a> <sup>[822]</sup>
<b>Add Element</b>	Select this option to add a new element to the selected Package.	<b>Ctrl+M</b>	<a href="#">Add Elements Directly To Packages</a> <sup>[903]</sup>
<b>Package Control</b>	Select this option to submit the selected Package (s) to Package control and version control.		<a href="#">Controlled Package Menu</a> <sup>[485]</sup> <a href="#">Package Version</a>

			<a href="#">Control Menu</a> <sup>[427]</sup>
<b>Copy/Paste</b>	Select this option to access a menu of options for copying and moving the Package contents to other areas of the model.		<a href="#">Copy/Paste Submenu</a> <sup>[654]</sup>
<b>Move Up</b>	Select this option to move the Package further up the list in the Project Browser.		
<b>Move Down</b>	Select this option to move the Package further down the list in the Project Browser.		
<b>Contents</b>	Select this option to reorganize the Package contents after making changes.		<a href="#">Contents Sub-menu</a> <sup>[655]</sup>
<b>Find in All Diagrams</b>	Select this option to list all diagrams in which this Package is depicted, in the Element Usage dialog.	<b>Ctrl+U</b>	<a href="#">Show Element Use</a> <sup>[910]</sup>
<b>Documentation</b>	Select this option to produce a variety of reports and documentation on the model.		<a href="#">Documentation Sub-menu</a> <sup>[656]</sup>
<b>Code Engineering</b>	Select this option to perform Code Engineering functions.		<a href="#">Code Engineering Options</a> <sup>[657]</sup>
<b>Execution Analyzer</b>	Select this option to display the Execution Analyzer window, with the Analyzer Scripts for the selected Package expanded.  If no Analyzer Scripts have been configured you are given the opportunity to create one.		<a href="#">Execution Analyzer</a> <sup>[2527]</sup>
<b>Import/Export</b>	Select this option to import and export the Package contents using XML text files.		<a href="#">Import/Export Sub-menu</a> <sup>[659]</sup>
<b>Set View Icon</b>	Select this option to change the display icon for the selected Package (View level Packages only).		<a href="#">Views</a> <sup>[769]</sup>
<b>Delete &lt;packagename&gt;</b>	Select this option to delete the selected Package and its contents. A prompt displays to confirm the deletion.		
<b>Help</b>	Select this option to display the Help topic for the Project Browser.		

#### 4.1.1.1.2.1 Advanced Sub-Menu

In the Project Browser, right-click on a Package and select the **Advanced** context menu option.

Option	Action	Shortcut	See also
<b>Turn On Level Numbering</b> (Turn Off Level Numbering)	<p>Select this option to add a sequence number to each element in the Package, based on the element's position in the Package hierarchy.</p> <p>For nested elements, the numbering indicates level; that is:</p> <p>3.2 3.2.1 3.2.1.1</p> <p>This option is only available for Packages, and the numbering only applies to the elements in the Package, not to the diagrams.</p> <p>If elements are added, moved or deleted from the Package, the numbering automatically adjusts.</p>		<a href="#">Model Requirements</a> <sup>[1766]</sup>
<b>Update Package Status</b>	Select this option to provide a bulk update on the status, phase and version of a Package and its elements.		<a href="#">Update Package Status</a> <sup>[590]</sup>
<b>Apply Auto Naming to Elements</b>	Select this option to apply configured auto naming conventions to existing elements in the selected Package.		<a href="#">Apply Auto Naming to Existing Elements</a> <sup>[906]</sup>
<b>Transform Package</b>	Select this option to perform a model transformation on the selected Package.	<b>Ctrl+Shift+H</b>	<a href="#">Transform Elements</a> <sup>[2017]</sup>
<b>Save Package as UML Profile</b>	Select this option to save the selected Package as a UML Profile.		<a href="#">Using UML Profiles</a> <sup>[1472]</sup>

#### 4.1.1.1.2.2 Copy/Paste Submenu

In the Project Browser, right-click on a Package and select the **Copy/Paste** context menu option.

Option	Action	Shortcut	See also
<b>Cut</b>	Select this option to remove the selected Package from its parent Package, to be pasted under another Package. The selected Package remains where it is until the <b>Paste</b> operation (below) is invoked.	<b>Ctrl+X</b>	<a href="#">Move Elements Between Packages</a> <sup>[913]</sup>
<b>Paste</b>	Select this option to paste a <b>cut</b> Package under the selected Package. If the selected Package is not a valid target, the <b>Paste</b> option is grayed out.	<b>Ctrl+V</b>	

Option	Action	Shortcut	See also
	(This option is not available for <b>copied</b> Packages - use the <b>Paste Package from Clipboard</b> context menu option.)		
<b>Copy Package to Clipboard</b>	Select this option to copy the selected Package to the clipboard, to be copied into another Package in the same project file or a different project file.	<b>Ctrl+C</b>	<a href="#">Copy a Package</a> [77]
<b>Paste Package from Clipboard</b>	Select this option to paste a Package from the clipboard into the selected Package.	<b>Ctrl+V</b>	
<b>Paste Element (s) from Clipboard</b>	Select this option to paste elements copied to the clipboard into the selected Package.	<b>Ctrl+V</b>	<a href="#">Copy Elements Between Packages</a> [915]
<b>Paste Diagram</b>	If you have copied a diagram from another Package, select this option to paste the diagram into the currently-selected Package.		
<b>Copy Node Path to Clipboard</b>	Select this option to copy the Package's node path (hierarchy structure) to the Enterprise Architect clipboard, as a reference.		
<b>Copy Node GUID to Clipboard</b>	Select this option to copy the Package's node GUID to the Enterprise Architect clipboard, as a reference.		

#### 4.1.1.1.2.3 Contents Sub-Menu

In the Project Browser, right-click on a Package and select the **Contents** context menu option.

Option	Action	Shortcut	See also
<b>Find in Project Browser</b>	Select this option to search the Project Browser for specific elements (identified using the Find in Project Browser dialog).	<b>Ctrl+Shift+F</b>	
<b>Bookmarks</b>	Select this option to bookmark all elements in the selected folder.		<a href="#">Manage Bookmarks</a> [59]
<b>Expand Branch</b>	Select this option to expand all of the items in the Project Browser.		
<b>Collapse</b>	Select this option to collapse all of the items in the Project		

Option	Action	Shortcut	See also
<b>Branch</b>	Browser.		
<b>Reset Sort Order</b>	Select this option to return sorting of Package contents to list in alphabetical order.		
<b>Reload Current Package</b>	Select this option to refresh the current Package in the Project Browser.		<a href="#">Refresh Model View</a> [308]

#### 4.1.1.1.2.4 Documentation Sub-Menu

In the Project Browser, right-click on a package and select the **Documentation** context menu option.

Option	Description	Shortcut	See also
<b>Generate Documentation</b>	Select this option to design and generate reports and documentation on the contents of the selected Package.  The Document Generation dialog displays.	<b>F8</b>	<a href="#">Document Reports</a> [2640]
<b>HTML Report</b>	Select this option to design and generate web documentation on the contents of the selected Package.  The Publish as HTML dialog displays.	<b>Shift+F8</b>	<a href="#">Create a Web Page Report</a> [2744]
<b>Diagrams Only Report</b>	Select this option to generate a Diagrams Only report, which describes the diagrams in the selected Package.  The Export Diagrams to RTF Document dialog displays.	<b>Ctrl+Shift+F8</b>	<a href="#">Diagrams Only Report</a> [2741]
<b>Testing Report</b>	Select this option to generate a Testing report, which describes the software development tests defined for the selected Package.  The Generate Test Documentation dialog displays.		<a href="#">Test Documentation</a> [2617]
<b>Open in Relationship Matrix</b>	Select this option to open the Relationship Matrix with the selected Package as the source or target Package for the Matrix.		<a href="#">Relationship Matrix</a> [727]
<b>Generated Report Options</b>	Select this option to exclude the selected Package from any reports generated from now		<a href="#">Exclude Package from Report</a> [2761]



Option	Description	Shortcut	See also
	onwards, or to include the excluded Package in reports again.		
<b>Copy Documentation Bookmark</b>	<p>When using the Legacy Report Generator, if you want to create a complex document that consists of <b>sections</b> of one or more document reports, rather than one <b>complete</b> report, you can create a master document in Microsoft Word™ and within it create links to the report sections using their <b>bookmarks</b>.</p> <p>Select this menu option to copy the bookmark on the selected Package to the clipboard, to be pasted into a search or other reference to the bookmark.</p>		<a href="#">Document Bookmarks</a> <sup>[2730]</sup>
<b>Package Metrics</b>	<p>Select this option to perform some calculations on the selected Package to establish Project Metrics for planning purposes.</p> <p>The <b>QA Reports view</b> opens on the Use Case Metrics tab.</p>		<a href="#">Estimating Project Size</a> <sup>[588]</sup>

#### 4.1.1.1.2.5 Code Engineering Options

**Access** Project Browser, right-click on a Package | Code Engineering

Option	Description	Shortcut	See also
<b>Generate Source Code</b>	Select this option to generate the code for the elements in the selected Package.	<b>Ctrl+Alt+K</b>	<a href="#">Generate a Package</a> <sup>[2115]</sup>
<b>Import Source Directory</b>	Select this option to import code from all source files in a complete directory structure, which helps you to add or synchronize multiple files in a directory.	<b>Ctrl+Shift+U</b>	<a href="#">Import Source Directory</a> <sup>[2142]</sup>
<b>Import from source File(s)</b>	Select this option to import source code into the project from one or more external source files.		<a href="#">Import Source Code</a> <sup>[2138]</sup>
<b>Import Binary Module</b>	<p>Select this option to import binary into your model from external source files. Files you can currently import include:</p> <ul style="list-style-type: none"> <li>• Java Archive (.jar)</li> <li>• .Net PE file (.exe, .dll) - Native Windows DLL and EXE files are <b>not</b> supported, only PE files containing .NET assembly data</li> </ul>		<a href="#">Import Binary</a> <sup>[2143]</sup>

Option	Description	Shortcut	See also
	<ul style="list-style-type: none"> <li>Intermediate Language file (.il)</li> </ul>		
<b>Import Resource Script</b>	Select this option to import Microsoft Windows™ <b>Resource Scripts</b> (as .rc files), which contain the Win32 dialog definitions (those with the stereotype «win32Dialog») for an application's graphical user interface.		<a href="#">Import Resource Script</a> <sup>[214]</sup>
<b>Synchronize Package With Code</b>	Select this option to synchronize the elements in the selected Package in the Project Browser with the source code.	<b>Ctrl+Alt+M</b>	<a href="#">Update Package Contents</a> <sup>[211]</sup> <a href="#">Element Options in the Project Browser</a> <sup>[66]</sup> <a href="#">Operation Menu - Project Browser</a> <sup>[66]</sup>
<b>Generate DDL</b>	Select this option to generate DDL from the selected Package, for Tables, Stored Procedures and Views.		<a href="#">Generate DDL</a> <sup>[238]</sup>
<b>Import DB schema from ODBC</b>	Select this option to import database schema into a standard UML model, to initially create a database definition and subsequently update the model from the live database.		<a href="#">Import DB schema from ODBC</a> <sup>[237]</sup>
<b>Generate XML Schema</b>	Select this option to forward-engineer an XML Schema model into W3C XML Schema (XSD) files.		<a href="#">Generate XSD</a> <sup>[241]</sup>
<b>Import XML Schema</b>	Select this option to reverse engineer a W3C XML Schema (XSD) file to create or overwrite a Package of your UML Class model.		<a href="#">Import XSD</a> <sup>[242]</sup>
<b>Generate WSDL</b>	Select this option to forward-engineer a WSDL model into WSDL 1.1 files.		<a href="#">Generate WSDL</a> <sup>[244]</sup>
<b>Import WSDL</b>	Select this option to reverse engineer a WSDL 1.1 file to create or overwrite a Package of your UML Class model.		<a href="#">Import WSDL</a> <sup>[244]</sup>
<b>Reset Options for this Package</b>	<p>Select this option to manually reset the stored options for all Classes in a Package.</p> <p>Some Class options are stored when a Class is created, and so for existing Classes are not updated by changes to</p>		<a href="#">Reset Options</a> <sup>[227]</sup>

Option	Description	Shortcut	See also
	the Options dialog. These include the code language, whether child Packages are processed, and whether filenames are cleared, and are the options you update using this method.		
<b>Reset DBMS Options</b>	Select this option to map the DBMS data types of the Package to the data types of another DBMS.		<a href="#">Reset DBMS Options</a> <sup>[2349]</sup>
<b>Set as Namespace Root</b> <b>Clear Namespace Root</b>	Select these options to set the selected Package as the Namespace Root for its hierarchy, or to clear the Package as a namespace root.		<a href="#">Namespaces</a> <sup>[2120]</sup>
<b>Suppress Namespace</b> <b>Show Namespace</b>	Select these options to temporarily omit the selected Package from a namespace definition, and to restore it again.		
<b>Live Code Generation</b>	Select this option to generate code from the elements in the Package as they are added or changed.		

#### 4.1.1.1.2.6 Import/Export Sub-Menu

In the Project Browser, right-click on a Package and select the **Import/Export** context menu option.









Option	Description	Shortcut	See also
<b>Import package from XMI file</b>	Select this option to <b>import</b> a Package from an <b>XMI</b> (XML based) file into the currently-selected Package.	<b>Ctrl+Alt +I</b>	<a href="#">Import Package</a> <sup>[478]</sup>
<b>Export package to XMI file</b>	Select this option to <b>export</b> the currently selected Package to an <b>XMI</b> (XML based) file.	<b>Ctrl+Alt +E</b>	<a href="#">Export Package</a> <sup>[475]</sup>
<b>CSV Import / Export</b>	Select this option to <b>import</b> or <b>export</b> information on the model elements in the selected Package, in <b>CSV</b> format.	<b>Ctrl+Alt +C</b>	CSV <a href="#">Import</a> <sup>[503]</sup> / <a href="#">Export</a> <sup>[501]</sup>

#### 4.1.1.1.3 Element Options in the Project Browser

The element context menu in the Project Browser is available for all types of element. It provides options for performing a wide range of operations on both individual elements and multiple elements, including adding element properties and features, and moving the element around the project or between projects.

##### **Access**    **Project Browser | Right click on element**

Option	Action	Shortcut	See also
<b>Extensions</b>	Select this option to access a list of enabled Technologies on the system (for example, ArcGIS or TOGAF), each of which provides options for performing technology-specific operations on the element.  This option is available in the Professional, Corporate and Extended editions of Enterprise Architect.		<a href="#">Geodatabase Design for ArcGIS</a> <sup>[1944]</sup>
<b>Scripts</b>	Select this option to access a list of the scripts enabled for execution directly from the Project Browser.  (This option does not display if no Project Browser scripts exist for operating on elements.)		<a href="#">Scripts</a> <sup>[2794]</sup>
<b>Properties</b>	Select this option to display the element Properties dialog, to view and modify the element properties.		<a href="#">Properties Dialog</a> <sup>[956]</sup>
<b>Linked Document</b>	Select this option to create or edit a Linked Document (Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions).	<b>Ctrl+Alt+D</b>	<a href="#">Linked Documents</a> <sup>[1044]</sup>
<b>Delete Linked Document</b>	Select this option to delete the Linked Document attached to the selected element.		
<b>Generate Documentation</b>	Select this option to generate a range of reports and other documents on the element.	<b>F8</b>	<a href="#">Document Reports</a> <sup>[2640]</sup>
<b>Lock</b>	Select this option to apply or release a lock for editing the selected element. (The option is displayed only when user security is enabled in <b>User/Group</b> locking mode.)		<a href="#">Lock Model Elements</a> <sup>[336]</sup>
<b>Apply/Release User Lock</b>	Select this option to apply or release a user lock on the selected element. (The option is displayed only when user security is enabled in <b>Require User Lock to Edit</b> locking mode.)		<a href="#">Apply a User Lock</a> <sup>[339]</sup>

<b>Add</b>	Select this option to access a submenu of options for creating a child element and diagram (Classifier elements) or a connector to another element.		<a href="#">Add Sub-Menu</a>  <small>662</small>
<b>Attributes</b>	Select this option to display the Attribute dialog ready to create a new attribute.	<b>F9</b>	<a href="#">General Properties of Attributes</a>  <small>1007</small>
<b>Operations</b>	Select this option to display the Operations dialog ready to create a new operation.	<b>F10</b>	<a href="#">General Properties of Operations</a>  <small>1015</small>
<b>Override Attribute Initializers</b>	(For a Class element with attributes.) Select this option to pre-define initial values for attributes that can be used to override existing defaults.	<b>Ctrl+Shift+R</b>	<a href="#">Display Inherited Attributes</a>  <small>1007</small>
<b>Copy/Paste</b>	Select this option to access a menu of options for copying and moving the Package contents to other areas of the model.		<a href="#">Element Copy/Paste Submenu</a>  <small>663</small>
<b>Move Up</b>	Select this option to move the element up in the list of elements within this Package.		
<b>Move Down</b>	Select this option to move the element up in the list of elements within this Package.		
<b>Find in all Diagrams</b>	Select this option to locate the element in all open diagrams.  This option also operates on Port and Part Property Type Classifiers.	<b>Ctrl+U</b>	<a href="#">Show Element Use</a>  <small>910</small>
<b>Locate in Current Diagram</b>	Select this option to select the element in the currently-visible diagram.  If the element is not in the diagram, this option is grayed-out.		
<b>View Audit History</b>	Select this option to open the Audit View to see a history of changes made to the selected element.		<a href="#">The Audit View</a>  <small>457</small>
<b>Code Engineering</b>	Select this option to access a submenu of options to generate, view and synchronize code for the selected elements.		<a href="#">Element Code Engineering Menu</a>  <small>664</small>

<b>Rule Composer</b>	For a Rule Task element, select this option to invoke the Rule Composer in Business Rule Modeling.		<a href="#">Compose Business Rules</a> <small>[1833]</small>
<b>Delete &lt;element Name&gt;</b>	Select this option to delete the element. A prompt displays for you to confirm the deletion.		
<b>Help</b>	Select this option to display the Help topic for the Project Browser.		

#### 4.1.1.1.3.1 Add Sub Menu

The **Add** submenu provides a list of appropriate elements that can be added as child elements to the selected element. It also provides options for creating child diagrams, including Composite diagrams, and for adding connectors.

**Access** **Project Browser** | **Right-click element** | **Add**

Option	Action	See also
<b>Port</b>	Select this option to add a Port element to the border of the selected element.	<a href="#">Port</a> <small>[1384]</small>
<b>Activity</b>	Select this option to add an Activity behavior element and one of its associated diagrams to the selected classifier element.	<a href="#">Activity</a> <small>[1279]</small> <a href="#">Classifiers and Instances</a> <small>[1009]</small>
<b>Interaction</b>	Select this option to add an Interaction behavior element and one of its associated diagrams to the selected classifier element.	<a href="#">Interaction Occurrence</a> <small>[1317]</small> <a href="#">Classifiers and Instances</a> <small>[1009]</small>
<b>State Machine</b>	Select this option to add a State Machine Behavior element and one of its associated diagrams to the selected classifier element.	<a href="#">State Machine</a> <small>[1338]</small> <a href="#">Classifiers and Instances</a> <small>[1009]</small>
<b>RuleFlow Activity</b>	Select this option to add a Rule Flow Activity element and associated diagram to a Class, in Business Rule Modeling.	<a href="#">Create a Rule Flow Activity</a> <small>[1826]</small>
<b>Composite Structure Diagram</b>	Select this option to set the selected element as a Composite Element and create a new Composite Structure Diagram (if none exists). If the selected element already contains a child diagram, the existing diagram will instead be referenced as the element's composite diagram.	<a href="#">Composite Elements</a> <small>[936]</small> <a href="#">Composite Structure Diagram</a> <small>[1188]</small>

Option	Action	See also
<b>Select Composite Diagram</b>	Select this option to set the selected element as Composite and link it to a specified diagram. The linked diagram may exist anywhere within the current repository.	
<b>Add Diagram</b>	Select this option to create a diagram to explain or expand on the selected classifier element, using the New Diagram dialog.	<a href="#">Add New Diagrams</a> [822]
<b>Add Custom Reference</b>	Select this option to set up cross references between elements in a diagram and the selected element in the Project Browser.	<a href="#">Cross references</a> [916]
<b>Create Link</b>	Select this option to create a connector to another element.	<a href="#">Create Connector in Project Browser</a> [1118]

#### Notes

- Elements such as Actors, Classes and Activities can define a large amount of information that can be conveniently represented by or expanded in a child diagram; the **Add** sub-menu for these elements provides all of the options listed above
- Elements such as Timing, Exit and History have much more specific functions that do not require expansion; the **Add** sub-menu for these elements only provides the **Create Link** option

#### 4.1.1.1.3.2 Element Copy/Paste Submenu

The **Copy/Paste** context menu provides a range of options for copying or moving elements between Packages or between models, within the Project Browser.

**Access** **Project Browser | Right-click element | Copy/Paste**

Option	Action	Shortcut	See also
<b>Copy Element(s) to Clipboard</b>	Select this option to copy the selected elements to the clipboard, to be pasted into another Package in the same project file or a different project file using the <b>Paste Elements from Clipboard</b> menu option in the <b>Package</b> context menu.		
<b>Cut</b>	Select this option to remove the selected element from its parent Package, to be pasted under another element or Package. The selected element remains where it is until the <b>Paste</b> operation (below) is invoked.	<b>Ctrl+X</b>	<a href="#">Move Elements Between Packages</a> [913]
<b>Paste</b>	Select this option to paste a <b>cut</b> element under the selected element or Package. If the selected element is not a valid target, the <b>Paste</b> option is grayed out.	<b>Ctrl+V</b>	

Option	Action	Shortcut	See also
	(This option is not available for <b>copied</b> elements - use the <b>Paste Package from Clipboard</b> menu option in the <b>Package</b> context menu.)		<a href="#">Copy Elements Between Packages</a> [915]
<b>Copy Documentation Bookmark</b>	When using the Legacy Report Generator, if you want to create a complex document that consists of <b>sections</b> of one or more document reports, rather than one <b>complete</b> report, you can create a master document in Microsoft Word™ and within it create links to the report sections using their <b>bookmarks</b> .  Select this menu option to copy the bookmark to the clipboard, to be pasted into a search or other reference to the bookmark.		<a href="#">Document Bookmarks</a> [2730]
<b>Copy Node Path to Clipboard</b>	Select this option to copy the Package's node path (hierarchy structure) to the Enterprise Architect clipboard, as a reference.		
<b>Copy Node GUID to Clipboard</b>	Select this option to copy the Package's node GUID to the Enterprise Architect clipboard, as a reference.		

#### Learn more

- [Element Options in the Project Browser](#) [660]


#### 4.1.1.3.3 Element Code Engineering Menu

The Code Engineering submenu provides a set of options for generating, viewing and synchronizing code for an element or for a group of elements.



**Access** **Project Browser | Right-click element | Code Engineering**

Option	Action	Shortcut	See also
<b>Generate Source Code</b>	Select this option to generate the code for this element.	<b>F11</b>	<a href="#">Generate Source Code</a> [2117]
<b>Synchronize With Code</b>	Select this option to synchronize the element in the Project Browser with the source code.  You can also synchronize Class elements from the source code on a diagram, and reverse engineer or forward engineer model Packages and code to synchronize them and incorporate changes made in one but not in the other.	<b>F7</b>	<a href="#">Import Source Code</a> [2136] <a href="#">Operation Menu - Project Browser</a> [667] <a href="#">Update Package Contents</a> [2117]








Option	Action	Shortcut	See also
<b>View Source Code</b>	Select this option to view the source code of the element.	<b>F12</b>	<a href="#">Editing Source Code</a> 
<b>Open Source Directory</b>	Select this option to open the source directory containing the source code for the element.	<b>Ctrl+Alt +Y</b>	




#### Learn more

- [Element Options in the Project Browser](#) 
- [Generate Source Code](#) 

#### 4.1.1.1.4 Diagram Menu - Project Browser

Right-click on a diagram name in the Project Browser to open the **Diagram** context menu.

Option	Action	Shortcut	See also
<b>Extensions</b>	Select this option to access a submenu of enabled Technologies on the system (for example, ArcGIS or TOGAF), each of which provides options for performing technology-specific operations on the diagram.  This option is available in the Professional, Corporate and Extended editions of Enterprise Architect.		<a href="#">Geodatabase Design for ArcGIS</a> 
<b>Scripts</b>	Select this option to list the scripts enabled for execution directly from the Project Browser.  (This option does not display if no Project Browser scripts exist.)		<a href="#">Scripts</a> 
<b>Properties</b>	Select this option to display the diagram Properties dialog, to view and modify the diagram's properties.	<b>F5</b>	<a href="#">Set Diagram Properties</a> 
<b>Open</b>	Select this option to open the diagram in the Diagram View.		
<b>View Diagram As</b>	Select this option to display the contents of the selected diagram in an alternative viewing mode: <ul style="list-style-type: none"> <li>• <b>List View</b> - Show all elements on the selected diagram in a tabular list format</li> </ul>		<a href="#">Diagram List</a>  <a href="#">The Gantt View</a> 

Option	Action	Shortcut	See also
	<ul style="list-style-type: none"> <li>• <b>Gantt View</b> - Show all elements on the selected diagram in a Gantt Chart format</li> </ul>		
<b>View as Diagram</b>	Select this option to switch the display format back to elements rendered on a diagram.		
<b>Copy Diagram</b>	Select this option to copy the diagram for pasting into another location.		<a href="#">Copy a Diagram</a> 
<b>Copy Documentation Bookmark</b>	Select this option to copy a bookmark on the diagram to the clipboard.		<a href="#">Document Bookmarks</a> 
<b>Add Custom Reference</b>	Select this option to add this diagram as a cross reference to other elements.		<a href="#">Set Up Cross References</a> 
<b>Print Diagram(s)</b>	<p>Select this option to print the currently-selected diagram or diagrams (hold <b>Ctrl</b> or <b>Shift</b> while selecting).</p> <p>The Print dialog displays, on which you specify the print parameters.</p>	<b>Ctrl+P</b>	
<b>Copy Reference</b>	<p>Select this option to copy a reference to the diagram node to the Enterprise Architect clipboard.</p> <p>Select the appropriate sub-option to copy the:</p> <ul style="list-style-type: none"> <li>• selected hierarchy structure (node path) or</li> <li>• node GUID</li> </ul>		
<b>Cut</b>	<p>Select this option to <b>remove</b> the selected diagram from its current location, to be pasted under another Package or element. The selected diagram remains where it is until the Paste operation is invoked.</p> <p>To paste the diagram, use the:</p> <ul style="list-style-type: none"> <li>• <b>Copy/Paste   Paste Diagram</b> option on the Package context menu, or</li> <li>• <b>Copy/Paste   Paste</b> option on the element context menu</li> </ul>		
<b>Paste</b>	Not used.		

Option	Action	Shortcut	See also
<b>Compare Diagram to Baseline</b>	<p>Select this option to perform a visual comparison of the selected diagram against a version from a previously saved baseline.</p> <p>In this comparison, you can roll-back individual changes made to this diagram since the baseline was created.</p>		<a href="#">Check Visual Changes to Diagrams</a> <sup>[466]</sup> <a href="#">Manage Baselines</a> <sup>[467]</sup>
<b>Move up</b>	Select this option to move the diagram up in the list of diagrams within this Package.		
<b>Move down</b>	Select this option to move the diagram down in the list of diagrams within this Package.		
<b>Delete '&lt;diagram name&gt;'</b>	<p>Select this option to delete the selected diagram.</p> <p>A prompt displays to confirm the deletion.</p>		
<b>Delete selected items</b>	<p>Select this option to delete <b>several</b> selected diagrams in the same Package (hold <b>Ctrl</b> or <b>Shift</b> while selecting).</p> <p>A prompt displays to confirm the deletion of each diagram separately, or all together.</p>		
<b>Help</b>	Select this option to display the Help topic for the Project Browser.		

#### 4.1.1.1.5 Operation Menu - Project Browser

To display the **Operation** (or Method) context menu, right-click on an Operation name in the Project Browser.

A very similar menu displays if you right-click on an **Attribute** name in the Project Browser.

Option	Action	Shortcut	See also
<b>Extensions</b>	<p>Select this option to access a submenu of enabled Technologies on the system (for example, ArcGIS or TOGAF), each of which provides options for performing technology-specific operations on the operation.</p> <p>This option is available in the Professional, Corporate and Extended editions of Enterprise Architect.</p>		<a href="#">Geodatabase Design for ArcGIS</a> <sup>[1944]</sup>
<b>Scripts</b>	Select this option to list the scripts enabled for execution directly from the Project Browser.		<a href="#">Script Group Properties</a> <sup>[2794]</sup>

Option	Action	Shortcut	See also
	<p>(This option does not display if no Project Browser scripts exist.)</p> <p>Click on a script name to run that script.</p>		
<b>Executable Statemachine</b>	(For Attributes.) Select this option to access a submenu of options for treating the <b>Attribute</b> like an <b>Executable Statemachine Artifact</b> and generating code from it.		<a href="#">Code Generation - State Machines</a> <sup>[2122]</sup>
<b>Generate Code</b>	<p>Select this option to generate code from the operation, within its parent Class.</p> <p>The Generate Code dialog displays.</p>	<b>F11</b>	<a href="#">Generate a Single Class</a> <sup>[2113]</sup>
<b>Synchronize With Code</b>	<p>Select this option to synchronize the operation with its code.</p> <p>You can also synchronize Class elements from the source code on a diagram, and reverse engineer or forward engineer model Packages and code to synchronize them and incorporate changes made in one but not in the other.</p>	<b>F7</b>	<a href="#">Import Source Code</a> <sup>[2136]</sup> <a href="#">Element Options in the Project Browser</a> <sup>[660]</sup> <a href="#">Update Package Contents</a> <sup>[2117]</sup>
<b>View Source Code</b>	Select this option to open the Source Code Viewer and display the code for the operation.	<b>F12</b>	<a href="#">Editing Source Code</a> <sup>[2146]</sup>
<b>Operation Properties</b>	Select this option to display the Properties dialog for the operation, to view and edit the properties of the operations in the parent element.		<a href="#">General Properties of Operations</a> <sup>[1015]</sup>
<b>Find in all Diagrams</b>	(For Operations.) Select this option to locate any elements derived from the operation in any diagram in the models.	<b>Ctrl+U</b>	<a href="#">Show Element Use</a> <sup>[910]</sup>
<b>Copy Reference</b>	<p>Select this option to copy a reference to the operation to the Enterprise Architect clipboard.</p> <p>Select the appropriate sub-option to copy the:</p> <ul style="list-style-type: none"> <li>Selected hierarchy structure (node path)</li> <li>Node GUID</li> </ul>		
<b>Cut</b>	<p>Select this option to cut the operation from its current element and paste it into another element. The operation does not move from the original element until you paste it into the target.</p> <p>Use the element <b>Copy/Paste   Paste</b> menu option to paste the operation into its new parent.</p>		<a href="#">Element Copy/Paste Submenu</a> <sup>[663]</sup>

Option	Action	Shortcut	See also
<b>Paste</b>	Not used.		
<b>Delete Operation</b>	Select this option to delete the operation.  The operation is immediately removed from the element.		
<b>Help</b>	Select this option to display the Help topic for the Project Browser.		

### Notes

- You can display an equivalent context menu for an attribute by right-clicking on the attribute in the Project Browser

#### 4.1.1.2 Project Browser Toolbar

The Project Browser toolbar provides access to a range of operations you can perform on your project structure and content.



**Access** [View | Project Browser](#)


### Options






Option	Action	Shortcut	See also
<b>New Model from Pattern</b>	Create a new Model Package in the project, from a predefined UML or Technology pattern.  The list of available Model Pattern Technologies includes the MDG Technology Builder, which creates a Model package containing templates that you can use to generate a new MDG Technology containing a set of Profiles.	<b>(Ctrl+Shift+M)</b>	<a href="#">Model Wizard</a> <sup>[753]</sup>
<b>New Package</b>	Create a new child package under the selected package.		<a href="#">Add a Package</a> <sup>[772]</sup>
<b>New Diagram</b>	Create a new child diagram under the selected package or element.		<a href="#">Add New Diagrams</a> <sup>[822]</sup>

<b>New Element</b>	Create a new child element under the selected package or element.		<a href="#">Add Elements Directly to Packages</a> <sup>[903]</sup>
<b>Find in Project Browser</b>	Perform a simple search for a text string in the Project Browser.		
<b>Documentation</b>	Provide options to generate, on the selected Package in the Project Browser, a: <ul style="list-style-type: none"> <li>Document report</li> <li>Web report or</li> <li>Diagram Only report</li> </ul>	(F8) (Shift+F8) (Ctrl+Shift+F8)	<a href="#">Generate Document Reports</a> <sup>[2642]</sup> <a href="#">Web Reports</a> <sup>[2744]</sup> <a href="#">Diagrams Only Report</a> <sup>[2741]</sup>
<b>Code and DB Generation</b>	Provide options to: <ul style="list-style-type: none"> <li>Generate source code</li> <li>Generate DDL</li> <li>Import a source directory</li> <li>Import a binary module</li> <li>Import a database schema</li> <li>Generate package contents to synchronize with package code</li> <li>Reset the source code language</li> </ul> <p>All these options operate on the selected package.</p>	(Ctrl+Alt+K) (Ctrl+Shift+U) (Ctrl+Alt+M)	<a href="#">Generate a Package</a> <sup>[2115]</sup> <a href="#">Generate DDL for a Package</a> <sup>[2381]</sup> <a href="#">Import a Directory Structure</a> <sup>[2142]</sup> <a href="#">Import Binary Module</a> <sup>[2143]</sup> <a href="#">Import Database Schema</a> <sup>[2376]</sup> <a href="#">Update Package Contents</a> <sup>[2117]</sup> <a href="#">Reset Options</a> <sup>[2279]</sup>
<b>Move Element Up</b>	Move the selected package or element further up the Project Browser, within its parent package.		
<b>Move Element Down</b>	Move the selected package or element further down the Project Browser, within its parent package.		
<b>Help</b>	Display the Help topic on the Project Browser.		



#### 4.1.1.3 Project Browser Icon Overlays


The Project Browser displays the status of each package in the model by overlaying status icons on the package icon. The following table describes what each overlaid icon means.

Icon Overlay	Indicates that...	See also
	This package is controlled and is represented by an XML file on disk; version control either is not being used or is not available.	<a href="#">Controlled Packages</a> <sup>[484]</sup>

Icon Overlay	Indicates that...	See also
	You can edit the package.	
	This package is version controlled and checked out to you, therefore you can edit the package.	<a href="#">Use Version Control</a> <sup>[422]</sup>
	This package is version controlled and not checked out to you, therefore you cannot edit the package (unless you check the package out).	<a href="#">Use Version Control</a> <sup>[422]</sup>
	This package is version controlled, but you checked it out whilst not connected to the version control server.  You can edit the package but there could be version conflicts when you check the package in again.	<a href="#">Offline Version Control</a> <sup>[392]</sup>
	This package is a namespace root, which denotes where the namespace structure starts; packages below this point are generated as namespaces to code.	<a href="#">Namespaces</a> <sup>[2120]</sup>
	The version-controlled package has been flagged as <i>Read Only</i> , using a package control method applied through the Automation Interface - as a script, for example.  The icon displays with other package control icons except for the 'blue box' icon (above) which takes precedence.  The <i>Read Only</i> flag can be applied to packages whether they are checked-in or checked-out.	<a href="#">Package Class</a> <sup>[2835]</sup>
<MDG Add-In icon>	MDG Add-Ins specify their own icon to denote that this branch of the model belongs to that Add-In.  All packages connected to an MDG Add-In correspond to a namespace root, so the namespace root icon is not displayed.	<a href="#">MDG Add-Ins</a> <sup>[3097]</sup>

Similarly, the Project Browser indicates attribute and operation scope status with icons. The following table describes what each indicator icon means.

Icon Overlay	Indicates that...	See also
	The attribute or operation is scoped as protected.	
		
	The attribute or operation is scoped as private.	

Icon Overlay	Indicates that...	See also
		

### Notes

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions, if project User Security is on, the Project Browser also has element locking indicators (red and blue exclamation marks) that indicate the lock status of individual elements and packages

The availability of these elements for editing depends on whether user locks are required or not

### Learn more

- [User Security](#)<sup>[316]</sup>
- [Locked Element Indicators](#)<sup>[340]</sup>

#### 4.1.1.4 Order Package Contents

Ordering elements is very important when it comes to structuring your model, especially packages. Report documents honor any custom ordering when printing documentation. You can change the order of elements listed in the Project Browser.

By default, elements are first listed in:

- Order of type, then
- Order of set position, then
- Alphabetically

You can use the context menu options to move an element up or down within its type, but not outside its type. This means you can re-sequence Packages or Diagrams or Use Cases, but you cannot mix up elements.

However, you can change this default behavior to allow elements to be re-ordered within the package, **regardless** of type.

### Learn more

- [Set Default Behavior](#)<sup>[672]</sup>

#### 4.1.1.5 Set Default Behavior

The General page of the Options dialog provides several options for altering the look and behavior of the Project Browser.

**Access** [Tools](#) | [Options](#) | [General](#)

Panel	Option	Effect	See also
<b>Double-click On Browser</b>	<b>Shows Properties</b>	When an item in the Project Browser is double-clicked, the item's	



Panel	Option	Effect	See also
		Properties dialog (if available) displays.	
	<b>Opens Branch</b>	When an item in the Project Browser is double-clicked, the hierarchy expands to show the item's children.  If there are no children, nothing happens.	
	<b>Opens Branch &amp; Diagram</b>	As above, but also opens the first diagram beneath the item, if applicable.	
<b>Project Browser</b>	<b>Allow Free Sorting</b>	Enables you to re-order elements within a package <b>regardless</b> of type, within the Project Browser.	<a href="#">Order Package Contents</a> <sup>[672]</sup>
	<b>Show Stereotypes</b>	When a stereotype is defined for an element, the stereotype name then displays in front of the element name.  You set the stereotype of an element in its Properties dialog.  Shut down and restart Enterprise Architect to enable this change to take effect.	

### 4.1.2 Package Browser

If you want to quickly create and operate on elements in a **Package**, you can use the **Package Browser** as a workbench. The Package Browser is a tabular, editable view of the elements in a selected Package, which can be displayed either in the Diagram View or as a floating window, and as either a List display or a Gantt chart. It can be particularly useful for setting up a Package for a specific purpose, such as when an Analyst creates and maintains formal requirement definitions within the model. When you have checked the contents of the Package, you can either print the list of elements or generate a document report on all or a subset of the elements.

The Package Browser is related to, and can be used in conjunction with, the:

- **Diagram List**, which lists the elements in a selected **diagram**, and
- **Specification Manager**, which is a document-based interface to the Package, providing the means of creating and reviewing elements as text representations of model objects in that Package


**Access** **Project Browser > Package | View | Package Browser (Ctrl+Alt+R)**


**Project Browser > right-click Package | List View**

**Project Browser > right-click Package | Gantt View**

**Diagram > right-click Package | Package Browser**

**Features**

Feature	Detail	See also
<b>Select Standard View or Gantt View</b>	<p>When you select one of the menu paths, above, a submenu displays for you to select the <b>Standard View</b> of the Package Browser, or the <b>Gantt View</b>.</p> <ul style="list-style-type: none"> <li>• <b>Standard (or List) View</b> lists all the elements in the Package, with their properties; it has two formats, in which the elements are either grouped by model hierarchy or grouped (by you) according to one or more properties</li> <li>• <b>Gantt View</b> lists all the elements in the Package, grouped by model hierarchy, but showing the percent-completion of all tasks associated with each element with the work timeline for each element and for each resource assigned to work on the element</li> </ul> <p>Once you have displayed one of these views, you can switch to the other by right-clicking on the display and selecting the appropriate context menu option:</p> <ul style="list-style-type: none"> <li>• <b>Switch to List View</b> or</li> <li>• <b>Switch to Gantt View</b></li> </ul>	<p><a href="#">Element View</a> <sup>[542]</sup></p> <p><a href="#">The Gantt View</a> <sup>[594]</sup></p> <p><a href="#">Package Browser Options</a> <sup>[679]</sup></p>
<b>Switch Standard View Formats</b>	<p>In Standard View, the Package Browser has two formats:</p> <ul style="list-style-type: none"> <li>• <b>User Defined</b> format, initially a flat list, where <b>you</b> can define how the element information is grouped on the screen using the value-grouping band below the toolbar</li> <li>• <b>Model Hierarchy</b> format, where the elements are already organized into their Package and element hierarchies in the display</li> </ul> <p>You switch between these formats using the <b>Show Hierarchy</b> button (  ) in the toolbar.</p>	<p><a href="#">List Header</a> <sup>[677]</sup></p>
<b>Change Standard View display</b>	<p>It is possible to reorganize the information in the Project Browser in a number of ways:</p> <ul style="list-style-type: none"> <li>• In User Defined format, sort the items by any column value in ascending or descending order, by clicking on the column header; initially the elements are listed in numerical order (if level numbering is turned on in the Project Browser) or alphabetical order within type</li> <li>• In User Defined format, apply a diagram filter for elements (the <b>Fade</b>, <b>Gray Scale</b> and <b>Hide</b> modes reduce the number of elements listed; <b>Select</b> mode has no effect)</li> <li>• In either format, change the sequence of columns by dragging column headers left or right <ul style="list-style-type: none"> <li>• In Model Hierarchy format, the <b>Name</b> column is always on the left; you cannot move any other column into that position, although you can rearrange the order of the rest of the columns</li> </ul> </li> </ul>	<p><a href="#">Diagram filters</a> <sup>[718]</sup></p>

Feature	Detail	See also
	<p>Because of this, if you group or sort information in User Defined format and switch to Model Hierarchy format your information structure is altered, and it is not restored when you switch back to User Defined format</p>	
<b>Show Elements in a Child Package</b>	<p>Double-click on the child Package in the list to re-focus the display, showing the contents of that Package only.</p> <p>In either format, click on the  button in the toolbar to expand or collapse the contents of the child Packages.</p>	
<b>Display Element Properties</b>	Double-click on the <b>element</b> line to display the element Properties dialog.	
<b>Select Items</b>	<p>Select:</p> <ul style="list-style-type: none"> <li>• an element by clicking on it</li> <li>• a specific value by clicking twice on it (not double-clicking); either the value becomes directly editable (overtyping or selecting from drop-down) or the Properties dialog displays in which you can edit the value</li> <li>• several individual elements by holding <b>Ctrl</b> as you click on them</li> <li>• a range of elements by holding <b>Shift</b> as you click on the first and last in the range</li> </ul>	
<b>Add new items to the Package</b>	<p>Click on a listed element and press <b>Ctrl+N</b> or <b>Insert</b>, or right-click and select the <b>Add New</b> context menu option. The New Element dialog displays; complete the fields and click on the <b>Create</b> button.</p> <ul style="list-style-type: none"> <li>• In Model Hierarchy format, the new element is inserted as a <b>sibling</b> of the <b>selected element</b>, in its parent Package</li> <li>• In User Defined format, the new element is added to the <b>root Package</b> selected for display in the Package Browser</li> </ul>	<a href="#">Add Elements Directly to Packages</a> <sup>[903]</sup>
<b>Add a child element to the selected element</b>	<p>Click on the selected element and press <b>Ctrl+Shift+N</b>. The New Element dialog displays; complete the fields and click on the <b>Create</b> button.</p> <ul style="list-style-type: none"> <li>• in Model Hierarchy format, the new element is inserted as a <b>child</b> of the <b>selected element</b></li> <li>• in User Defined format, the new element is added to the <b>root Package</b> selected for display in the Package Browser</li> </ul>	<a href="#">Add Elements Directly to Packages</a> <sup>[903]</sup>
<b>Add elements to the Package Browser</b>	Drag the required elements from anywhere in the Project Browser. The element is listed in the Package Browser, and moved into the target Package in the Project Browser.	

Feature	Detail	See also
	You can also drag elements <i>from</i> the Package Browser <i>into</i> a package in the Project Browser, in either List view or Gantt view.	
<b>Add elements to a diagram from the Package Browser</b>	<p>In either List view or Gantt view, you can drag one or more elements from the Package Browser onto a diagram to immediately create an <b>instance</b> of each element on the diagram (and in the Project Browser for the diagram's parent Package).</p> <p>Click each element (pressing <b>Ctrl</b> or <b>Shift</b> for multiple selections) to select it before dragging it onto the diagram.</p> <p>One advantage of dragging elements from the Gantt view is that it provides the additional filter on task or resource, so that you can isolate elements to be copied to a diagram or Package specific to a task or resource.</p>	
<b>Delete elements from the list</b>	<p>Select the item and press <b>Ctrl+D</b>. This deletes the element from all diagrams and from the model itself.</p> <p>In Model Hierarchy format, you cannot delete a parent element until all its child elements have been removed or deleted.</p>	
<b>Include each element's notes (documentation)</b>	<p>Notes are shown underneath the element.</p> <p>To add or edit notes, click on the item and press <b>Ctrl+Shift+Space</b> to transfer control to the Notes window, in which you create or edit the note text.</p> <p>Alternatively, right-click on the element line and select the <b>Edit Notes</b> context menu option.</p>	<a href="#">Package Browser Options</a> <sup>[679]</sup>
<b>Review the element's custom (advanced) properties</b>	<p>Click on the item and press <b>Ctrl+Enter</b>.</p> <p>The Custom Properties dialog for the element displays.</p>	
<b>Further Work</b>	You can do further work on the Package Browser using the toolbar and context menu options.	<a href="#">Package Browser Options</a> <sup>[679]</sup>

#### Learn more

- [Diagram List](#) <sup>[684]</sup>
- [Specification Manager](#) <sup>[1728]</sup>

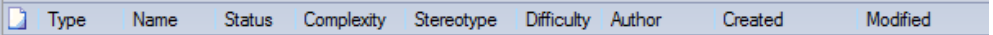
#### 4.1.2.1 List Header


In your work on your project you might use any of a number of 'listing' reports in which information is organized in several columns. Such reports include the:

- Package Browser
- Diagram List
- Model Search
- Model Mail window
- Attribute and Operation dialog General pages
- Testing and Maintenance tabs of the QA Reports and Metrics view
- The Gantt View
- The Project Management window

In these reports, you can quickly and easily reorganize the sequence and combination of columns to present the information in the most appropriate format to suit your needs, and filter the data according to the values of the text strings displayed in the column fields.

#### Reorganize layout of display

Tool	Description
<b>View Header</b>	<p>Drag a column header here to group by that column.</p>  <p>The View Header defines the columns of information that are presented by the report, and the order in which data items are presented.</p> <p>To review the column headings:</p> <ul style="list-style-type: none"> <li>• Right-click on the header to display the <b>Field Chooser</b> context menu option; the Field Chooser dialog displays</li> </ul> <p>This enables you to add columns to or remove columns from the output; between them, the View Header and Field Chooser dialog show the full range of column headers available.</p> <p>To add a column heading to the View Header:</p> <ul style="list-style-type: none"> <li>• Drag it from the Field Chooser dialog onto the header to the position you want the column of data to display</li> <li>• When you have selected the column headings you require, click on the red cross in the top right corner of the Field Chooser dialog to close it</li> </ul> <p>To remove a column from the output:</p> <ul style="list-style-type: none"> <li>• Drag the column heading downwards out of the View Header</li> </ul> <p>To change the sequence of columns:</p> <ul style="list-style-type: none"> <li>• Drag the required column header left or right, as appropriate</li> </ul>

Tool	Description
<b>Value Grouping</b>	<p>For most reports, and when the Package Browser or Diagram List is in <i>User Defined</i> format, you can organize the reported data according to the value of one or more of the column categories (this facility is not available in the Package Browser in <i>Model Hierarchy</i> format).</p> <p>You might organize the data by <b>Type</b>, and within each Type by <b>Name</b>; if you then click on any other column heading, the data within this grouping is further sorted with the values of the selected column (for example, <b>Created</b>) in ascending or descending order.</p> <p>To set up the value grouping, drag the column heading representing the primary grouping (such as <b>Author</b>) onto the <b>Drag a column header here to group by that column</b> field.</p> <p>Now drag the column heading for the next level of grouping (such as <b>Status</b>) to the right of the first heading; the two heading titles display as connected blocks, as shown below:</p>  <p>You can, if required, add further levels of grouping by dragging other column headings onto the hierarchy (such as <b>Created</b>), and restructure the order by dragging existing or additional headings into the level you want them to hold.</p> <p>For example, you could make <b>Type</b> the secondary grouping by dragging it to the right of <b>Author</b>, or drop <b>Status</b> between <b>Author</b> and <b>Type</b>.</p> <p>To remove a grouping level, drag the appropriate column heading out of the sequence and below the View header; any subordinate groupings move up a level.</p>
<b>Filter Bar</b>	<p>The Filter Bar displays underneath the column headings; you can hide or show it using the <b>Toggle Filter Bar</b> icon in the toolbar.</p> <p>The Filter Bar consists of a row of fields, one field per column; as you type a value into one of these fields, the item lists are automatically filtered to show entries with a value in the corresponding column that contain the text string in the field.</p> <p>For example:</p> <ul style="list-style-type: none"> <li>If you type the letter <b>F</b> in the Filter Bar field under the <b>Author</b> column, the results immediately reduce to entries for which the letter <b>F</b> occurs anywhere in the name (<b>F</b>red, <b>F</b>itzgibbon, <b>R</b>ufus, <b>W</b>olf)</li> <li>If you then go on to type <b>Fi</b>, the results immediately reduce again to entries for which the combination <b>fi</b> appears in the name (<b>F</b>itzgibbon, <b>A</b>mal<b>fi</b>)</li> </ul> <p>For reports and displays that identify the percentage completion of a task or piece of work, you can specify a percentage value or a range to display items that match that value. In the Filter Bar, in the field under the <b>%Completed</b> or <b>Complete</b> field, type:</p> <ul style="list-style-type: none"> <li><b>&lt;x</b> to display items less than x percent complete; for example, <b>&lt;30</b> (all items less than 30 percent complete, but <b>not</b> those exactly 30 percent complete)</li> <li><b>&gt;x</b> to display items more than x percent complete; for example, <b>&gt;30</b> (all items more than 30 percent complete, but <b>not</b> those exactly 30 percent complete)</li> <li><b>x</b> to display items <b>exactly</b> x percent complete; for example, <b>30</b> (all items exactly 30 percent complete)</li> <li><b>-x</b> to display items exactly x percent complete and less than that; for example, <b>-30</b></li> </ul>

Tool	Description
	<p>(all items less than 30 percent complete <b>and</b> those exactly 30 percent complete)</p> <ul style="list-style-type: none"> <li>• <b>x-</b> to display items exactly x percent complete and more than that; for example, <b>+30</b> (all items more than 30 percent complete <b>and</b> those exactly 30 percent complete)</li> <li>• <b>x-y</b> to display items between x-1 percent complete and y+1 percent complete; for example <b>30-45</b> (all items <b>between</b> 30 and 45 percent complete, including those <b>exactly</b> 30 percent and 45 percent complete)</li> </ul>

#### Learn more

- [Package Browser](#)<sup>[673]</sup>
- [Diagram List](#)<sup>[684]</sup>
- [Report View](#)<sup>[546]</sup>
- [Review Allocated Work](#)<sup>[555]</sup>
- [Resource View](#)<sup>[538]</sup>
- [Element View](#)<sup>[542]</sup>
- [Add, Modify and Delete Tasks](#)<sup>[527]</sup>
- [The Gantt View](#)<sup>[594]</sup>

#### 4.1.2.2 Package Browser Options



In building and managing your model you can use the Package Browser as a workbench to perform many operations, such as reorganizing the package structure, adding new packages and elements, and generating reports on the package contents. You can also modify the display by:

- Filtering the elements listed, by type or by Toolbox group
- Adding columns to show the values of specific tags on the elements

To perform all of these operations, you select options on either the Project Browser toolbar or the context menu.




**Access** **Project Browser > Right-click package | Package Browser > Toolbar**  
**Project Browser > Right-click package | Package Browser > right-click on items**

#### Toolbar Options

Icon	Action	Shortcut	See also
	Add a new element to the package. The New Element dialog displays.	<b>Ctrl+N</b>	<a href="#">Add Elements Directly to Packages</a> <sup>[903]</sup>
	Display the Notes window, to add or edit notes for the selected element. The cursor is positioned at the start of the text entry area of the window.	<b>Ctrl+Shift+Space</b>	<a href="#">Notes</a> <sup>[1142]</sup>

	<p>Delete the selected element from all diagrams and from the model.</p> <p>A confirmatory prompt displays; click on the <b>Yes</b> button.</p>	<b>Ctrl+D</b>	
	<p>Print the current contents of the Package Browser.</p> <p>The Print dialog displays, on which you specify the print job.</p>		
	<p>Display the Generate Documentation dialog, to create one document report on the selected element(s). You can apply the full facilities of this dialog to tailor the report to your requirements.</p>		<a href="#">Generate Documentation</a> <small>[2644]</small>
	<p>Click on the drop-down arrow and select:</p> <ul style="list-style-type: none"> <li>• A specific element type from the drop-down list, or</li> <li>• <b>All</b> to list elements of all types in the package</li> </ul> <p>The report then lists only elements of the selected type(s).</p>		
	<p>Select another toolbox or technology - select the UML, Extended or MDG Toolbox category to set the types of element shown in the drop-down element filter list (above).</p>		
	<p>Toggle between:</p> <ul style="list-style-type: none"> <li>• Including child packages and their contents in the list, and</li> <li>• Showing only the first-level contents of the selected diagram or package</li> </ul>		
	<p>Toggle the display between:</p> <ul style="list-style-type: none"> <li>• Model Hierarchy format and</li> <li>• User-Defined, value-grouping format</li> </ul>		<a href="#">Package Browser</a> <small>[673]</small> <a href="#">List Header</a> <small>[677]</small>
	<p>Display a short menu from which you can select to:</p> <ul style="list-style-type: none"> <li>• Hide the contents of the Notes compartment of each element</li> <li>• Display the first few words as a preview, or</li> <li>• Display the full text of each note</li> </ul>		



	Display or hide the Filter Bar fields underneath the column headings, in any of which you can type a text string to filter the data in the corresponding column to show only data containing the typed string.		<a href="#">List Header</a> <sup>[677]</sup>
	Change the display to show the contents of the <b>parent</b> package of the selected package.	<b>Ctrl+Shift+Up</b>	
	Display Help on the Package Browser.		

### Context Menu Options

Option	Description	Shortcut	See also
<b>Properties</b>	Display the Properties dialog for the selected element.		<a href="#">Properties Dialog</a> <sup>[956]</sup>
<b>Edit Notes</b>	Add or edit notes on the element, in the Notes window.	<b>Ctrl+Shift+Space</b>	<a href="#">Notes</a> <sup>[1142]</sup>
<b>Create Linked Document</b> (Edit Linked Document)	Create (or edit) a Linked Document. (Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions.)	<b>Ctrl+Alt+D</b>	<a href="#">Linked Documents</a> <sup>[1044]</sup> <a href="#">Create Linked Document on an Element</a> <sup>[1047]</sup> <a href="#">Edit Linked Documents</a> <sup>[1048]</sup>
<b>Delete Linked Document</b>	(If the element has a linked document.) Delete the existing linked document.		
<b>Add New Element</b>	If the <b>Display elements of this type</b> field in the toolbar is set to: <ul style="list-style-type: none"> <li><b>All</b>, display the New Element dialog, through which you create an element of any type</li> <li>A specific element type, add an element of that type to the package in the Package Browser or to the diagram in the Diagram List, and to the Project Browser</li> </ul>		<a href="#">Add Elements Directly to Packages</a> <sup>[903]</sup>
<b>Find in Diagrams</b>	Display:		

Option	Description	Shortcut	See also
	<ul style="list-style-type: none"> <li>The diagram that uses the element, or</li> <li>If the element is used in multiple diagrams, the Element Usage dialog, which lists the diagrams that contain the element</li> </ul>		<a href="#">Show Element Use</a> <sup>[910]</sup>
<b>Find In Project Browser</b>	Highlight the selected element in the Project Browser.		
<b>Bookmark Item</b>	Bookmark the element.		
<b>Switch to Gantt View</b>	If in Standard (or List) View format, switch the display to Gantt View format.		<a href="#">Package Browser</a> <sup>[673]</sup> <a href="#">Element View</a> <sup>[542]</sup>
<b>Documentation</b>	<p>Generate a document report; you have two options:</p> <ul style="list-style-type: none"> <li>Generate a separate report on each selected object in the report</li> <li>Generate one report on all selected objects</li> </ul> <p>In either case, the Generate Documentation dialog displays. If you have selected to generate separate reports, the Generate Documentation dialog displays separately for each report in turn.</p> <p>You also have the option to print out the Package Browser list itself.</p>		<a href="#">Generate Documentation</a> <sup>[2644]</sup>
<b>Sort Contents</b>	<p>In <i>Model Hierarchy</i> format, synchronize the list with the Project Browser hierarchy, to ensure that all element and package hierarchies and sequences are - if necessary - updated.</p> <p>Normally changes are updated automatically, but there can be delays if changes are made outside the Package Browser.</p>		
<b>Reload</b>	Reload the list of elements to refresh the order and content with any recent changes.		
<b>Add Tag Value Column</b>	Add a column to show the values for a selected Tagged Value, for each element in the list (see below).		
<b>Columns Layout</b>	Preserve, apply or delete a profile of the column layout of the Package Browser display. A layout		

Option	Description	Shortcut	See also
	<p>consists of the number of columns and (where the workstation screen layout permits) the column widths.</p> <p>When you mouse-over this option, a submenu displays:</p> <ul style="list-style-type: none"> <li>• <b>Load Columns Layout</b> - displays a dialog in which you select to apply the default layout (all possible columns) or one that you have previously saved; click on the <b>OK</b> button to apply the layout</li> <li>• <b>Save Columns Layout</b> - displays a dialog in which you specify the name under which to save the layout - you can replace an existing layout by clicking on the drop-down list and selecting the name, or save as a new layout by typing a new name</li> <li>• <b>Delete Columns Layout</b> - displays a dialog in which you specify the layout profile to delete; when you click on the <b>OK</b> button, the profile is deleted (and cannot be retrieved again) but if that profile was in use when it was deleted, the layout persists until it is specifically changed, even if you close and re-open the Package Browser</li> </ul>		
<b>Print</b>	<p>Print the contents of the Package Browser.</p> <p>The Print dialog displays, on which you specify the print job.</p>		
<b>Delete Selected</b>	<p>Delete the selected element from the Package Browser.</p> <p>A confirmatory prompt displays; click on the <b>Yes</b> button.</p>		

### Add Tagged Value Column

If you want to show the value that any element in the list has for a particular tag, you can add a *read-only* column for the tag. There are two ways in which to do this:

Method	Action	See also
<b>1</b>	Open the Tagged Values window for the element and drag the tag onto the Package Browser; a new column is added for each tag you drag onto the list.	
<b>2</b>	Right-click on the element in the Package Browser or Diagram List and select the <b>Add Tag Value Column</b> context menu option; the Add Tag Value Column dialog displays.	

Method	Action	See also
	<p>Either type the tag name into the <b>Tagged Value</b> field, or select one of the two options:</p> <ul style="list-style-type: none"> <li>• <b>Global Tagged Values</b> - populates the <b>Tagged Value</b> drop-down list with all global Tagged Values in the project; this enables you to select a Tagged Value that is applicable to a large number of elements</li> <li>• <b>Tagged Values from the selected Element</b> - populates the <b>Tagged Value</b> drop-down list with the Tagged Values defined for the selected element (as listed on the Tagged Values window for that element); this enables you to filter the initial selection to a specific set of tags</li> </ul> <p>When you have selected the required tag, click on the <b>OK</b> button to insert a column for that tag into the Package Browser or Diagram List header.</p>	

### Notes

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if Auditing is turned on and the Package Browser is open, you can view a history of changes to any selected element or connector, in the Audit History tab of the Output window (if Security is enabled, you must have at least **Audit View** permissions to display the audit history)

### Learn more

- [Auditing](#)<sup>[446]</sup>
- [Audit History](#)<sup>[455]</sup>
- [List of Permissions](#)<sup>[329]</sup>

## 4.1.3 Diagram List

The Diagram List is a tabular, editable view of elements in a selected diagram, which can be displayed in the main workspace. You can use the Diagram List to:




- Streamline the process of creating and updating elements in a diagram selected from the Project Browser; this can be particularly useful for analysts to create and maintain formal requirement definitions within the model
- Print the list or generate a document report directly from the entries on the list

**Access** **diagram | Diagram | Show Diagram As | List View**  
**Project Browser > diagram ( Ctrl+Alt+R )**  
**diagram > Right-click | Switch to List View**

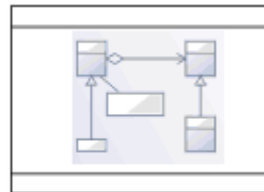
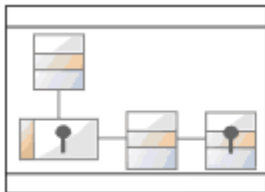
### Topics

Topic	Detail	See also
<b>The Diagram List</b>	The Diagram List is related to the Package Browser, which lists the elements in a selected package; however, the Package Browser has a wider range of options.	<a href="#">Package Browser</a> <sup>[673]</sup>

Topic	Detail	See also
<b>Formats</b>	The Diagram List is displayed in <i>User Defined</i> format, where you can change how the element information is structured on the screen using the value-grouping band below the toolbar.	<a href="#">List Header</a> <sup>[677]</sup>
<b>Change display</b>	<ul style="list-style-type: none"> <li>Sort the items by any column value in ascending or descending order, by clicking on the column header; initially the elements are listed in numerical order (if level numbering is turned on in the Project Browser) or alphabetical order within type</li> <li>Apply a diagram filter for elements (the <b>Fade</b>, <b>Gray Scale</b> and <b>Hide</b> modes reduce the number of elements listed; <b>Select</b> mode has no effect)</li> <li>Change the sequence of columns, by dragging column headers left or right</li> </ul>	<a href="#">Diagram filters</a> <sup>[718]</sup>
<b>Display Element Properties</b>	Double-click on the element entry.	
<b>Select Items</b>	Select: <ul style="list-style-type: none"> <li>an element by clicking on it</li> <li>a specific value by clicking twice on it (not double-clicking); either the value becomes directly editable or the Properties dialog displays in which you can edit the value</li> <li>several individual elements by holding ( <b>Ctrl</b> ) as you click on them</li> <li>a range of elements by holding ( <b>Shift</b> ) as you click on the first and last in the range</li> </ul>	
<b>Add new items to the diagram</b>	Create new elements on the Diagram List, using the toolbar or context menu options; these elements are automatically added to the diagram.	<a href="#">Package Browser Options</a> <sup>[679]</sup>
<b>Delete elements from the list</b>	Select the item and press ( <b>Ctrl+D</b> ).	
<b>Include each element's notes (documentation)</b>	Notes are shown underneath the element. To add or edit notes, click on the item and press ( <b>Ctrl+Space</b> ) to transfer control to the Notes window, in which you create or edit the note text.	<a href="#">Package Browser Options</a> <sup>[679]</sup>
<b>Review the element's custom (advanced)</b>	Right-click on the item and select the Properties context menu option, then select the Advanced page.	

Topic	Detail	See also
<b>properties</b>		
<b>Further Options</b>	<p>You can do further work on the Diagram List using the same toolbar and context menu options as for the Package Browser, except that the Diagram List <i>does not have</i>:</p> <ul style="list-style-type: none"> <li>The <i>Model Hierarchy</i> format and hence does not have the <b>Show Hierarchy</b> button () in the toolbar</li> <li>The  toggle button</li> <li>The  toggle button</li> </ul> <p>The Diagram List <i>does have</i> these additional context menu options:</p> <ul style="list-style-type: none"> <li><b>Diagram View</b> - show the elements of the selected diagram as a diagram instead of as the Diagram List</li> <li><b>Gantt View</b> - show the list as a Gantt chart of resource allocations</li> <li><b>Diagram Properties</b> - display the Diagram Properties dialog for the diagram</li> </ul>	<a href="#">Package Browser Options</a> <sup>[679]</sup> <a href="#">Diagram Properties</a> <sup>[823]</sup>  <a href="#">The Gantt View</a> <sup>[594]</sup>

#### 4.1.4 Model Views













The Model Views facility captures information on the objects in your model from different perspectives that you define yourself, helping you to encapsulate your model under the areas you are interested in.

**Access** **View | Model Views** (Ctrl+Shift+5)

#### Topics

Topic	Detail	See also
<b>View Root Nodes</b>	<p>There are three types of View root-node available:</p> <ul style="list-style-type: none"> <li><i>Model Views</i> - stored in the model and visible to all users; you can have many of these</li> <li><i>My Views</i> - stored locally on your machine and visible only to you; you can have only one of these</li> <li><i>Technology-defined Views</i> - read only; each View is stored with and populated by the corresponding active MDG Technology</li> </ul>	<a href="#">Model Views Operations</a> <sup>[693]</sup> <a href="#">Team Review Tools</a> <sup>[343]</sup> <a href="#">Team Review Connections</a> <sup>[363]</sup> <a href="#">Model Views</a>

Topic	Detail	See also
	<p>Additionally, there is a <i>Recent Team Reviews</i> folder that contains current correspondence from the Team Review concerning items that are held in any of the Views; this folder has a separate repository of postings for each Team Review server connection you access through the model. You can control how recent these postings must be, and how many are to be listed.</p> <p>When you open the Model Views window for the first time on a project, a <i>Model Views</i> root section, <i>My Views</i> root section and <i>Recent Team Reviews</i> folder are added for you. These can not be deleted or renamed; however, you can create further Model View root nodes that you can modify and delete.</p>	<a href="#">Context Menus</a> <sup>[690]</sup>
<b>Subordinate Folders</b>	<p>Under the <i>My Views</i> root node you can add a single level of <i>View</i> folders, which enable you to group <i>Search View</i> folders as best suit your requirements.</p> <ul style="list-style-type: none"> <li>• A <i>Search View</i> is a folder of elements or structures that you assemble by assigning a model search to the folder; when you double-click on or expand the folder, the search runs and refreshes the folder contents</li> <li>• You can also set a search to refresh at a defined interval, and to notify you if new results are found</li> </ul> <p>Under a <i>Model View</i> root node you can create Views folders that can contain Search View folders, <i>Slideshow</i> folders and <i>Favorites</i> folders.</p> <ul style="list-style-type: none"> <li>• A <i>Slideshow folder</i> contains diagrams only, which you can display as a slide show with diagrams being shown in the sequence in which they are listed in the folder</li> </ul> <p>One folder represents one slide show - you can run the slide show automatically or manually; in either case the diagrams are closed after they have been displayed</p> <ul style="list-style-type: none"> <li>• A <i>Favorites</i> folder gives you easy access to commonly-used items in the Project Browser; to create hyperlinks in a Favorites folder to the required items in the Project Browser, drag items from the Project Browser into the Favorites folder</li> </ul> <p>You can also export all of the View folders containing Views from any root section as an XML file, and import a <i>Views XML file</i> as an additional, editable <i>Model View</i> root node.</p> <p>These are single-level items; if you drag a Package into the Favorites folder, you cannot expand that Package there.</p> <p>To select specific items inside a Package, expand it in the Project Browser and then drag the items into the Favorites folder.</p>	<a href="#">Model Views Operations</a> <sup>[693]</sup> <a href="#">Diagram Slide Show</a> <sup>[695]</sup>

Topic	Detail	See also
<b>Key to Model View Contents</b>	<p>Items on the model view can be identified by their accompanying icons:</p> <ul style="list-style-type: none"> <li> - A Model Views root node</li> <li> - A View Folder</li> <li> - A Favorites Folder</li> <li> - A View (search-based)</li> <li> - A Slideshow Folder</li> <li> - The My Views root node</li> <li> - The UPDM (DoDAF-MODAF) View (a Technology-defined View)</li> <li> - The <i>Recent Team Reviews</i> folder</li> <li> - A repository of documents from a selected Team Review, concerning elements in the Model Views</li> <li> - The recent posts folder for the selected team review</li> </ul>	
<b>Accessing Operations</b>	<p>To perform operations on the Model View contents, you can use the:</p> <ul style="list-style-type: none"> <li>• Model Views window Toolbar or the</li> <li>• Item context menus</li> </ul> <p>Each level of the Model Views hierarchy has a slightly different context menu.</p>	<a href="#">Model Views Toolbar</a> <sup>[688]</sup> <a href="#">Model Views Context Menus</a> <sup>[690]</sup>

#### 4.1.4.1 Model Views Toolbar

**Access** **View | Model Views > Toolbar**











##### Toolbar Options




The availability of the **Model Views** toolbar options depends on the type of object selected. The options are, from left to right:

Icon	Action	Shortcut	See also
	Click on this icon to display the appropriate <b>Properties</b> dialog for the selected item.	Double-click on the item, or	



Icon	Action	Shortcut	See also
		<b>Enter</b>	
	Click on this icon to locate the selected object in any diagrams in which it has been used in the model, and either: <ul style="list-style-type: none"> <li>• Display the single diagram with the object highlighted or</li> <li>• List the several diagrams in which the object has been located</li> </ul>		
	Click on this icon to locate and highlight the selected object in the Project Browser.		
	Click on this icon to create a new <i>Model View</i> root node, and to display the New Model View dialog in which you enter the root node name.		
	Click on this icon to create a new <i>Views</i> folder in the currently-selected root node.		
	Click on this icon to create a new <i>Favorites</i> folder in the currently-selected <i>Views</i> folder.		
	Click on this icon to create a new Slideshow folder in the currently-selected <i>Views</i> folder.		<a href="#">Diagram Slideshow</a> <sup>[695]</sup>
	Click on this icon to create a new <i>View</i> in the currently-selected <i>Views</i> folder, and to display the Create New View dialog to define the search that populates the View.		<a href="#">Model Views Operations</a> <sup>[693]</sup>
	Click on this icon to refresh the selected <i>Model Views</i> root node, folder, <i>View</i> or <i>Favorites</i> ; for a <i>View</i> , this runs the Model Search defined in the View properties.		
	Click on this icon to move the currently-selected object up or down <b>within its type</b> ; you cannot move - for example - a Package below a diagram, or a View above a Favorites folder.		
	Click on this icon to delete the selected object and - if appropriate - its contents. A prompt displays to confirm deletion.  You cannot delete the original <i>Model Views</i> , <i>My Views</i> or <i>Recent Team Reviews</i> root nodes, or any technology-defined Views.		

Icon	Action	Shortcut	See also
	Click on this icon to display Help on the Model Views.		

#### 4.1.4.2 Model Views Context Menus

**Access** **View | Model Views: Right-click on object**

##### Options

The Model Views window context menus display different options, depending on which level of the Model Views hierarchy you right-click on.

Menu Option	Action	Shortcut	See also
<b>Properties</b>	<p>(Not for the <i>My Views</i>, <i>Recent Team Reviews</i>, initial <i>Model Views</i> or <i>Technology-defined</i> root nodes.)</p> <p>Display the appropriate Properties dialog for the selected object.</p> <p>You can edit any of the properties, if required; changes to objects populated from the model are reflected in all other views (Properties window, diagrams, reports) of that object.</p> <p>The <b>Properties</b> option for the <i>Documents and Comments</i> folder displays the Recent Post Options dialog, which enables you to specify the number of days back from which to extract postings from the Team Review, and the number of postings to list.</p> <p>The <b>Properties</b> option for a <i>Slideshow</i> folder displays the Slideshow Properties dialog, which enables you to automate the slide show and set the number of seconds for which each diagram is displayed.</p> <p>If you deselect the <b>Enable</b> checkbox, you must press <b>Spacebar</b> to display each diagram.</p>		<a href="#">Diagram Slide Show</a> <sup>[695]</sup>
<b>New Views Folder</b>	<p>(Root node only.)</p> <p>Display a prompt for the <i>Views folder</i> name and create the folder in the selected root node.</p>		
<b>Import Views From XML</b>	<p>(Root node only.)</p> <p>Prompt for the XML file location and create a new <i>Model Views</i> node to hold the imported Views.</p>		
<b>Export to XML (Views Only)</b>	<p>(Root node only.)</p>		<a href="#">Model Views Operations</a> <sup>[693]</sup>

Menu Option	Action	Shortcut	See also
	Prompt for a file path and name, and copy all Views under the selected root node to an XML file at that location.		
<b>Remove Model View</b>	(Not for the <i>My Views</i> , initial <i>Model Views</i> or <i>Technology-defined</i> root nodes.)  Display a prompt to delete the selected user-defined Model View and, if confirmed, delete the root node and all contents.		
<b>New Search Folder</b>	( <i>View</i> folder only.)  Display the Create New View dialog (similar to the View Properties dialog) for you to define the search that populates the View.		<a href="#">Model Views Operations</a> <sup>[693]</sup>
<b>New Favorites Folder</b>	( <i>View</i> folder only.)  Display the Create a new favorites based folder dialog, which prompts for the folder name.		
<b>New Slideshow</b>	( <i>Model View</i> , <i>View</i> folder only.)  Display the Create a new slideshow dialog, in which you type the name of the slide show.  You must use the Properties dialog to define the properties of the slide show.		<a href="#">Diagram Slide Show</a> <sup>[695]</sup>
<b>Open Search</b>	( <i>View</i> only.)  Display the Model Search tab in the main work area, listing the full results of the search and giving access to all the facilities of the Model Search.	<b>Shift+Space</b>	<a href="#">Customizing the Search View</a> <sup>[708]</sup>
<b>Refresh</b>	Refresh the search and open the View or Documents and Comments repository to show the elements or documents retrieved by the search.	<b>Space</b>	
<b>Double Click Opens Search</b>	Enable you to perform the Open Search function by double-clicking on the View.  If you deselect this option, double-click refreshes the search and opens the View to show the elements retrieved by the search.		
<b>Open Team Review</b>	Open the Project Team Review at the top level (Category).		

Menu Option	Action	Shortcut	See also
<b>Edit Connections</b>	Display the Team Review Server Connections dialog, to select which review to open.		<a href="#">Team Review Connections</a> <sup>[363]</sup>
<b>Remove Folder</b>	(View folder only.) Display a prompt to delete the selected Views folder and, if confirmed, delete the folder and all contents.		
<b>Remove View</b>	Display a prompt to delete the selected View and, if confirmed, delete the View and all contents.		
<b>Remove Favorites</b>	Display a prompt to delete the selected <i>Favorites</i> folder and, if confirmed, delete the folder and all contents.		
<b>Remove Slideshow</b>	Display a prompt to confirm deletion of the selected slide show and, if confirmed, delete the slide show and all its diagrams.		
<b>Run Slideshow</b>	Run the slide show in the Diagram View.		<a href="#">Diagram Slide Show</a> <sup>[695]</sup>
<b>Run Slideshow Full Screen</b>	Run the slide show in full screen mode, so that the slide show fills the whole screen.		
<b>Stop Slideshow</b>	Cancel execution of the slide show running in the Diagram View.	<b>Esc</b>	
<b>In Project Browser</b>	(An element, diagram or Package object only.) Highlight the selected item in the Project Browser.		
<b>In Diagrams</b>	(An element or child Package object only.) Locate the selected object in any diagrams in which it has been used in the model, and either: <ul style="list-style-type: none"> <li>• Display the single diagram with the object highlighted or</li> <li>• List the several diagrams in which the object has been located</li> </ul>		
<b>Remove Linked Item</b>	(An element, diagram or Package object only.) Display a prompt to delete the selected object and, if confirmed, remove the object from the folder.  This has no effect on the object in the Project Browser or		


Menu Option	Action	Shortcut	See also
	any diagrams. You would not delete an object in a View, as it is replaced the next time the View is refreshed.		
<b>Help</b>	Display Help on Model Views.		

#### 4.1.4.3 Model Views Operations



From the Model Views window you can perform a number of operations to define the information you extract from the model and to further process that information, as described here.

Access **View | Model Views**

##### Operations

Operation	Detail	See also
<b>Define View Search</b>	<p>When you:</p> <ul style="list-style-type: none"> <li>Create a View, the Create New View dialog displays</li> <li>Display the View properties, the View Properties dialog displays</li> </ul> <p>These dialogs are identical.</p> <p>In the <b>Name</b> field, type a name for the View.</p> <p>In the <b>Search</b> field, either:</p> <ul style="list-style-type: none"> <li>Click on the drop-down arrow and select an existing search from the lists, or</li> <li>Click on the ( ... ) (Browse) button to display the Manage Searches dialog, edit an existing search or define a new one, then <b>Close</b> the dialog and select that search name in the <b>Search</b> field</li> </ul> <p>A custom SQL search statement should return the <i>guid</i> and <i>type</i> of the object found so that the system can search for the selected item in the Project Browser.</p> <pre>SELECT ea_guid AS CLASSGUID, Object_Type AS CLASSTYPE, Name FROM t_object</pre> <p>If required, in the <b>Search Term</b> field type a specific value to search for.</p> <p>If you are working with the Corporate or extended versions of Enterprise Architect, and you want the search to run automatically and refresh the results, select the <b>Refresh this search</b> checkbox.</p> <p>The <b>Frequency</b> field has three sections, for hours, minutes and seconds; click on the appropriate section and use the up and down arrows at the end of the field to set the interval for refreshing the search results.</p>	<p><a href="#">Create &amp; Modify Searches</a> </p>

Operation	Detail	See also
	<p>You can also set the refresh to display a pop-up notification if the search results change. To do this, select the <b>Notify me when new results found</b> checkbox.</p> <p>Click on the <b>OK</b> button; the View is created (or updated) in a collapsed state.</p> <p>When you expand the View, the search executes and populates the View.</p>	
<b>Display Recent Postings</b>	<p>If a new document is added to the Team Review, or an existing document is updated, that document is also added automatically to the <i>Recent Team Reviews</i> / <i>&lt;Connection&gt; / Documents and Comments</i> folder in the Model View.</p> <p>To open a document, double-click on the entry; the Team Review tab displays, showing the selected message.</p> <p>You can control how many documents are displayed, and for what period of time, using the Model Views context menu for the <i>Recent Team Reviews</i> folder.</p>	<a href="#">Model Views</a> [686] <a href="#">Model Views Context Menus</a> [690]
<b>Move Objects Into Favorites</b>	<p>Drag any required Package, diagram or element from the Project Browser into the required <i>Favorites</i> folder.</p>	
<b>Move Objects Between Views</b>	<p>Views and <i>Favorites</i> folders are fixed in the <i>Views</i> folder in which you create them, and you cannot move them.</p> <p>However, you can:</p> <ul style="list-style-type: none"> <li>• Copy (by dragging) objects from any View into any <i>Favorites</i> folder, and</li> <li>• Move (by dragging) objects between any two <i>Favorites</i> folders</li> </ul>	
<b>Use Objects From Model Views</b>	<p>To make use of an element, diagram or Package held in any View or <i>Favorites</i> folder, click on the item and drag it into a diagram or a Team Review posting</p> <p>The item behaves in the same way as if you dragged it from the Project Browser.</p>	<a href="#">Team Review</a> [343]
<b>Create Documents containing Objects</b>	<p>A very useful feature of the Model Views facility is the ability to drag separate items from the Views into an existing <b>Linked Document</b>.</p> <p>To do this you create your Linked Document, generate your Model View, and open the document and Model View as docked windows. You then drag the items - individually or as a multiple selection - from the View into the document, selecting one of two options:</p> <ul style="list-style-type: none"> <li>• <b>Insert as Hyperlink(s)</b> and</li> <li>• <b>Render as RTF</b></li> </ul> <p>If you select <b>Insert as Hyperlink(s)</b>, a hyperlink is embedded in the document for each dragged object, linking to the object in the Project</p>	<a href="#">Create Document Artifact</a> [1046] <a href="#">Create Linked Document on an Element</a> [1047]

Operation	Detail	See also
	<p>Browser.</p> <p>If you select <b>Render as RTF</b>, a prompt displays for you to select the RTF template (if any) from which to generate a small report on the dragged objects. You can select different templates for different objects. However, there needs to be compatibility between the selected object type (such as <i>Diagram</i>) and the RTF template used (such as <i>Diagram report</i>).</p> <p>You can have hyperlinked objects and RTF-rendered objects in the same document.</p>	
<b>Export/ Import Views</b>	<p>You export Views to create an XML file that you can:</p> <ul style="list-style-type: none"> <li>• Import into another model as a user-created Model View or</li> <li>• Call from an MDG Technology Selection (MTS) file to access the Technology-defined View provided by the active MDG Technology</li> </ul> <p>The export and import functions are available from the Model Views root-node context menus.</p> <p>When you use the export function, it acts on the complete set of View folders in the selected <i>My Views</i> root node, <i>Model Views</i> root node, or user-generated root node; you cannot export individual Views, nor can you export Favorites folders.</p> <p>The function displays the Save As dialog, on which you browse for the directory location for the exported XML file, and specify the file name.</p> <p>When you use the import function, it displays the Select Import Filename dialog on which you browse for the directory and XML file you want to import; the import creates a new Model View folder with the same name as the copied root node.</p>	<a href="#">Model Views Context Menus</a> 
<b>Set Up a Technology- Defined View</b>	<p>To set up the Technology-defined View for an MDG Technology, you:</p> <ol style="list-style-type: none"> <li>1. Create a user-generated Model View in Enterprise Architect while using the technology.</li> <li>2. Populate it with the required View folders and Views.</li> <li>3. Export the Views from that Model View as an XML file to an appropriate location.</li> <li>4. Create a call to the file from the technology's MTS file.</li> </ol> <p>Thereafter, any model for which the MDG Technology is active automatically displays those Views in a Technology-defined View.</p>	<a href="#">Create MDG Technologies</a> 

#### 4.1.4.4 Diagram Slide Show

The Slide show facility in the Model Views window enables you to create and run screen-based presentations of sets of diagrams. When you run a slide show, it displays the diagrams in the folder in the sequence in which they are listed in the folder. After each diagram has been displayed, it is closed. You can use this facility to:

- Create a slide show as a folder within a Views folder under a Model View node
- Run a slide show manually

- Run a slide show automatically with each diagram displaying for a period that you define
- Display a slide show within the Enterprise Architect work area
- Display a slide show in full screen mode
- Delete a slide show

**Access**   **View | Model Views**

**Create a slide show**



Step	Action	See also
1	Under a Model Views node, click on the Views folder to contain the slide show.	
2	<p>Either:</p> <ul style="list-style-type: none"> <li>• Right-click on the Views folder and select the <b>New Slideshow</b> option or</li> <li>• Click on the <b>New Slideshow folder</b> icon in the Model Views toolbar</li> </ul> <p>The Create a New Slideshow Folder dialog displays.</p>	
3	In the <b>Name</b> field, type the name of the slide show.	
4	<p>Click on the <b>OK</b> button.</p> <p>The new slide show folder is added to the selected View folder.</p>	
5	Dock the Project Browser window separately from the Model Views window.	<a href="#">Dock Windows</a> <sup>[128]</sup>
6	<p>Drag the diagrams to be displayed as part of the slide show from the Project Browser into the new slide show folder.</p> <p>Organize the diagrams in the order in which they are to be displayed, using the up/down green arrows in the Model Views toolbar.</p> <p>To remove a diagram that is not required, right-click on it in the slide show folder and select the <b>Remove Linked Item</b> context menu option.</p> <p>The diagram items in the slide show folder are links to the diagrams in the Project Browser, not copies of the diagrams or the diagrams themselves; any actions you take in the slide show have no impact on the original diagrams.</p>	<a href="#">Model Views Toolbar</a> <sup>[688]</sup>
7	Close the folder.	









**Define how the slide show is to operate - automated or manual**

Step	Action	See also
1	Right-click on the slide show folder and select the <b>Properties</b> context menu option. The Slideshow Properties dialog displays.	
2	To automate the slide show, select the <b>Enable</b> checkbox. To run the slide show manually, deselect the checkbox.	
3	If you are automating the slide show, in the <b>Time</b> field type the number of seconds that each diagram is to remain on display.	
4	Click on the <b>OK</b> button.	

**Run a slide show of diagrams, in either Diagram View or full screen**

Step	Action	See also
1	Right-click on the slide show folder and select either the: <ul style="list-style-type: none"> <li>• <b>Run Slideshow</b> option, to run the slideshow in the Diagram View</li> <li>• <b>Run Slideshow Fullscreen</b> option to run the slideshow using the full screen</li> </ul> The first diagram in the slide show displays.	
2	If you have set up the slide show to run automatically, you can leave it to display the diagrams as defined. You can also moderate the slide show using manual commands.	
3	If you are controlling the slide show manually, right-click on a slide. The following toolbar displays: 	
4	Control the slide show using the toolbar icons and other aids, as follows: <ul style="list-style-type: none"> <li>• Display the next slide - , ( <b>Spacebar</b> ) or ( <b>↑</b> )</li> </ul>	

Step	Action	See also
	<ul style="list-style-type: none"> <li>• Display the previous slide - , ( ← )</li> <li>• Display the first slide - , ( ↑ )</li> <li>• Display the final slide - , ( ↓ )</li> <li>• Pause the slide show - </li> <li>• Resume the slide show - </li> <li>• Stop the slide show -  or, in Diagram View, right-click on the slide show folder and select the <b>Stop Slideshow</b> option</li> </ul>	

#### Delete a slide show

Step	Action	See also
1	Right-click on the slide show folder and select the <b>Remove Slideshow</b> context menu option. Enterprise Architect prompts you to confirm the deletion.	
2	Click on the <b>Yes</b> button. The folder and its list of links to diagrams in the model is removed.	

### 4.1.5 The Pan & Zoom Window

The Pan & Zoom window provides a 'birds-eye' view of diagrams. It enables you to navigate quickly around large diagrams.

**Access** **Diagram | Pan & Zoom** (Ctrl+Shift+N)

#### Reference

Topic	Detail	See also
<b>View</b>	<p>The window shows a reduced image of the whole diagram.</p> <p>The shaded box represents the viewed area, displayed on the open diagram.</p>	

Topic	Detail	See also
<b>Toolbar</b>	<p>The toolbar provides the following functions (in order):</p> <ul style="list-style-type: none"><li>• Zoom In</li><li>• Zoom Out</li><li>• Zoom to fit diagram</li><li>• Zoom to fit page</li><li>• Zoom to 100%</li><li>• Zoom Slider</li></ul>	
<b>Use</b>	<p>Open the required diagram.</p> <p>Move the cursor onto the shaded box inside the Pan &amp; Zoom window.</p> <p>Hold down the mouse button as you move the cursor to move the shaded box over the open diagram.</p> <p>To zoom, use either the Zoom Slider or the buttons located on the tool bar.</p>	

## 4.2 Search: Finding Information



Searching involves using ad hoc search terms and various search types to locate elements and information within your model, regardless of location or context.

The result sets can be used to manage elements, to create documentation or to find the location of elements within the complete model structure.

### Topics

Topic	Link
Model Search - how to use the searching capability within Enterprise Architect to find and display various kinds of information	<a href="#">Model Search</a> <sup>[700]</sup> <a href="#">Model Search Context Menu</a> <sup>[703]</sup> <a href="#">Model Search Toolbar</a> <sup>[705]</sup> <a href="#">Pre-defined Searches</a> <sup>[706]</sup> <a href="#">Customizing the View</a> <sup>[708]</sup>
Create and Modify Searches - how to modify and work with search definitions	<a href="#">Create &amp; Modify Searches</a> <sup>[709]</sup>
Adding Filters - further modifications to searches	<a href="#">Add Filters</a> <sup>[716]</sup>
Diagram Filters- how to suppress or highlight information in a diagram using customizable filtering mechanisms based on element properties such as status and complexity	<a href="#">Diagram Filters</a> <sup>[718]</sup>

### 4.2.1 Model Search

The Model Search allows you to quickly navigate and find elements within your model. It lists each element that meets the search criteria you specify within the search terms and search type. Many different kinds of searches are already built into Enterprise Architect, and it is possible to create very detailed and specific searches of your own. It is even possible to export and share your favorite searches.

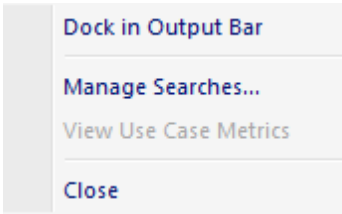

**Access** **Edit | Find in Project** ( **Ctrl+Alt+A** ) or ( **Ctrl+F** )

#### Use to

- Search for elements within your model based on various criteria
- Generate targeted reports based on your result set

- Manage result sets - for example set properties and delete elements

Field	Usage	See also
<b>Search Term</b>	Type in the word, phrase or characters to search on.  Search filters enable you to perform customized searches on a Search Term in order to locate model elements having specific characteristics.	
<b>Search</b>	The Model Search window provides a set of <i>built in</i> searches for your convenience; the Search drop-down list provides several pre-defined searches including. <ul style="list-style-type: none"> <li>• Simple</li> <li>• Extended</li> <li>• Find Orphans</li> <li>• Recently Modified Elements</li> </ul> <p>In the drop-down list, the built in searches are followed by searches that:</p> <ul style="list-style-type: none"> <li>• You have created yourself</li> <li>• Your organization has created for use within the project</li> <li>• Have been created as part of workflow script</li> <li>• Have been created in the Enterprise Architect Scripting facility</li> </ul>	<a href="#">Pre-defined searches</a> <sup>[706]</sup> <a href="#">Functions - Create a Search With User Tasks</a> <sup>[370]</sup> <a href="#">Workflow Data Structures You Fill</a> <sup>[372]</sup> <a href="#">Script Group Properties</a> <sup>[2794]</sup>
<b>Run</b>	Run the search using the entered text and selected Search.	
<b>Options</b>	<p>The <b>Options</b> button displays the <b>Search Options</b> submenu, which enables you to display the search results as a tab of the Output window rather than in the Model Search View.</p> <p>An advantage of moving the search results to the Output window is that you can select items from the search results and drag them onto a diagram, which you cannot do when the results are in the Model Search View.</p> <p>If you select the <b>Dock in Output Bar</b> menu option, when you next display the menu this option becomes <b>Dock in Main View</b>.</p> <p>The <b>Search Options</b> submenu also provides the means of performing advanced searches on your project, and displaying project metrics.</p>	<a href="#">The System Output Window</a> <sup>[169]</sup>  <a href="#">Advanced Search Options</a> <sup>[715]</sup> <a href="#">Estimating Project Size</a> <sup>[588]</sup>






		
<b>Builder</b>	Show or hide the Search Builder facility.	<a href="#">Create Search Definitions</a> 
<b>Result List</b>	Display the results of your search.	

**Keyboard Options:**

- Display the properties of an element (**Double Click or Enter**)
- Drag an item(s) onto a diagram (**Ctrl+drag a selected item**)
- Toggle between the Notes and the Model Search window (**Ctrl+Shift+Enter**)
- Open the Linked Document window for the selected Element (**Ctrl+Alt+D**)
- Select All Items (**Ctrl+Alt+A**)

**Notes**

- You can access the Model Search facilities and perform specific searches indirectly, from Add-Ins, from MDG Technologies, from a hyperlink and from a shortcut to access your model; this entails setting up a search profile either in the appropriate tool, or as an XML file accessed by the tool

Searching from	See
An MDG Technology Selection (MT ) File (using an exported search definition)	<a href="#">Working With MTS Files</a>  <a href="#">Create &amp; Modify Searches</a> 
A Login Shortcut	<a href="#">Project Shortcuts</a> 
An Add-In	<a href="#">Add-In Search</a> 
A Hyperlink	<a href="#">Hyperlinks</a> 

- Running a custom or diagram based search disables some context menu options

#### 4.2.1.1 Model Search Context Menu

You can select elements or diagrams in the Model Search and perform various operations on them, as well as simply dragging the item into a Team Review document or Linked Document. You can:

- Generate reports from search results
- Print search results
- Copy and work with search results
- Manage elements returned by a search

**Access**   **Edit | Find in Project ( Ctrl+Alt+A ) Results pane, right-click on an element**

##### Options

Menu Option	Action	Shortcut	See also
<b>Scripts</b>	<p>Displays a list of any Model Search scripts you have created, which can be executed on the selected item.</p> <p>This option displays only if:</p> <ul style="list-style-type: none"> <li>• You have created Model Search scripts in a Model Search Script Group folder, and</li> <li>• The item results from a successful Query search, or</li> <li>• The item results from a successful SQL search that includes CLASSGUID and CLASSTYPE</li> </ul>		<a href="#">Script Group Properties</a> <sup>[2794]</sup>
<b>Properties</b>	Display the Properties dialog for the element.		
<b>Copy Selected to Clipboard</b>	Copy the selected item to the MS Windows clipboard so that it can be pasted to a document, spreadsheet or email.		
<b>Documentation</b>	<p>Generate a document report; you have two options:</p> <ul style="list-style-type: none"> <li>• Generate a separate report on each selected object in the Model Search</li> <li>• Generate one report on all selected objects</li> </ul> <p>In either case, the Generate Documentation dialog displays.</p> <p>If you generate the report using a custom SQL search, the SQL must include <code>ea_guid AS CLASSGUID</code> and the object type.</p>		<a href="#">Generate Documentation</a> <sup>[2644]</sup>  <a href="#">Create Search Definitions</a> <sup>[711]</sup>

<b>Create Linked Document</b> <b>(Edit Linked Document)</b>	Create (or edit) a linked document (Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions).	<b>Ctrl+Alt+D</b>	<a href="#">Create Linked Document on an Element</a> <sup>[1047]</sup> <a href="#">Edit Linked Documents</a> <sup>[1048]</sup> <a href="#">Linked Documents</a> <sup>[1044]</sup>
<b>Delete Linked Document</b>	Delete an existing linked document. (This option only displays if the element has a linked document.)		
<b>Print</b>	Print out the filtered results.		
<b>Clear Results</b>	Clear the search results from the Model Search.		
<b>Find in Diagrams</b>	Display the diagram that uses the element or, if the element is used in multiple diagrams, display a list of diagrams to choose from.	<b>Ctrl+U</b>	
<b>Find in Project Browser</b>	Highlight the element in the Project Browser.	<b>Alt+G</b>	
<b>Bookmark Selected</b>	Bookmark the element.		
<b>Delete Selected</b>	Delete the selected element from the Model Search.		
<b>Close</b>	Close the Model Search.		
<b>Help</b>	Display this Help topic on the Model Search.		

#### Add Items To Linked Document

A very useful feature of the Model Views facility is the ability to drag separate items from the Views into an existing **Linked Document**.

To do this you create your Linked Document, generate your Model View, and open the document and Model View as docked windows. You then drag the items - individually or as a multiple selection - from the View into the document, selecting one of two options:

- **Insert as Hyperlink(s)** and



- **Render as RTF**

If you select **Insert as Hyperlink(s)**, a hyperlink is embedded in the document for each dragged object, linking to the object in the Project Browser.

If you select **Render as RTF**, a prompt displays for you to select the RTF template (if any) from which to generate a small report on the dragged objects. You can select different templates for different objects. However, there needs to be compatibility between the selected object type (such as *Diagram*) and the RTF template used (such as *Diagram report*).

You can have hyperlinked objects and RTF-rendered objects in the same document.

#### Learn more

- [Add a New Post](#)<sup>[353]</sup>

### 4.2.1.2 Model Search Toolbar

The Model Search toolbar displays at the top of the Model Search window, enabling you to operate on the results of your model search.

**Access** **Edit | Find in Project ( Ctrl+Alt+A )**

#### Use to

- Edit the text of notes for an item
- Delete selected items from the model
- Print the search results
- Generate and print a document report on selected items
- Hide or abbreviate notes text in the output
- Hide or show the filter bar under the column headings
- Display Help text

#### Reference



Option	Action	Shortcut	See also
<b>Edit Notes</b>	For the selected item, open the Notes window (if it is not already open) so that you can edit the text of the notes.	<b>Ctrl+Spacebar</b>	<a href="#">Notes</a> <sup>[1142]</sup>
<b>Delete</b>	For a selected item or group of items, delete them from the model; refresh the project to check that the items have been deleted.	<b>Ctrl+D</b>	
<b>Print</b>	Print the complete set of search results		

<b>Rich Text Report</b>	For a selected item or group of items, generate and print a document report.		<a href="#">Generate Documentation</a> [2644]
<b>View Notes</b>	Display a short menu that enables you to select whether, for all items, to: <ul style="list-style-type: none"> <li>• Hide any Notes text from display in the search results</li> <li>• Display the first few words of the Notes text in the search results</li> <li>• Display the full Notes text in the search results</li> </ul>		
<b>Toggle Filter Bar</b>	Hide or reveal the filter bar underneath the column headings.		<a href="#">Customizing the Search View</a> [708]
<b>Help</b>	Display the Enterprise Architect Help, starting with the Model Search Help topic.		

#### 4.2.1.3 Pre-defined Searches

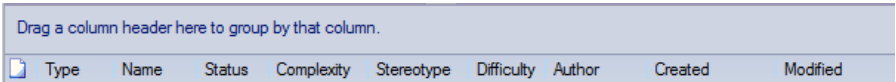
The following pre-defined searches are provided with Enterprise Architect, and are listed in the *Built-In* category in the **Search** drop-down list.


Search	Description	See also
<b>Simple</b>	Searches the <b>Name</b> , <b>Alias</b> and <b>Notes</b> fields of all elements for the given search term.	
<b>Extended</b>	Searches many additional fields relating to the element, including <b>Attributes</b> , <b>Operations</b> , <b>Tagged Values</b> and <b>Test Cases</b> .	
<b>Element Name</b>	Searches for an exact match against the element name, alias, method or operation; the default search in the <b>Search in Model</b> menu option in the Code Editor.	<a href="#">Code Editor Context Menu</a> [2152]
<b>Attribute Details</b>	Searches for elements with attributes relating to the search term, including Tagged Values, constraints, and common attribute data fields.	
<b>Find Orphans</b>	Searches for orphaned elements throughout the model, with the ability to filter on common element fields using a search term.  An 'orphaned' element is an element that does not appear on any diagram in the model.	

<b>Failed Internal Tests</b>	Searches for elements containing internal test cases where the search term is in any common Test Case field and the <b>Status</b> value is <i>Fail</i> .	
<b>Method Details</b>	Searches for elements with operations and methods relating to the search term, including Tagged Values, constraints and common operation and method data fields.	
<b>Responsibility</b>	Searches for elements with internal responsibilities/requirements where the search term relates to any common <b>Responsibility/Requirement</b> field.	
<b>Resources</b>	Searches for elements with assigned resources where the search term relates to any common <b>Resource</b> field.	
<b>Requirements</b>	Searches for Requirement element types where the search term relates to any common element field.	
<b>Find Bookmarked Elements</b>	Searches for elements that have been bookmarked, anywhere in the project.	
<b>Recently Modified Elements</b>	<p>Searches for elements that have been recently modified, anywhere in the project.</p> <p>The search term relates to any common element field.</p> <p>The default is to show elements modified in the last three days, but you can set an alternative interval by typing the appropriate number of days in the <b>Search Term</b> field.</p>	
<b>Recently Modified Diagrams</b>	<p>Searches for diagrams that have been recently modified, anywhere in the project.</p> <p>The search term relates to any common diagram properties field.</p> <p>The default is to show diagrams modified in the last three days, but you can set an alternative interval by typing the appropriate number of days in the <b>Search Term</b> field.</p>	
<b>My Checked Out Packages</b>	Searches for packages that are marked as checked out by the currently-logged in user.	

#### 4.2.1.4 Customizing the Search View

This topic explains how to customize the display of search results after they are generated.

Topic	Detail	See also
<b>Sort and Select</b>	<p>In the Model Search you can:</p> <ul style="list-style-type: none"> <li>Sort the items by any column value in ascending or descending order, by clicking on the column header</li> <li>Display element or diagram properties, by double-clicking on the item</li> <li>Select: <ul style="list-style-type: none"> <li>An element or diagram by clicking on it</li> <li>Several individual elements or diagrams by holding ( <b>Ctrl</b> ) as you click on them</li> <li>A range of elements or diagrams by holding ( <b>Shift</b> ) as you click on the first and last in the range</li> <li>All elements or diagrams in the list by pressing ( <b>Ctrl+A</b> )</li> </ul> </li> </ul>	
<b>The View Header</b>	 <p>The View Header defines the columns of information that are presented by the Model Search, and the order in which data items are presented.</p> <p>To review the column headings:</p> <ul style="list-style-type: none"> <li>Right-click on the header to display the <b>Field Chooser</b> context menu option; the Field Chooser dialog displays</li> </ul> <p>This enables you to add columns to or remove columns from the output; between them, the View Header and Field Chooser dialog show the full range of column headers available.</p> <p>To add a column heading to the View Header:</p> <ul style="list-style-type: none"> <li>Drag it from the Field Chooser dialog onto the header to the position you want the column of data to display</li> <li>When you have selected the column headings you require, click on the red cross in the top right corner of the Field Chooser dialog to close it</li> </ul> <p>To remove a column from the output:</p> <ul style="list-style-type: none"> <li>Drag the column heading downwards out of the View Header</li> </ul> <p>To change the sequence of columns:</p> <ul style="list-style-type: none"> <li>Drag the required column header left or right, as appropriate</li> </ul>	
<b>Value Grouping</b>	You can organize the reported data hierarchically according to the value of one or more of the column categories.	

Topic	Detail	See also
	<p>You might organize the data by <b>Type</b>, and within each Type by <b>Name</b>; if you then click on any of the other column headings, the data within this grouping is further sorted with the values of the selected column (for example, <b>Created</b>) in ascending or descending order.</p> <p>To set up the value grouping, drag the column heading representing the primary grouping (such as <b>Author</b>) onto the <b>Drag a column header here to group by that column</b> field.</p> <p>Now drag the column heading for the next level of grouping (such as <b>Status</b>) to the right of the first heading; the two heading titles display as connected blocks, as shown below:</p>  <p>You can, if required, add further levels of grouping by dragging other column headings onto the hierarchy (such as <b>Created</b>), and restructure the order by dragging existing or additional headings into the level you want them to hold.</p> <p>For example, you could make <b>Type</b> the secondary grouping by dragging it to the right of <b>Author</b>, or drop <b>Status</b> between <b>Author</b> and <b>Type</b>.</p> <p>To remove a grouping level, drag the appropriate column heading out of the sequence and below the View header; any subordinate groupings move up a level.</p>	
<b>Filter Bar</b>	<p>The Filter Bar displays underneath the column headings; you can hide or show it using the <b>Toggle Filter Bar</b> icon in the toolbar.</p> <p>The Filter Bar consists of a row of fields, one field per column; as you type a value into one of these fields, the Model Search results are automatically filtered to show entries with a value in the corresponding column that contain the value in the field.</p> <p>For example:</p> <ul style="list-style-type: none"> <li>If you type the letter <b>F</b> in the Filter Bar field under the <b>Author</b> column, the results immediately reduce to entries for which the letter <b>F</b> occurs anywhere in the name (<b>F</b>red, <b>F</b>itzgibbon, <b>R</b>ufus, <b>W</b>olf)</li> <li>If you then go on to type <b>Fi</b>, the results immediately reduce again to entries for which the combination <b>fi</b> appears in the name (<b>F</b>itzgibbon, <b>A</b>mal<b>f</b>i)</li> </ul>	

#### 4.2.1.5 Create & Modify Searches

You provide search filters and create new search definitions using the Search Builder tab, displayed underneath the **Search Term** and **Search** fields on the Find in Model view.

**Access** **Edit | Find in Project (Ctrl+Alt+A): Builder**





##### Use to

- Create new search definitions and edit them

**Search Builder Toolbar:**

The search builder toolbar enables you to configure the system-provided searches, and to create and edit your own searches

The toolbar icons, from left to right, provide access to the following functions:

Option	Usage	See also
<b>New Search</b>	Create a new search definition, with new search criteria.	<a href="#">Create Search Definitions</a> 
<b>Save Search</b>	Save a modified or new search.	
<b>Copy Search</b>	Copy the existing search selected in the <b>Search</b> field, to modify. You can copy: <ul style="list-style-type: none"> <li>Searches you or other users have created</li> <li>Predefined searches and edit the copies, but you cannot edit the predefined searches themselves</li> </ul>	
<b>Restore Default</b>	Restore any changed parameters to the default settings and format.	
<b>Delete Search</b>	Delete the search definition from the <b>Search</b> drop-down list.	
<b>Add Filter</b>	Add a new set of parameters to filter the search on.	<a href="#">Add Filters</a> 
<b>Remove Filter</b>	Delete the selected filter from the search.	
<b>Export Search</b>	Display a selection box that enables you to select searches to export to an external directory as an XML Search file.	
<b>Import Search</b>	Display the Windows Directory Explorer Open dialog to enable you to import searches as XML Search files from an external directory.	
<b>Search Options</b>	Display the Advanced Options dialog, to define where a query-builder search should operate and how the search should match filters.	<a href="#">Advanced Search Options</a>  <a href="#">Create Search Definitions</a> 

**Search Builder**

You use the main body of the Search Builder tab to configure the element search filters that are contained in the selected search.

The Search Builder tab defaults to a **Simple** search, which changes as you select to create a new search or select the search to modify.


Column	Usage	See also
<b>Search In</b>	Select the type and name of each element feature (the element field name) to search on.	
<b>Condition</b>	Select the condition of the search parameter (the condition placed on the field value).  The available options are <b>Contains</b> , <b>Equal To</b> , <b>Not Equals</b> and <b>One Of</b> .	<a href="#">Fields and Conditions</a> <sup>718</sup>
<b>Look for</b>	Specify the search term (the actual value or delimiting value) to perform the conditional search on.  This value can pertain to the selected element type; for example, the value could be a date for <i>DateCreated</i> or a text value for other fields.  The search term can contain multiple values, separated by commas.	
<b>Required</b>	Indicate whether the field is mandatory - that is, whether the search results must include elements with your search term in that field.  The following provides example of using the <b>Required</b> checkbox, illustrated by the default <b>Simple</b> search, which checks the <b>Name</b> , <b>Alias</b> and <b>Notes</b> fields of all elements: <ul style="list-style-type: none"><li>• If the <b>Required</b> checkbox is not ticked for any field, then if the search term is found in the <b>Name</b>, <b>Alias</b> OR <b>Notes</b> field for an element, that element is listed in the results</li><li>• If the <b>Required</b> checkbox is ticked for all fields in the search, then the search term must be found in all of those fields before the element is listed in the results; that is: <b>Name AND Alias AND Notes</b></li><li>• If the <b>Required</b> checkbox is ticked for some fields but not others, the search term must be found in all fields for which the checkbox is ticked, and <i>at least one</i> of the fields in which the checkbox is not ticked; that is: <b>Name AND (Alias OR Notes)</b></li></ul> This checkbox helps you to define the filters on a specific <b>feature</b> (such as Element, Diagram or Attribute). You can also set up a filter to check for a defined value in <b>every feature</b> covered by a search (AND filter) or in <b>any feature</b> covered by the search (OR filter).	<a href="#">Advanced Search Options</a> <sup>719</sup>

#### 4.2.1.5.1 Create Search Definitions



If you want to define your own searches, you can do so using the SQL Editor, Query Builder or an Add-In, through the Search Builder tab. User-defined searches are stored in the user application data for the machine being used, and not in the project repository.

[Access](#) [Edit](#) | [Find in Project: Builder](#)

**Create a new search definition**

Step	Action	See also
1	Click on the <b>New Search</b> icon in the toolbar. The Create New Search Query dialog displays.	
2	In the <b>Search Name</b> field, type a name for the search.	
3	Select the radio button for the type of search you are creating: <ul style="list-style-type: none"> <li><b>Query Builder</b> - provides an interface through which you can design your own search</li> <li><b>SQL Editor</b> - for advanced users to directly write SQL <i>SELECT</i> statements</li> <li><b>Add-In Search</b> - where you supply the name of your Add-In and a method (for example <i>MyAddin.RunThisMethod</i>), which is called whenever the search is run; this search can be exported and distributed as a part of your Add-In</li> </ul>	<a href="#">Add-In Search</a> 
4	Click on the <b>OK</b> button, and refer to the next table.	

**Search Types**

Search	Description	See also
<b>Query Builder</b>	Your search name appears as being selected in the <b>Search</b> drop-down field.  Click on the <b>Add Filter</b> toolbar icon to add filters and construct the search.	<a href="#">Add Filters</a> 
<b>SQL Editor</b>	The SQL editor displays within the <b>Search Builder</b> tab, in which you input your <i>SELECT</i> statement.  The SQL editor is based on the common Code Editor, and provides an Intelli-sense autocompletion list populated with the Enterprise Architect repository structure; to display the autocompletion list, press <b>Ctrl+Spacebar</b> .  A simple search might be to locate an object from a table, given a search term that the user enters in the <b>Search Term</b> field  <i>SELECT * FROM t_object WHERE NAME='&lt;Search Term&gt;'</i>  In the <i>WHERE</i> statements you can also use <b>#xxx#</b> macros as string replacers, so that the same search can be used by different people in different environments; these macros include: <ul style="list-style-type: none"> <li><b>#WC#</b> - Gets the appropriate wild card for the current database, so the search can be performed on models on different databases</li> </ul> <i>t_object.Name LIKE '#WC#Test#WC#'</i>	<a href="#">Code Editor Functions</a> 



- **#Author#** - Takes the user name from the **Author** field in the Options dialog General page, so the defined search can be performed on objects created by that user (this value can be manually re-set in the Options dialog)
- **#DB=<DBNAME>#** where **<DBNAME>** can be one of the following:
  - MYSQL
  - JET
  - ORACLE
  - SQLSVR
  - ASA
  - OPENEDGE
  - POSTGRES

Only uses the section of code between two matching **#DB=<DBNAME>#** macros if the current database type matches the specified DBNAME. Can be used where a section of the SQL might require special handling depending upon the current database type.

For example:

```
#DB=ORACLE# t_object.ModifiedDate >= (SYSDATE - INTERVAL
'<Search Term>' DAY) #DB=ORACLE#
```

- **#UserName#** - Gets the name of the person logged into version control

```
t_package.PackageFlags LIKE
'#WC#VCCFG=#WC#CheckedOutTo=#UserName##WC#'
```

(this is from the built in search *My Checked Out Packages*)

- **#Now#** - Inserts the current date plus or minus a specified number of hours or days; the default is days (the date **format** is adjusted to suit the database in use)

```
t_object.ModifiedDate >=#Now <Search Term>#
```

For example:

```
t_object.ModifiedDate >= #Now -4d#    d is days
t_object.ModifiedDate >= #Now -5h#    h is hours
t_object.ModifiedDate >= #Now +3#
t_object.ModifiedDate >= #Now#
```

- **#Package#** - gets the currently-selected Package's package\_ID

```
t_object.Package_ID = #Package#
```

- **#Branch#** - gets the IDs of the child Packages of the currently-selected Package, working recursively down to the lowest level of sub-Package

	<p><i>t_object.Package_ID IN (#Branch#)</i></p> <p>For all functions in which you use a custom SQL statement (such as Document Reporting or Model Views) the statement must return the <b>guid</b> and <b>type</b> of the object found so that the system can search for the selected item in the Project Browser:</p> <pre>SELECT ea_guid AS CLASSGUID, Object_Type AS CLASSTYPE, Name FROM t_object</pre> <p>You can extend the usability of your SQL searches using the aliases <b>CLASSGUID</b> and <b>CLASSTYPE</b>, so that you can display the Properties dialog and icon for elements, connectors, attributes or operations, as well as selecting them in the Project Browser.</p> <p>Some simple examples for using these aliased fields are provided below:</p> <ul style="list-style-type: none"> <li>• <i>SELECT ea_guid AS CLASSGUID, Object_Type AS CLASSTYPE, Name FROM t_object</i></li> <li>• <i>SELECT ea_guid AS CLASSGUID, Connector_Type AS CLASSTYPE, Name FROM t_connector</i></li> <li>• <i>SELECT ea_guid AS CLASSGUID, 'Operation' AS CLASSTYPE, Name FROM t_operation</i></li> <li>• <i>SELECT ea_guid AS CLASSGUID, 'Attribute' AS CLASSTYPE, Name FROM t_attribute</i></li> </ul> <p>You can enable your search users to <b>drag and drop</b> elements from the <b>search results</b> onto a <b>diagram</b>, by including one or other of the following in your search's <i>SELECT</i> statement:</p> <ul style="list-style-type: none"> <li>• <i>(t_object.Object_ID and t_object.Object_Type)</i> or</li> <li>• <i>t_object.ea_guid AS CLASSGUID</i></li> </ul> <p>When you have defined the <b>SELECT</b> statement, click on the <b>Save</b> button to save this search; the search is then available from the <b>Search</b> drop-down list.</p>	
<b>Add-In Search</b>	<p>Type in:</p> <ul style="list-style-type: none"> <li>• The name of your Add-In</li> <li>• A period (full stop) and</li> <li>• The name of the method to be called (for example, <i>MyAddin.RunThisMethod</i>)</li> </ul> <p>Your search is automatically saved and available from the <b>Search</b> drop-down list.</p>	<p><a href="#">Add-In Search</a></p> <p>3019</p>

**Notes**

- When constructing an SQL search, do not switch to the **SQL tab**; this is for pasting in, modifying and running ad-hoc *SELECT* statements, and does not support the use of **#xxx#** macros

**Learn more**

- [Customizing the Search View](#)<sup>[708]</sup>
- [Create & Modify Searches](#)<sup>[709]</sup>

**4.2.1.5.2 Advanced Search Options**

The advanced search filter options help you to apply greater control over how your customized search operates on your model.

You can set values and checkboxes in your search definition to find specific criteria within a **feature**, for example, element properties. However, you can **also** set filters to run the search across the whole model or a specific package, and to locate objects that have a specified value in:

- All** features (for example, the value exists in an element's properties **and** one of its attributes' properties) or
- Any one** of its features (for example, the value exists in an element's properties **or** any of its attributes' properties)

The filters for each feature operate on a database table. In the default Enterprise Architect searches, for example, filters on element properties operate on the *t\_element* table, filters on attribute properties operate on *t\_attribute*, and filters on method parameters operate on *t\_operationparams*.

**Access**   **Edit | Find in Project ( Ctrl+Alt+A ):** **Builder: Search Options (toolbar icon)**

**Use to**

- Define whether your search should operate on the whole model, or on a specific package within the model
- Define whether your search should return items that match the search term in **every** feature or **any** feature defined for that search

**Reference**

Action	Usage	See also
<b>Return matching items for the entire model</b>	Run the search across the entire model (this is the default option). The search checks every package in the model, and returns items that match the search criteria from wherever they are located.	
<b>Return matching items for the selected Package</b>	Run the search on a specific package that you select from the Project Browser.  If you run a search and then select a different package in the Project Browser, the search results do not change until you click on the <b>Run</b>	

Action	Usage	See also
	<p>button again; that is, to search different areas of the project:</p> <ul style="list-style-type: none"> <li>• Click on the first required package in the Project Browser</li> <li>• Click on the <b>Run</b> button</li> <li>• Check the result</li> <li>• Click on another package in the Project Browser</li> <li>• Click on the <b>Run</b> button</li> </ul>	
<b>All Tables</b>	<p>List objects that have a match in <b>every</b> feature table checked in the search.</p> <p>For example, only list elements where both the element <b>and</b> one or more of its attributes have the same stereotype name.</p>	
<b>At Least One Table</b>	<p>List objects that have a match in at least one of the feature tables checked in the search (the default).</p> <p>For example, list elements that either have the specified stereotype name, <b>or</b> have attributes that have the specified stereotype name.</p>	

#### Learn more

- [Create & Modify Searches](#) 

#### 4.2.1.5.3 Add Filters

**Access** **Project | Documentation | Rich Text Format (RTF) Report: Add Filter**  
**Edit | Find in Project ( Ctrl+Alt+A ): Builder: Add Filter (toolbar icon)**

#### Use to

- Add filter criteria to a model search or a document report template

#### Reference

Option	Action	See also
<b>Search On</b>	<p>Select items to build up search filters on any information about an object.</p> <p>The following is a list of what is available, before you have defined a search:</p>	



Option	Action	See also
	<div> <p>                     Element                      Diagram                      Attribute                      Attribute.AttConstraint                      Attribute.AttTagValue                      Change                      Custom Property                      Constraint                      Method                      Method.MethodTagValue                      Method.Parameter                      Method.PostCondition                      Method.PreCondition                      Method.Parameter.ParamTagValue                      File                      Issue                      Defect                      Scenario                      TagValue                      Task                      Test                      Responsibility                      Resource                 </p> </div> <p>If you are adding filters to an existing search, the list contains only items appropriate to the initial filter.</p> <p>For example:</p> <ul style="list-style-type: none"> <li>• If the initial filter is set to search on <b>Element</b>, the drop-down list for subsequent filters on the search will show a long list of element properties that you can filter more specifically on</li> <li>• If the initial filter is set to search on <b>Diagram</b>, the list for any subsequent filters for the search only contains the <b>Diagram</b> option, as there are no other filters that can be applied to a search on diagrams</li> </ul>	
<b>Include</b>	Select each field item to include in your search (select the checkbox).	
<b>Field</b>	Identify the name of the field to search. The list presents items specific to the filter <i>Search On</i> item.	<a href="#">Fields and Conditions</a> <sup>718</sup>
<b>Condition</b>	Specify the condition of the search parameter.	
<b>Value</b>	Type a value pertaining to the selected element field. For example, the value could be a date for <i>DateCreated</i> or a text value for other fields. The search term can contain multiple values separated by commas.	
<b>Required</b>	Select a particular field to generate a result set that <i>must</i> contain your search term in that field.	

Option	Action	See also
<b>Check All</b>	Select all the items to include them in the search definition.	
<b>Uncheck All</b>	Deselect all the items to omit them from the search definition.	
<b>OK</b>	Apply the filter. The fields selected are added to the search definition.	

### Notes

- You can add multiple search definitions as necessary; however, if you select the **Required** field in multiple definitions the search rapidly becomes impractical
- Multiple search definitions are better for 'and/or' searches

### Learn more



- [Create Search Definitions](#) 
- [Create & Modify Searches](#) 
- [Generate Documentation](#) 

#### 4.2.1.5.3.1 Fields and Conditions

When you click on a condition for a particular field, a selection of condition options becomes available. The list of options varies dependent on the type of information required.

For some conditions, the **Value** field contains an ellipsis ( ... ). Click on this to display a selection dialog. This dialog varies dependant on the information required.

### Learn more

- [Create Search Definitions](#) 
- [Add Filters](#) 

## 4.2.2 Diagram Filters

Using **Diagram Filters (Dynamic Visual filters)**, you can modify the display of diagram components so that the relevant items are immediately identified for the reader's attention, without damaging the structure and integrity of the model. The filters operate on elements and connectors, according to properties such as Author, Status, Date Created or Stereotype and, for connectors, whether the connector is hidden or visible.

You can use Diagram Filters singly, in sequence, or in combination to tailor the display of diagrams to show:

- Information for different users, so that - for example - technical staff and stakeholders each have a view that highlights the information pertinent to them
- What elements have been recently developed or changed
- Which part of a model was developed by a particular person
- Which parts of a diagram are at a particular phase, status or version

Examples of ways in which you might apply filtering include:

- Setting up a filter for immediate use on a diagram, and modifying that filter as you review the diagram so that you highlight elements with different values for the same properties - perhaps, by filtering on **Phase**, to compare 'as-is' and 'to-be' elements
- Setting up a filter and leaving it active so that all diagrams you display are automatically filtered the same way
- Setting up a series of filters to use:
  - in one or more sequences to progressively highlight a diminishing set of items, or
  - alternately to highlight contrasting views of the diagram

Filters for elements can also be applied to the Package Browser and Diagram List.

The Diagram Filters context menu (right-click on the body of the window) provides an option for toggling between the filters and simple context filtering of diagrams; if you select the **Context Filtering** option, the Diagram Filters are disabled and the contents of the diagram are grayed-out **except for** a selected element and those elements directly related to it.

#### Access **Diagram | Diagram Filters**

#### Facilities

Facility	Detail	See also
<b>Developing Filters</b>	<p>You create and define as many filters as you require, setting up each filter by defining which element or connector properties to specifically check for and (depending on how you set up the filter parameters, below) whether to include or exclude elements or connectors having particular property values.</p> <p>You cannot combine element and connector properties in one filter, but you can apply an element filter and a connector filter to a diagram at the same time.</p> <p>You can select to:</p> <ul style="list-style-type: none"> <li>• Mute the irrelevant items in <b>Gray Scale</b>, or <b>Fade</b> the display color</li> <li>• <b>Hide</b> the irrelevant items completely, or</li> <li>• <b>Select</b> and highlight (with a hashed line) the items that <b>are</b> relevant</li> </ul>	<a href="#">Work With Diagram Filters</a> [720]
<b>Filter Effect - Gray Scale, Fade or Hide</b>	<p>If you select <b>Gray Scale</b>, <b>Fade</b> or <b>Hide</b>, the action of the filters is to <i>exclude</i> items that <i>do not</i> match the parameters rather than <i>include</i> items that do.</p> <p>If, for example, you selected to filter on element name, looking for elements with the word <i>Class</i> in the name, the filter would apply the following logic:</p> <p style="padding-left: 40px;">Does <b>Name</b> contain string <i>Class</i>? If <b>No</b>, apply effect; if <b>Yes</b>, take <i>no</i> action.</p> <p>The elements you want are therefore what is left on the diagram, rather than what was operated on.</p>	

Facility	Detail	See also
	The filter effect remains in force while you do other work on the diagram, until such time as you disable the filter.	
<b>Filter Effect - Highlight</b>	<p>If you click on <b>Select</b>, the logic is reversed:</p> <p>Does <b>Name</b> contain string <i>Class</i>? If <b>No</b>, take <i>no</i> action. If <b>Yes</b>, apply effect.</p> <p>In this case, the filter effect is not permanent, and clicking off the items deselects them.</p> <p>This effect is excellent for selecting items having the same characteristics across a large diagram, to be processed in a single task; instead of having to locate the items and select them with ( <b>Shift</b> )+click individually, you can apply the filter.</p> <p>If you inadvertently lose the selection by clicking off the items, you can get it back again almost immediately by re-applying the filter.</p>	

#### Learn more

- [Context Filter a Diagram](#) <sup>[789]</sup>
- [Package Browser](#) <sup>[673]</sup>
- [Diagram List](#) <sup>[684]</sup>
- [A Quick-Start Guide to Using Diagram Filters](#)

### 4.2.2.1 Work With Diagram Filters


You can modify the display of diagram components using diagram filters, which you create and use through the Diagram Filters window. This window is dockable, so you can move it around or 'fix' it next to the Diagram View while you activate, deactivate and edit the filters.

**Access**   **Diagram | Diagram Filters**

#### Create a filter to be applied to your diagrams

Step	Action	See also
<b>1</b>	<p>Perform one of the following actions:</p> <ul style="list-style-type: none"> <li>• On the Diagram Filter toolbar, click on the <b>New Filter</b> icon - the first on the left</li> <li>• Right-click on the list panel and select the <b>New Filter</b> context menu option</li> </ul> <p>The Create New Diagram Filter dialog displays.</p>	
<b>2</b>	In the <b>Enter Filter Name</b> field, type a name for the filter, then click on the <b>OK</b> button.	



Step	Action	See also
	The Diagram Filter dialog displays.	
3	<p>In the <b>Search On</b> field, either:</p> <ul style="list-style-type: none"> <li>• Use the default value of <b>Element</b> or</li> <li>• Click on the drop-down arrow and select <b>Connector</b></li> </ul> <p>If you select <b>Connector</b>, the dialog changes to offer a smaller set of properties. The <b>Visible</b> option enables you to filter out hidden or visible connectors that meet other filter conditions that you set.</p>	
4	Scroll through the properties to filter on, and select the checkbox against each property you require.	
5	<p>For each property, click on the <b>Condition</b> field and select, from the drop-down list, the comparison condition to be applied.</p> <p>Consider how the combination of <b>Condition - Equal To / Not Equal To</b> and <b>Filter Effect</b> might affect the results on the diagram.</p>	<a href="#">Diagram Filters</a> 
6	For each property, <b>double-click</b> on the <b>Value</b> field and type or select any specific value to filter on.	
7	Click on the <b>OK</b> button to save the filter and return to the Diagram Filters window.	

**Other Operations:**

Topic	Detail	See also
<b>Edit a Filter</b>	<p>To edit an existing filter on the Diagram Filters window, either:</p> <ul style="list-style-type: none"> <li>• Double-click on the filter name</li> <li>• Click on the filter name and select the <b>Properties</b> icon from the toolbar (the second icon from the left), or</li> <li>• Right-click on the name and select the <b>Properties</b> context menu option</li> </ul> <p>The Diagram Filter dialog displays; adjust the filtered fields as described above.</p>	
<b>Change the Name of the Filter</b>	<p>Right-click on the name and select the <b>Change Name</b> context menu option.</p> <p>The Create New Diagram Filter dialog displays.</p> <p>Type over the existing name with the new name, and click on the <b>OK</b> button.</p>	

Topic	Detail	See also
<b>Set Effect of Filters</b>	<p>To set how your filters identify selected items on your diagrams, click on the drop-down arrow of the toolbar <b>Filter Effect</b> field, and select one of the following options:</p> <ul style="list-style-type: none"> <li>• <b>Fade</b> - display all items that do <i>not</i> match the filter criteria in a pale version of the diagram background color</li> <li>• <b>Gray Scale</b> - display all items that do <i>not</i> match the filter criteria in pale gray</li> <li>• <b>Hide</b> - conceal all items that do <i>not</i> match the filter criteria</li> <li>• <b>Select</b> - select and highlight (with a hashed line) all those items that <i>do</i> match the filter criteria</li> </ul> <p>If the filter is for elements in the Package Browser or Diagram List, note that the <i>Select</i> filter effect cannot show up in this context.</p>	
<b>Enable Filters</b>	<p>To enable a filter so that it takes immediate effect on your diagrams, select the check box against the filter name.</p> <p>You can select more than one filter at a time, to combine their effects.</p>	
<b>Disable/ Clear Filters</b>	<p>To disable a filter and clear the effect of the filter on the diagram so that it displays in full, clear the checkbox.</p> <p>To disable all filters, either:</p> <ul style="list-style-type: none"> <li>• Click on the <b>Reload Filters</b> icon in the toolbar (third option from the left), or</li> <li>• Right-click on the list panel and select the <b>Reload Filters</b> context menu option</li> </ul>	
<b>Toggle between Diagram Filters and Context Filtering</b>	<p>To disable the filters you have defined and simply highlight a selected element and its immediate relatives, right-click on the Diagram Filters window background or the diagram background itself and select the <b>Context Filtering</b> option.</p> <p>To re-enable the Diagram Filter effects, display the Diagram Filters window or diagram context menus again and <b>deselect</b> the <b>Context Filtering</b> option.</p>	<a href="#">Context Filter a Diagram</a> <sup>789</sup>
<b>Delete a Filter</b>	<p>To remove a filter from the Diagram Filters window, either:</p> <ul style="list-style-type: none"> <li>• Click on the filter name and click on the <b>Delete</b> icon in the toolbar, or</li> <li>• Right-click on the filter name and select the <b>Delete Filter</b> context menu option</li> </ul>	

## 4.3 Trace: Tracking Dependencies



**Traceability** helps you to identify the way a given requirement or process has been implemented in a system, providing facilities to follow the path of dependencies from the initial request, through a modeled solution, to the deployed physical system or process. A well developed model provides full traceability in any direction, providing answers to questions such as "Why was that particular hardware used in the solution?", assuming there is traceability back to a requirement that specified particular response times or functional capability.

The result sets can be used to manage elements, to create documentation or to find the location of elements within the complete model structure.

### Traceability Tools

Enterprise Architect provides a number of tools for tracing the definition and implementation of a process, from initial requirement to generated code or technical deployment, or vice versa. Such tools include the:

- Traceability window
- Relationship Matrix
- Gap Analysis Matrix
- Relationships window
- Project Browser, and
- Traceability diagrams

The **Traceability window**, in particular, is designed to provide very detailed information on an element's relationships and dependencies, both immediate and distant.

### Tracing Transformations

If you have performed any **transformations** in developing your model and code, Enterprise Architect automatically creates **Transformation Dependency** connectors that you can trace - using the Traceability window - to establish what objects and code have been generated from each PSM element, or what the initial PSM element was for a generated object.

Whether you use transformations or develop the stages of the model in other ways, you can build up a range of **Traceability diagrams** (Custom diagrams) to identify the development pathway and the dependencies between entities such as Requirements, Use Cases, Classes, Packages, Test Cases and other model artifacts, or even between these entities and the overall business process model.

### Learn more

- [Traceability Tools](#) <sup>[724]</sup>
- [Gap Analysis Matrix](#) <sup>[745]</sup>
- [The Project Browser](#) <sup>[646]</sup>
- [Example Traceability Diagram](#) <sup>[743]</sup>

- [Model Transformation](#)<sup>[2013]</sup>
- [A Complete Business Process](#)<sup>[1813]</sup>

### 4.3.1 Traceability Tools

The **structure** of your model and a **Traceability diagram** act as the starting points for tracing the definition, design and implementation of a specific feature of a system or process. By applying tools such as the **Relationship Matrix** and **Traceability window**, you can follow threads through the model to determine how the feature is implemented and tested. You can also obtain information on what elements realize and are realized by the elements in a given package, using the **Dependency report** and **Implementation report**, respectively.

#### Principal Tools

Tool	Detail	See also
<b>Traceability window</b>	<p>The Traceability window is a very useful and versatile traceability tool; starting with a Traceability diagram or a package structure in the Project Browser, you can use the Traceability window to quickly explore the relationship chain of which any element is a component.</p> <ul style="list-style-type: none"> <li>• When you click on <b>an element</b>, it immediately becomes the top point in the Traceability window</li> <li>• When you click on the <b>background of a diagram</b>, <i>all</i> elements in the diagram are listed in the Traceability window, and you can follow the threads starting at each element through the diagram</li> </ul> <p>If you require a rapid, broad-brush view of <b>relationship flows</b> in the project structure, starting with a general list of - say - all functional Requirements, you can use a combination of Model Search, Project Browser and Traceability window; this is a powerful means of scanning your project, identifying how elements have been organized, and seeing how they interact. For example, the Model Search would list all the Requirements. You could rapidly click on each element and immediately see in the Project Browser where it has been grouped, and at the same time - in the Traceability window - how that element interacts with other elements in the model.</p> <p>By moving the cursor around a diagram or the Project Browser, and/or changing the relationship type combinations in the Traceability window, you can quickly see how elements are connected and how they influence each other. For example, you could see that - say - REQ017 is realized by two Use Cases, so you might then explore what else influences and is influenced by these two Use Cases. The Traceability window takes you well beyond what is likely to be depicted on any single diagram.</p> <p>If you have used <b>transformations</b> to develop your model, you can also follow the Transformation Dependencies that exist between an element in a PIM and elements in the PSMs.</p>	<p><a href="#">Example Traceability Diagram</a><sup>[743]</sup></p> <p><a href="#">The Traceability Window</a><sup>[723]</sup></p> <p><a href="#">Model Search</a><sup>[700]</sup></p> <p><a href="#">The Project Browser</a><sup>[646]</sup></p>
<b>Relationship Matrix</b>	<p>Using the Relationship Matrix, you can both create and study the relationships between, for example, the Requirements and Use Cases of a module.</p> <p>You might identify the 'theme' package (such as <i>Manage Users</i>) as the source package in the <b>Requirements</b> model and the target package in the <b>Use Case</b> model, and explore the likely element and connector types in the packages; this, like the Traceability diagram, identifies which</p>	<p><a href="#">Relationship Matrix</a><sup>[727]</sup></p>

Tool	Detail	See also
	<p>Requirements are (or should be) realized by which Use Cases.</p> <p>You can then perform similar checks with the <i>Manage Users</i> packages in, say, the <b>Use Case</b> and <b>Implementation</b> models.</p> <p>Using the <b>Source</b> and <b>Target</b> field browsers ( ... ) you can examine <b>child</b> packages within the 'theme' package, and obtain further detail on how the feature at this stage is defined.</p>	
<b>Relationships window</b>	<p>The Relationships window shows a single level of relationships from the currently selected element.</p> <p>Each line corresponds to a single relationship, providing information about that relationship and the element at the other end of the relationship.</p>	<a href="#">The Relationships Window</a> <sup>[742]</sup>
<b>Dependency report</b>	The Dependency report provides a list of dependencies for all elements in the model.	<a href="#">Dependency Report</a> <sup>[2738]</sup>
<b>Implementation report</b>	The Implementation report provides a list of all elements in the model that have to be implemented, and the elements that implement them.	<a href="#">Implementation Report</a> <sup>[2737]</sup>

### 4.3.2 The Traceability Window

Using the Traceability window you can quickly see how elements are connected and how they influence each other. You can:

- Locate related elements in the Project Browser
- View the properties of related elements
- Open the diagrams in which related elements are used
- Add related elements to the current diagram
- View the source code for related elements
- Change the display using the window toolbar options (see below)

When an **element** is selected, it immediately becomes the top point in the Traceability window. When a **diagram** is selected, all elements in the diagram are listed in the Traceability window, so you can use any of them as a starting point.

**Access** **View | Traceability** ( **Ctrl+Shift+4** )

#### Traceability window toolbar options

Option/Icon	Action	See also
<b>Locate Element</b>	Click on this icon to make the selected element the focus of the Traceability window and to show all relationships and elements with which the selected element forms a	

	<p>relationship chain.</p> <p>Click on the diagram to list all elements in the diagram.</p>	
<b>Related Element Properties</b>	Click on this icon to display the element Properties dialog for the currently-selected element in the Traceability window.	
<b>Find Element Usage</b>	<p>Click on this icon to display the Element Usage dialog, listing all diagrams in which the currently-selected element is used.</p> <p>(Shortcut keys: <b>Ctrl+U</b>)</p>	<a href="#">Show Element Use</a> <sup>[910]</sup>
<b>(Relationship selector)</b>	<p>Click on the drop-down arrow and select the checkbox against each relationship type to include in the trace shown in the window:</p> <ul style="list-style-type: none"> <li>• Generalizations</li> <li>• Aggregations</li> <li>• Nesting</li> <li>• Realizations</li> <li>• Dependencies</li> <li>• Transitions</li> <li>• Other (unspecified) links such as: <ul style="list-style-type: none"> <li>• Associations</li> <li>• Use Cases</li> <li>• Delegates</li> <li>• Assemblies</li> <li>• Deployments</li> <li>• Information Flows</li> <li>• Manifests</li> </ul> </li> <li>• Classifiers - where an element is used as the classifier of another element</li> <li>• Embedded Element Re-uses - where a Port or Part is also represented in another part of the model</li> <li>• Transformations - where an element is created by running a Transformation from another element</li> <li>• Custom References that have been added between elements</li> <li>• Qualified Names - element names including the names of owning objects</li> </ul>	<a href="#">Generalization</a> <sup>[1408]</sup> <a href="#">Aggregation</a> <sup>[1392]</sup> <a href="#">Nesting</a> <sup>[1434]</sup> <a href="#">Realization</a> <sup>[1440]</sup> <a href="#">Dependency</a> <sup>[1404]</sup> <a href="#">Transition</a> <sup>[1446]</sup>  <a href="#">Classifiers and Instances</a> <sup>[1009]</sup> <a href="#">Inherited and Redefined Ports</a> <sup>[1385]</sup> <a href="#">Model Transformation</a> <sup>[2013]</sup>  <a href="#">Set Up Cross References</a> <sup>[916]</sup> <a href="#">Configure Diagram Display</a> <sup>[825]</sup>
<b>Help</b>	Click on this icon to display this Help topic.	

**Notes**

- To make it easier and faster to find the elements you are interested in, restrict the relationship types shown to the smallest set of relevant types
- As an alternative view, or to include the traceability information in documentation, you can develop a Traceability diagram using the Traceability window
- The Traceability window does not currently recognize **extended** relationship types defined by Technologies such as BPMN or Archimate; relationships defined by a Profile or Technology are identified using their base UML type, so, for example, Archimate 'Realization' and 'UsedBy' relationships are each reported as UML 'Dependency' relationships

**Learn more**

- [Example Traceability Diagram](#) 

**4.3.3 Relationship Matrix**

When you need to visualize or manage relationships quickly and definitively, you can use the **Relationship Matrix** - a convenient and simple tool for reporting and working on all the relationships in a selected structure. The Relationship Matrix is a spreadsheet display of relationships between model elements within a Package or between two different Packages. You can filter the selection of relationships shown, according to:

- The element type of the source and target elements in each relationship
- The relationship type and direction
- The Package(s) in which the source elements and target elements are held

The matrix shows all the relationships of a specified type between source and target elements by:

- Listing the source Package elements down the side of the matrix
- Listing the target Package elements across the top of the matrix, and
- If a relationship exists between a source and target element, changing the background color of the intersecting grid square and displaying an arrow indicating the direction of the relationship

The direction is a reflection of which element is the source element and which the target (it does not indicate the **Direction** property of the connector, as defined in the connector Properties dialog). The display might also show a 'bent arrow' icon, indicating that the same kind of connectors exist in **both** directions between the source and target elements. The colored squares are normally **green**, indicating that the **source** element is not locked (that is, the parent Package **has not** been checked in under version control); if the element **is** locked (the parent Package **has been** checked in) the color is **pink**.

If you click on any square in the matrix, the square, the row header and the column header are highlighted. Right-clicking on a square also gives you the options of creating, modifying and deleting relationships between elements with a single mouse click - a quick way to set up complex sets of element relationships with a minimum of effort.

You can also create additional **elements** in the source and target Packages, and quickly add relationships to and/or from them.

**Access** [View | Relationship Matrix](#) or [Project Browser Package Context Menu | Documentation | Open in Relationship Matrix | As Source](#) or [As Target](#)

Operations

Operation	Detail	See also
<b>Specify the element types</b>	<p>The Relationship Matrix defaults to show all element types. If you want to examine relationships between specific types of element, you can filter for those types.</p> <p>You define the element types - if necessary - as the first step in configuring the Relationship Matrix.</p> <p>Click on the drop-down arrow in the <b>Type</b> field in the <b>Source</b> row, and/or in the <b>Target</b> row, and in each case click on the required element type in the list.</p> <p>The Relationship Matrix display immediately refreshes.</p>	<a href="#">Set Element Type</a> <sup>[729]</sup>
<b>Specify the connector type and direction</b>	<p>The Relationship Matrix requires a relationship type and direction to operate on.</p> <p>In the <b>Link Type</b> field, click on the drop down arrow and on the type of relationship to show.</p> <p>In the <b>Direction</b> field, click on the drop-down arrow and on one of the connector directions to show, or on <b>Both</b> to show relationships in any direction.</p> <p>In each case, if any relationships of that type exist they are immediately shown on the Relationship Matrix.</p>	<a href="#">Set Connector Type and Direction</a> <sup>[730]</sup>
<b>Select the Source and Target Packages</b>	You need to set the source and target Packages in order to display relationships, but you do this <b>after</b> setting the connector and element types; as the Matrix automatically refreshes after each change, this is usually faster.	<a href="#">Set Source and Target Package</a> <sup>[731]</sup>
<b>Update, delete and create relationships</b>	Having seen what relationships currently exist between the selected elements, you can review the relationship and element properties, modify them, add more relationships or delete relationships that are no longer applicable.	<a href="#">Creating and Deleting Relationships</a> <sup>[737]</sup>
<b>Create new elements</b>	<p>If the source or target element <b>Type</b> field is set to a specific element type, you can add elements of that type to the matrix and the model.</p> <p>In the top left corner of the matrix itself, the <b>Target</b> and <b>Source</b> labels have a <b>+</b> button. When you click on this button, the element properties dialog displays, through which you can define a new element of the corresponding (target or source) type.</p> <p>If there are stereotyped elements of the type (such as Actors as defined in UML, TOGAF and Archimate) you are prompted to select the variant that you need.</p> <p>This option is not available for the target or source axis if the corresponding <b>Type</b> field is set to <b>&lt;All&gt;</b>.</p>	<a href="#">Set Element Type</a> <sup>[729]</sup>  <a href="#">Properties Dialog</a> <sup>[956]</sup>
<b>Modify the</b>	It is possible to tailor the display of information to, for example, list the	<a href="#">Relationship</a>



Operation	Detail	See also
<b>display of information</b>	Package elements in alphabetical order, show elements from the Package hierarchy, and hide or show additional element name components such as level numbering, aliases and parent Package names.	<a href="#">Matrix Options</a> [738]
<b>Toggle Between relationship direction and CRUD formats</b>	You can switch the display between the connector direction format and a CRUD matrix format. The CRUD (or other, customized) indicators are defined in an overlay that you create.	<a href="#">Matrix Overlays</a> [732]
<b>Export the contents of the Relationship Matrix</b>	After reviewing the Relationship Matrix, you can export the contents to a CSV file, or capture the image of the contents as a .png file or a .emf graphics file.	<a href="#">Relationship Matrix Options</a> [738]
<b>Print the contents of the Relationship Matrix</b>	You can also send the contents of the Relationship Matrix to a printer, after previewing the print format if necessary. The printout can be on as many pages as it takes, or you can scale the print format so that the file occupies a specific number of pages.	<a href="#">Relationship Matrix Options</a> [738]
<b>Capture the Matrix settings</b>	If you want to use the same Relationship Matrix settings repeatedly to, for example, monitor development of the same source and target Packages, it is possible to save the settings as a profile that can be called to reapply those settings.	<a href="#">Matrix Profiles</a> [735]
<b>Investigate Source and Target elements in relationships</b>	Having identified relationships that might be of significance to you, you can check details of the elements in those relationships. For example, you can look at: <ul style="list-style-type: none"> <li>• The diagrams that the elements are used in</li> <li>• The properties of the elements, or</li> <li>• What other elements might have relationships with the selected element</li> </ul>	<a href="#">Review Source and Target Elements</a> [741]

#### 4.3.3.1 Set Element Type

The Relationship Matrix defaults to show all element types in both Source and Target Packages. You can also select to list elements of one specific type in the Source Package or the Target Package or both. For example:

- Any (<All>) types of element in the Source Package linked to Use Cases in the Target Package
- Use Cases in the Source Package linked to any (<All>) types of element in the Target Package
- Requirements in the Source Package linked to Use Cases in the Target Package
- Classes in the Source Package linked to Classes in the Target Package

Having set the element type for an axis, you can create further elements of that type within the Relationship

Matrix,

#### Set the source and/or target element type

Step	Action	See also
1	Click on the drop-down arrow in the <b>Type</b> field for the Source Package, or for the Target Package.	
2	Find the required element type in the list and click on it. The Relationship Matrix content automatically refreshes.	
3	If necessary, repeat the process to select a specific element type in the other Package.	

#### Learn more

- [Relationship Matrix](#)<sup>[72]</sup>

### 4.3.3.2 Set Connector Type and Direction

The Relationship Matrix shows relationships of a specific type and direction, which you define after selecting the element type(s) to show, and before you specify the Source and Target packages to search in.

#### Set the connector type and direction

Step	Action	See also
1	Click on the drop-down arrow in the <b>Link Type</b> field.	
2	Locate and click on the required connector type in the list.	
3	Click on the drop-down arrow in the <b>Direction</b> field.	
4	Click on the required direction in the list. The Relationship Matrix content automatically refreshes.	

#### Notes

- If you set the **Direction** field to **Both**, each relationship is indicated by two arrows - a *From-To* arrow and a *To-From* arrow
- The direction is a reflection of which elements are the source elements and which are the target.; it

does not indicate the **Direction** property of the connector, as defined in the connector Properties dialog

#### Learn more

- [Relationship Matrix](#)

### 4.3.3.3 Set Source and Target Package

Before you display relationships on the Relationship Matrix, you set both the source and target packages containing the elements between which the relationships might exist. However, you set the source and target packages **after** setting the element and connector types/details; as the Relationship Matrix refreshes after each change, this is usually faster. You can set the same package as the source and target, to investigate relationships within a package, or different source and target packages to investigate relationships between packages.

When deciding on which packages to investigate and how to filter the matrix contents, consider that:

- The Relationship Matrix includes **all** child **elements** in each package, within the criteria you specify; in a large model, with less-specific filters, this can expand the matrix with a large number of elements
- On the other hand, if you want to examine relationships within the package structure, you can select options to include elements in **child packages** under either the source package or the target package, or both

**Access** [View | Relationship Matrix](#) or [Project Browser Package Context Menu | Documentation | Open in Relationship Matrix | As Source](#) or [As Target](#)

#### Set source and target packages

Step	Action	See also
<b>If you select the first access option</b>		
1	In the Project Browser, click on the required source package, then press and hold ( <b>Ctrl</b> ) and click on the required target package, to select the two packages together.	
2	Drag the selected packages over the <b>Source</b> and <b>Target</b> fields.  The first-selected package name displays in the <b>Source</b> field, and the second-selected package name displays in the <b>Target</b> field.	
<b>Or, if you use the second access option</b>		
1	Select and drag a single package name over the remaining <b>Source</b> OR <b>Target</b> field, to change just the source or the target package.  If you drop the package name anywhere else on the Relationship Matrix, the system prompts you to specify whether to add it to the <b>Source</b> or <b>Target</b> field, or both.	

Step	Action	See also
<b>Alternatively</b>		
1	<p>Click on the ( ... ) (Browse) button at the end of the <b>Source</b> or <b>Target</b> field.</p> <p>A short menu displays, with options to:</p> <ul style="list-style-type: none"> <li>• Locate a package using the Browse Project dialog or</li> <li>• Select a Model Search from a drop-down list and run the search on a search term; the name of the search is displayed in the <b>Source</b> or <b>Target</b> field</li> </ul> <p>The target of the search depends on whether the <b>Return matching items for the selected Package</b> option is selected in the Search definition; if it is selected, the search operates on the current package, otherwise the search operates on the whole model</p> <p>It is possible to browse for a package in one field and run a search in the other.</p>	<a href="#">Advanced Search Options</a> <sup>[715]</sup>
2	Select the required package, or the required Model Search and search term, and click on the <b>OK</b> button.	

In all cases, the Relationship Matrix immediately:

- Populates the axes with the elements identified in the Source and Target packages or searches that meet the selection criteria, and
- Shows any relationships between the sets of elements that also meet the selection criteria

#### **Notes**

- Custom SQL searches are supported if they are returning elements; the SQL must include `ea_guid AS CLASSGUID` (case sensitive)
- Using ORDER BY in a SQL search will not affect the order of the Relationship Matrix

#### **Learn more**

- [Relationship Matrix](#) <sup>[727]</sup>
- [Create Search Definitions](#) <sup>[717]</sup>

### **4.3.3.4 Matrix Overlays**

The Relationship Matrix indicates the existence of connectors between elements by displaying an **arrow** in the cell at the intersection of the source element row and target element column.

If you prefer, you can toggle the display to a **CRUD matrix overlay**, where the presence of a connector is indicated by the appropriate **value** or **combination** of CRUD values in the intersection cell. You define the actual values that are entered and displayed, so the characters can refer to whatever interpretation of CRUD your organization uses, or they can be another standard set of indicators such as **BREAD**, or your own custom set of indicators. The values can, therefore, be a selection of the following, or anything else you

might use:

- **Create**
- **Add**
- **Browse**
- **List**
- **Read, Retrieve**
- **Update**
- **Edit**
- **Modify**
- **Delete, Destroy**

#### Toggle between the two display formats

Click on the **Overlays** field drop-down arrow and select either:

- **<None>** to use the direction arrow Relationship Matrix format
- The name of your overlay, to use the Matrix overlay format, or
- **<New Overlay>** to create and use a new Matrix overlay

The overlay values display if they have been specifically **applied** to the relationship via the **intersection cell** on the Relationship Matrix.

#### Learn more

- [Relationship Matrix](#)<sup>[727]</sup>
- [Create a Matrix Overlay](#)<sup>[733]</sup>
- [Apply an Overlay to a Cell](#)<sup>[734]</sup>

#### **4.3.3.4.1 Create a Matrix Overlay**

When you create a Matrix Overlay to apply to the cells of the Relationship Matrix, you define the values that can be assigned under that overlay, and whether the values must be single or can be set in a combination. The overlay definition is actually created as a Tagged Value on the connector, of the predefined Tagged Value Type **MatrixOverlay**.

Access   **View | Relationship Matrix: Overlays drop-down arrow | <New Overlay>**

#### Create an Overlay

Field/Button	Action	See also
<b>Overlay Name</b>	Type an appropriate name for the overlay. This also becomes the <b>tag</b> name.	
<b>Allowable Values</b>	Type in the allowable values for the overlay/tag, <b>not</b> separated by any punctuation or spaces (for example, <b>CRUD</b> ); the matrix cell is case sensitive, so type the values exactly as they are to be entered.	

Field/Button	Action	See also
<b>Mutually Exclusive</b>	<p>If you want only <b>one</b> allowable value to be set in a matrix cell at a time, select this checkbox. (This makes the Tagged Value an <b>Enum</b> type.)</p> <p>If you want to allow any <b>combination</b> of values to be set in a matrix cell at a time, clear the checkbox. (This makes the Tagged Value a <b>Custom</b> type.)</p>	
<b>OK</b>	<p>Click on this button to save the new overlay/Tagged Value.</p> <p>You can now view the overlay on the Tagged Value Types tab of the UML Types dialog, in the format:</p> <pre>Type=Enum; Values=C, R, U, D; MatrixOverlay=true;</pre>	<a href="#">Tagged Value Types</a> <small>[1150]</small>

#### Notes

- You cannot change an existing overlay on the Relationship Matrix; because the overlay is defined by a Tagged Value of type **MatrixOverlay**, you can delete the overlay or edit the values and/or the **Mutually Exclusive** setting by deleting or editing the Tagged Value on the Tagged Value Types tab of the UML Types dialog

#### Learn more

- [Matrix Overlays](#) [732]
- [Apply an Overlay to a Cell](#) [734]
- [Predefined Tagged Value Types](#) [1622]

#### 4.3.3.4.2 Apply an Overlay to a Cell

You can apply overlays (one overlay at a time) to the cells of the Relationship Matrix. After you select the required overlay in the **Overlays** field, you can:

- Apply an overlay value or values to a cell that identifies an existing relationship
- Create a new relationship in a cell, and immediately apply overlay values to that relationship
- Change the overlay values currently applied, or
- Remove the overlay from the cell

**Access** [View | Relationship Matrix: right-click on cell](#)

#### Context Menu - Overlay Options

Option	Action	See also
<b>Apply overlay</b>	<p>(On a cell that has an arrow icon, or the value of a different type of overlay.)</p> <p>Displays the Allowable values &lt;values&gt; dialog. Type in the required value or values and click on the <b>OK</b> button.</p>	

Option	Action	See also
	<ul style="list-style-type: none"> <li>If you have entered more than one value and the overlay is set to mutually exclusive, only a single value is permitted and an error message displays; click on the <b>OK</b> button and type in a single character</li> <li>If the overlay is not set to mutually exclusive, you can type in any individual value, or any combination of values in any sequence</li> </ul>	
<b>Create new relationship with overlay</b>	<p>(On a cell that has no relationship indicated in it.)</p> <p>If necessary, change the <b>Link Type</b> field to show the type of connector to create.</p> <p>When you click on the menu option, a submenu of variations of the selected relationship type might display. Click on the variation to create.</p> <p>The Allowable values &lt;values&gt; dialog displays; type in the required value or values and click on the <b>OK</b> button.</p> <p>A relationship of the selected type is created between the source and target elements, and the selected overlay values are displayed in the cell.</p>	<a href="#">Relationship Matrix</a> <sup>[72]</sup>
<b>Edit overlay</b>	<p>(On a cell showing a value of the current overlay.)</p> <p>Displays the Allowable values &lt;values&gt; dialog, showing the current value. Overtyping this with the new value or values and click on the <b>OK</b> button.</p> <p>The new value(s) display in the selected cell.</p>	
<b>Remove overlay</b>	<p>(On a cell showing a value of the current overlay.)</p> <p>Clears the overlay value and restores the appropriate connector direction arrow symbol to the cell.</p>	

**Notes**

- You cannot apply different overlays to the matrix at the same time; if you select a different overlay in the **Overlays** field, any values in the cells revert to either the value set under that overlay, or the direction arrow if no value is set under that overlay

**Learn more**

- [Relationship Matrix](#)<sup>[72]</sup>
- [Matrix Overlays](#)<sup>[73]</sup>
- [Create a Matrix Overlay](#)<sup>[73]</sup>

**4.3.3.5 Matrix Profiles**

You can use the Relationship Matrix as a monitoring tool, to help in tracking the development of elements and relationships in particular packages or pairs of packages over time. In this case you would re-use the same combinations of field settings many times. It is very easy to store a combination of settings as a Matrix Profile, so that later you can recall it for use whenever you need it.

**Access** **View | Relationship Matrix****Save a Relationship Matrix configuration as a named Profile**

Step	Action	See also
1	Set up the Relationship Matrix with the required: <ul style="list-style-type: none"> <li>Source and target element types</li> <li>Connector type and direction</li> <li>Source and target packages or Model Searches, and</li> <li>Overlay name or <b>&lt;None&gt;</b> selected</li> </ul>	<a href="#">Relationship Matrix</a> <sup>[72]</sup> <a href="#">Set Source and Target Package</a> <sup>[73]</sup> <a href="#">Matrix Overlays</a> <sup>[73]</sup>
2	Click on the <b>Options</b> button on the Relationship Matrix and select the <b>Profiles   Save as New Profile</b> menu option.  The Enter name for new profile dialog displays.	
3	In the <b>Enter Value</b> field, type a profile name of up to 12 characters.  Click on the <b>OK</b> button.	
4	Once you have created a profile, you can select it by clicking on the <b>Profile</b> drop-down arrow at the top of the Relationship Matrix screen and selecting the profile name from the list.  Profiles are also listed in the <i>Matrix Profiles</i> folder of the Resources window; double-click on a profile in the folder to display the Relationship Matrix with that profile active.	<a href="#">Resources</a> <sup>[1173]</sup>

**Notes**

- You can modify an existing profile by:
  - Selecting the profile name from the **Profile** drop-down list
  - Changing the field values, then
  - Selecting the **Options: Profiles | Update Current Profile** menu option
- To delete an existing profile, select it in the **Profile** drop-down list and select the **Options: Profiles | Delete Current** menu option
- You can capture Relationship Matrix profiles in document reports

**Learn more**

- [Reporting Profiled Relationship Matrices](#) <sup>[2692]</sup>



#### 4.3.3.6 Creating and Deleting Relationships

Having reviewed existing relationships on the Relationship Matrix, you can use the Matrix to quickly modify or delete those relationships and create others between the elements. This is a very useful facility for maintaining a large number of relationships in a short space of time. As you can filter for specific types of connector and source and target element, you can create a logical set of relationships without having to specify the type of each object every time.

**Access** [View | Relationship Matrix](#)

##### Create a new relationship

Step	Action	See also
1	In the <b>Link Type</b> field, select the type of relationship you want to create, and in the <b>Type</b> fields select the types of source and target element between which to create the relationships.	
2	Right-click on the empty square at the intersection of the appropriate source row and target column.  You can select and operate on several squares (creating several relationships) at once. Either: <ul style="list-style-type: none"> <li>Press <b>Ctrl</b> as you click on each intersecting square, then right-click on one of the selected squares, or</li> <li>Click on the first square in the required row, column or block, then press <b>Shift</b> as you click on the final square in the required row, column or block; right-click on one of the selected squares</li> </ul>	
3	Select either the: <ul style="list-style-type: none"> <li><b>Create new relationship</b> option, if no overlay is to be applied</li> <li><b>Create new relationship with overlay</b> option, to apply an overlay as you create the relationship</li> </ul> <p>If the <b>Direction</b> field is set to <b>Both</b>, you are offered the choice of creating the relationship to the Source element or to the Target element.</p> <p>A further submenu might display, listing any variations on the type of relationship you can create; in this case, click on the required type of relationship.</p>	<a href="#">Apply an Overlay to a Cell</a> <sup>734</sup>
4	A connector of the selected type is created between the two elements, and the appropriate indicator displays in the matrix cell.	

##### Modify or delete a relationship on the Relationship Matrix

Step	Action	See also
1	<p>Right-click the required highlighted relationship cell on the matrix, and select the appropriate context menu option:</p> <ul style="list-style-type: none"> <li>• <b>Apply overlay</b> - apply the CRUD overlay, if one is selected</li> <li>• <b>Edit overlay</b> - change the overlay values assigned to the cell</li> <li>• <b>Remove overlay</b> - take the overlay off this cell only, and revert to the connector direction arrow icon</li> <li>• <b>View relationship</b> - opens the Properties dialog for the selected relationship</li> <li>• <b>Source element properties</b> - opens the Properties dialog for the source element</li> <li>• <b>Target element properties</b> - opens the Properties dialog for the target element</li> <li>• <b>Delete relationship</b> - removes the relationship from between the elements</li> </ul>	
2	<p>If you have selected <b>Delete relationship</b>, Enterprise Architect prompts you to confirm this action.</p> <p>Click on the <b>Yes</b> button.</p> <p>The <b>Delete relationship</b> option is not available if:</p> <ul style="list-style-type: none"> <li>• The source element (that is, the owner) is locked</li> <li>• You have selected <b>Both</b> in the <b>Direction</b> field - you are effectively trying to delete half a relationship</li> </ul>	
3	<p>If you have selected one of the other options, modify the properties in the dialog fields as required.</p> <p>Click on the <b>OK</b> button to save the changes.</p>	<p><a href="#">Apply an Overlay to a Cell</a> <sup>[734]</sup></p> <p><a href="#">Connector Properties</a> <sup>[1126]</sup></p> <p><a href="#">Properties Dialog</a> <sup>[956]</sup></p>

#### Learn more

- [Relationship Matrix](#) <sup>[727]</sup>

### 4.3.3.7 Relationship Matrix Options

When you have displayed information on the Relationship Matrix, you might want to capture the display in some way, or modify the output. There are a number of options that you can select to, for example:

- Output the information on the Relationship Matrix to the printer or to a metafile, .png file or .csv file
- Create and update profiles of the configurations of the matrix that you have designed
- Define local settings to control what the Relationship Matrix displays

**Access** **View | Relationship Matrix: Options**

Options

Option	Action	See also
<b>Print Relationship Matrix</b>	<p>Click on the <b>Matrix   Print</b> menu option.</p> <p>The Print dialog displays, on which you select the output printer and specify the printer properties, the range of pages to print, and the number of copies.</p> <p>The output is a WYSIWYG representation of the Relationship Matrix contents.</p>	
<b>Display a preview of the printout</b>	<p>Click on the <b>Matrix   Print Preview</b> menu option.</p> <p>The Print Preview screen displays, showing the Relationship Matrix printout.</p>	<a href="#">Print Preview</a> <sup>[783]</sup>
<b>Scale the printout</b>	<p>When you print the Relationship Matrix, by default it prints on as many pages wide and long as the Matrix requires.</p> <p>You can scale the printout into a fixed number of pages wide; the row height is automatically adjusted to maintain the proportions of the Matrix. This reduces the overall size of the printout and improves appearance, especially when used in conjunction with the <b>Landscape</b> option in the printer properties.</p> <p>For example, a 16-page printout without scaling can, with a scaling of 2 pages wide, be reduced to 6 pages.</p>	
	<p>To set the page scaling:</p> <ol style="list-style-type: none"> <li>1. Click on the <b>Matrix   Scale Setting</b> menu option. The Scale Matrix dialog displays.</li> <li>2. Select the <b>Scale Matrix Width Into Pages</b> checkbox, and type or select the number of pages in width to scale to.</li> <li>3. Click on the <b>OK</b> button to apply the setting.</li> </ol>	
<b>Save Relationship Matrix as graphic file</b>	<p>Click on the <b>Matrix   Save as Metafile</b> or <b>Matrix   Save as PNG</b> menu options.</p> <p>A Browser dialog displays on which you select the target file location and specify the filename of the .emf or .png file in which to save the output.</p> <p>You can incorporate these files in a document or web report, as either a hyperlinked file or an included file.</p>	<a href="#">Insert Images, Objects and Frames</a> <sup>[1085]</sup>  <a href="#">Insert Reference Links</a> <sup>[1083]</sup>
<b>Export output to CSV file</b>	<p>The contents of the Relationship Matrix can be exported to a CSV file, which provides a convenient mechanism for moving the matrix data to a spreadsheet environment such as Microsoft Excel.</p> <p>This option is also active in the 'Lite', read-only version of Enterprise Architect.</p> <p>To export the Matrix to CSV:</p>	<a href="#">The Read-only 'Lite' Edition</a> <sup>[23]</sup>

Option	Action	See also
	<ol style="list-style-type: none"> <li>1. Select the <b>Matrix   Export to CSV</b> menu option. The Windows Browser dialog displays.</li> <li>2. Browse to the required file location and type in a .CSV filename to export to.</li> <li>3. Click on the <b>Save</b> button to export the data.</li> </ol>	
<b>Create and manage matrix profiles</b>	<p>If you use a particular combination of field values often, you can save that combination as a profile to make it easier to recall them for use.</p> <p>You can also include a Relationship Matrix Profile in a document report, to pull in specific relationship details using the Matrix.</p>	<a href="#">Matrix Profiles</a> <sup>[735]</sup> <a href="#">Reporting Profiled Relationship Matrices</a> <sup>[2692]</sup>
<b>Manage display content</b>	<p>You can extend the information shown by the Relationship Matrix to include related elements and/or additional properties of each element. You can also re-organize the display to list the element names in alphabetical order.</p> <p>Select the <b>Options</b> menu option. The Matrix Options dialog displays. Select one or more of the following checkboxes to define the information you want to display:</p> <ul style="list-style-type: none"> <li>• <b>Include Source Children</b> - to recursively include child packages and contents under the Source</li> <li>• <b>Include Target Children</b> - to recursively include child packages and contents under the Target</li> <li>• <b>Include All Extended Meta Types</b> - to include elements that are extensions of a specified meta-type; for example, if there are Block elements (extending Class) in the package, selecting this option and specifying the type Class includes Class and Block elements, and any further derivatives of Class in the matrix</li> <li>• <b>Sort Axes</b> - to display the element names in alphabetical order</li> <li>• <b>Show Package Names</b> - to hide or show the elements' parent package names in the Relationship Matrix; this is useful for shortening the displayed texts, especially in circumstances where packages have long names</li> <li>• <b>Use Element Alias If Available</b> - to display an element's alias instead of the element name, if an alias has been defined</li> <li>• <b>Show Level Numbering If Available</b> - to reproduce level numbering in the Relationship Matrix, if it is turned on in the Project Browser</li> </ul>	<a href="#">General Settings</a> <sup>[958]</sup> <a href="#">Package Menu</a> <sup>[651]</sup>

#### Learn more

- [Relationship Matrix](#) <sup>[727]</sup>

#### 4.3.3.8 Review Source and Target Elements

As you review the information on **relationships** in the Relationship Matrix, you can also obtain information on the source and target **elements** in any of the relationships.

Access **View | Relationship Matrix**

##### Review the elements

Action	Detail	See also
<b>Identify other elements in relationships with a source or target element</b>	Click on the source or target element name in the row or column titles. The entire row or column is highlighted. Scroll across or down the highlighted row or column and quickly identify where the relationships are; this is very useful if the row or column is long.	
<b>Display the Properties dialog for the selected element</b>	Right-click on the element name and select the <b>Properties</b> context menu option.	<a href="#">Properties Dialog</a> <sup>[956]</sup>
<b>Identify diagrams in which the element is used</b>	Right-click on the element name and select the <b>Find in Diagrams</b> context menu option. Either: <ul style="list-style-type: none"> <li>The only diagram in which the element is used displays, with the element highlighted, or</li> <li>A list of the diagrams in which the element is used displays; you then double-click on the required diagram to open it</li> </ul>	<a href="#">Show Element Use</a> <sup>[910]</sup>
<b>Highlight the element name in the Project Browser</b>	Right-click on the element name and select the <b>Locate in Project Browser</b> option. The Project Browser expands to the location of the element, and the element name is highlighted.	
<b>Make the element the focus in any docked screens or windows that are open</b>	Right-click on the element name and select the <b>Set Context Item</b> option; the selected element becomes the subject of any other windows or screens that are open.	

##### Learn more

- [Relationship Matrix](#)<sup>[727]</sup>

### 4.3.4 The Relationships Window

If you want a **quick overview** of all the relationships of a single element, you can obtain this from the **Relationships window**. This window lists all the relationships of the element currently selected in a diagram or the Project Browser, showing:

- The relationship type
- Any stereotype on the relationship
- The source and target elements in the relationship (the selected element can be either one in the relationship)
- The direction of the relationship
- Any stereotype on the related element
- The related element type
- The roles, if defined, of the target and source elements in the relationship

The Relationships window also has a **Target in Diagram** column which, if the row contains the value **Yes**, indicates that the related element is visible in the currently open diagram. If the related element is **not** in the current diagram (the row has no value), you have the option of adding it.

**Access** **Element | Relationships** (Ctrl+Shift+2)

#### Facilities

Facility	Detail	See also
<b>Display Connector Properties</b>	Double-click on a connector in the list.  The <connector type> Properties dialog displays, on which you can view and edit the connector properties.	<a href="#">Connector Properties</a> [1128]
<b>Find Related Element</b>	Right-click on the item line and select the <b>Locate Related Element</b> context menu option.  The related element is highlighted in the Project Browser.	
<b>Display Related Element Properties</b>	Right-click on the item line and select the <b>View Related Element Properties</b> context menu option.  The element Properties dialog displays.	<a href="#">Properties Dialog</a> [956]
<b>Delete Relationship</b>	Right-click on the item line and select the <b>Delete Connection</b> context menu option.  A prompt displays to warn you that the selected items will be deleted from all diagrams and from the model, and that the action cannot be reversed. This message refers to the <b>relationship</b> only - the source and target elements remain in the diagram and in the model.  Click on the <b>Yes</b> button to proceed with the deletion.	

Facility	Detail	See also
<b>Hide/Show Connector</b>	<p>If the related element exists on the current diagram, and the connector is shown, you can hide it. Right-click on the item line and select the <b>Hide Relation</b> context menu option.</p> <p>If the related element exists on the diagram and the connector is hidden, you can re-display it. Select the <b>Show Relation</b> context menu option.</p>	
<b>Locate Related Elements in Diagrams</b>	<p>Right-click on the item line and select the <b>Find in all Diagrams</b> context menu option.</p> <p>If the related element exists only in one diagram, that diagram is displayed with the related element highlighted. The focus of the Relationship window switches to the related element.</p> <p>If the related element exists in more than one diagram, the Element Usage dialog displays, listing the diagrams. Click on one of the diagrams and on the <b>Open</b> button to display the selected diagram with the related element highlighted.</p>	<a href="#">Show Element Use</a> [910]
<b>Add Related element to current diagram</b>	<p>Right-click on the item line and select the <b>Place Related Element in Diagram</b> context menu option. The cursor changes to the 'drop element' symbols (small rectangle and arrow with a boxed plus sign); move the cursor to the point in the diagram at which to drop the related element, and click the mouse button.</p> <p>This facility is useful in building up a picture of how an element interacts with the rest of the model, especially when reverse engineering an existing code base.</p>	

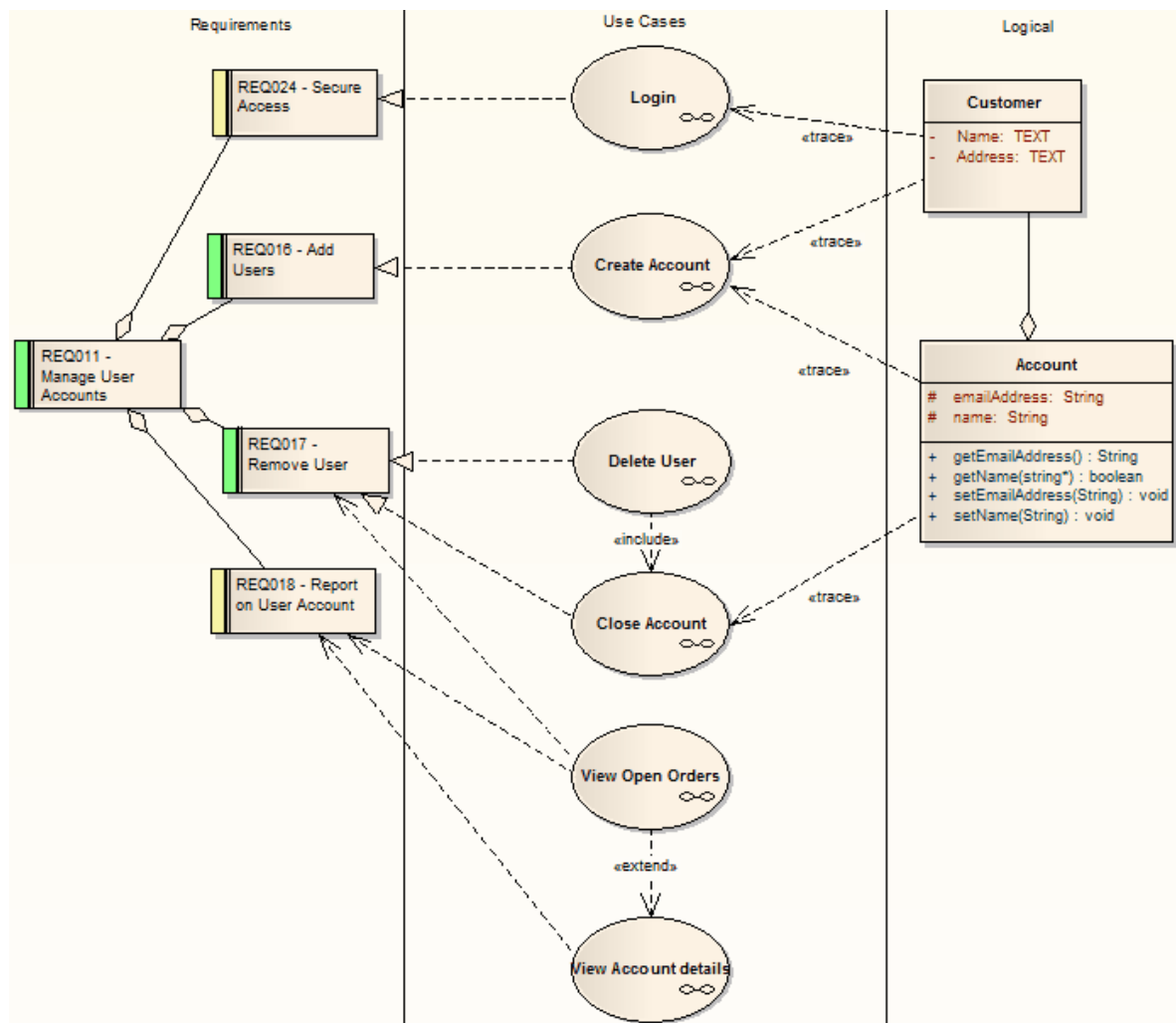
#### Notes

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions, with security set ON, the diagram and the source and target elements must be free for editing before these options are available for use
- Adding a related element from the Relationships window to the diagram does not change the focus of the window to that related element; the originally-selected element (and the original subject of the Relationships window) remains in context, and you continue working with that element.

### 4.3.5 Example Traceability Diagram

You can create **Traceability diagrams** to show a wide variety of relationships, and you are not restricted to having either a single element type or elements from a single phase on a diagram. Therefore, you can create appropriate relationships (such as Realize or Trace) to elements where no relationship was previously defined.

Consider the following diagram, which shows how a series of requirements are traced to Use Cases and then Classes:



The diagram instantly shows:

- That two levels of Requirements are realized by Use Cases
- Which Requirement is realized by which Use Case(s)
- How some of the Use Cases are implemented by Class elements

You can drill down on each Use Case (or, in other Traceability diagrams, any other composite element) to display more detailed diagrams showing how the Use Case meets the Requirement; the *Close Account* Use Case, for example, is a composite element containing a Communication diagram and a Sequence diagram.

You can tailor your Traceability diagrams to depict any level of granularity and any stages of development that are appropriate; for example, you might:

- Narrow the above diagram to show development from just the *Remove User* Requirement, or
- Extend it to include Interfaces, Components, Test Case elements or any other facet of the system or process

### Building a diagram



A number of tools are available to for creating diagrams that show traceability:

- Select the diagram context menu option **Add | Related Elements** to automatically add elements linked to the selected element
- On the Relationships window, use the context menu option **Place Related Element in Diagram** to add elements to the diagram that are directly linked to the selected element
- On the Traceability window, add elements that are indirectly linked

#### Notes

- Whilst the Traceability diagram itself provides information on the definition, design and implementation of a business process feature, much more information can be obtained using the Traceability tools

#### Learn more

- [Realization](#) <sup>[1440]</sup>
- [Insert Related Elements](#) <sup>[933]</sup>
- [The Relationships Window](#) <sup>[742]</sup>
- [Trace](#) <sup>[1446]</sup>
- [Traceability Tools](#) <sup>[724]</sup>
- [The Traceability Window](#) <sup>[725]</sup>

### 4.3.6 Gap Analysis Matrix

The Gap Analysis Matrix is a tool to analyze model artifacts for potential gaps in solutions.

The idea is to highlight a shortfall between the Baseline Architecture and the Target Architecture; that is, items that have been deliberately omitted, accidentally left out, or not yet defined.

#### Access **View | Gap Analysis Matrix**

Topic	Detail	See also
<b>Overview</b>	<p>The Gap Analysis Matrix is a convenient method of identifying gaps between solution architectures.</p> <p>It enables you to model the gaps in the repository, which can later be addressed and assigned as tasks; the identified gaps can also be used to prioritize activities.</p> <p>On the Gap Analysis Matrix, you select the:</p> <ul style="list-style-type: none"> <li>• Target Architecture package</li> <li>• Baseline Architecture package</li> <li>• Type of Target Architecture artifact</li> <li>• Type of Baseline Architecture artifact</li> <li>• Type of element to model the gap</li> </ul>	<a href="#">Include Other Users' Packages</a> <sup>[436]</sup>
<b>Operations</b>	<p>Once you open the Gap Analysis Matrix, you can:</p> <ul style="list-style-type: none"> <li>• Create, update and delete Gap Matrix Profiles to monitor</li> </ul>	<a href="#">Edit Gap Notes</a> <sup>[748]</sup>

Topic	Detail	See also
	<p>development of the Baseline and Target packages</p> <ul style="list-style-type: none"> <li>• Edit Gap notes during a discussion session for Gap analysis</li> <li>• Create elements to address gaps</li> <li>• Add links to existing elements to address gaps</li> <li>• Remove links to existing elements modeled as gap</li> </ul>	
<b>Select Package</b>	<p>Click on the ( ... ) (Browse) button at the end of the <b>Target Architecture</b> or <b>Baseline Architecture</b> field.</p> <p>The Browse Project dialog displays.</p> <p>Select the required package and click on the <b>OK</b> button; by default all the elements under the package are displayed.</p>	
<b>Set Element Type</b>	<p>Click on the drop-down arrow in the <b>Filter</b> field for the Target Architecture package, and/or for the Baseline Architecture.</p> <p>Find the required element type in the list and click on it; Enterprise Architect refreshes the matrix content.</p>	
<b>Set Gap Type</b>	<p>Click on the drop-down arrow in the <b>Record Gap As</b> field.</p> <p>Find the required element type in the list and click on it.</p> <p>When you create a gap, the specified type of element is created; for example, if you want to address the gap with an issue element then select <b>Issue</b> for this field.</p>	
<b>Baseline Architecture</b>	<p>The Baseline Architecture artifacts are listed along the rows of the matrix.</p> <p>The last row is for capturing the new elements in the target architecture along the columns but not in the baseline architecture, hence the name <b>New</b>.</p>	
<b>Target Architecture</b>	<p>The Target Architecture artifacts are listed along the columns of the matrix.</p> <p>The last column is for capturing the missing or intentionally eliminated elements in the baseline architecture along the rows but not in the target architecture; hence the name <b>Missing / Eliminated</b>.</p>	
<b>Modeling Gap</b>	<p>Right-click on a cell in the <b>New</b> row or <b>Missing / Eliminated</b> column.</p> <p>The context menu displays options to:</p> <ul style="list-style-type: none"> <li>• Create a Gap element</li> <li>• Add a link to an existing Gap element</li> </ul> <p>If a link to Gap element already exists in the cell then the following options are displayed:</p> <ul style="list-style-type: none"> <li>• Edit Gap element</li> </ul>	

Topic	Detail	See also
	<ul style="list-style-type: none"> <li>Find in Project Browser...</li> <li>Remove Gap element link</li> </ul>	
<b>Create Gap</b>	<p>Right-click on the cell and select the <b>Create Gap Element</b> context menu option.</p> <p>The Browse Project dialog displays.</p> <p>Select the package in which to create the Gap element and click on the <b>OK</b> button; a Gap element is created in the selected package and its Properties dialog displays to enable you to enter the element name and other required properties.</p>	
<b>Link to Existing Gap</b>	<p>If you intend to use a Gap element that is already available in the model, right-click on the appropriate cell in the <b>Missing / Eliminated</b> column or <b>New</b> row and select the <b>Link to Existing Gap Element</b> context menu option.</p> <p>The Select Classifier dialog displays, enabling you to select the existing Gap element.</p>	
<b>Remove Link to Gap</b>	<p>If you intend to remove a link to the Gap element in a cell, right-click on the appropriate cell in the <b>Missing / Eliminated</b> column or <b>New</b> row and select the <b>Remove Link to Gap Element</b> context menu option.</p> <p>The link is removed from the cell but the element still exists in the Project Browser.</p>	
<b>Review Gap Element</b>	<p>Right-click on the appropriate cell in the <b>Missing / Eliminated</b> column or <b>New</b> row and select the <b>Edit Gap Element</b> context menu option.</p> <p>The Properties dialog displays, enabling you to edit the selected Gap element.</p> <p>To locate the element in the Project Browser select the <b>Find in Project Browser...</b> context menu option, which highlights the element in the Project Browser.</p>	
<b>Gap Analysis Matrix Profiles</b>	<p>On the Gap Analysis Matrix, you can create and manage profiles to save commonly-used combinations of Target Architectures, Baseline Architectures and stereotypes.</p> <p>To work on Gap Analysis Matrix profiles, click on the <b>Options</b> button in the top right corner of the matrix; a submenu displays, listing options to:</p> <ul style="list-style-type: none"> <li>Create / Save the current matrix settings and content</li> <li>Update the currently selected profile in the <b>Profile</b> field</li> <li>Delete the currently selected profile in the <b>Profile</b> field</li> </ul>	
<b>Open a saved Profile</b>	<p>The <b>Profile</b> field drop-down list shows all the saved Gap Analysis Matrix profiles.</p>	

Topic	Detail	See also
	Click on the drop-down arrow in the <b>Profile</b> field, find the required Profile in the list and click on it to load the content of the selected profile in the matrix.	

### Notes

- The Gap Analysis Matrix is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect
- You can transport your Gap Analysis Matrix profiles between projects, using the **Export Reference Data** and **Import Reference Data** options

### Learn more

- [Export Reference Data](#)<sup>[376]</sup>
- [Import Reference Data](#)<sup>[380]</sup>

#### 4.3.6.1 Edit Gap Notes

The Gap Analysis Matrix provides a way to record notes during the initial phase or during a discussion session for Gap Analysis. These notes are saved in profiles, which can later be reviewed and the appropriate model elements created to address these gaps.

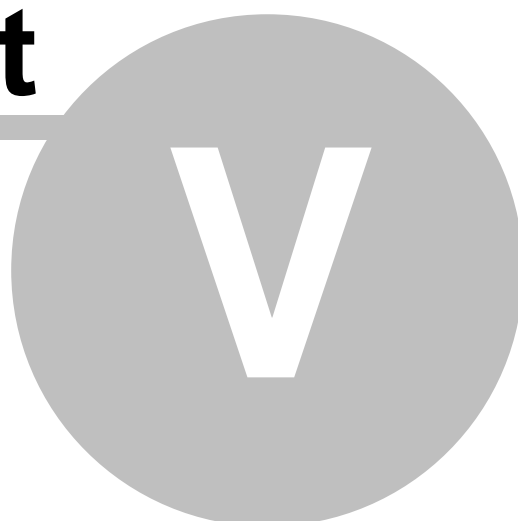
### How to

To edit gap notes

Step	Action	See also
1	Right click on the cell at the intersection of a Target and Baseline element and select the <b>Edit Gap Note</b> context menu option.  Alternatively, double click on the cell.  The Gap Note dialog displays.	<a href="#">Gap Analysis Matrix</a> <sup>[745]</sup>
2	Edit the notes as required and click on the <b>OK</b> button.	
3	The edited notes display in the selected cell in the matrix.	
4	Update or save the matrix to retrieve it when the Gap Matrix profile is loaded at a later stage.	

# Part

---



## 5 Modeling Basics



Modeling can be described as graphically representing a business process or software system. The resulting model can be used to emphasize a certain aspect of the system being represented, and to record, document and communicate its detail. A study of such a model can enable insight or understanding of the system. Enterprise Architect's modeling platform is based on the Unified Modeling Language (UML), a standard that defines rules and notations for specifying business and software systems.

Using Enterprise Architect, you can quickly build a model using a hierarchy of *Packages* to represent the structure and organization of the model.

### Structural Components

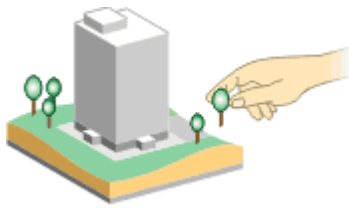
Component	Link
<p>Models - a model is the highest conceptual level, representing a distinct and complete representation of all or some part of a modeled system.</p> <p>A Project can contain multiple models.</p>	<p><a href="#">Models</a> <sup>[753]</sup></p> <p><a href="#">Modeling</a> <sup>[752]</sup></p> <p><a href="#">Model Wizard</a> <sup>[753]</sup></p>
<p>Views are the second level within a model and define a specific viewpoint of the system being modeled - for example a Use Case view, a Requirements View or a Dynamic (behavioral) View.</p> <p>Views are simply Packages which have an additional conceptual meaning.</p>	<p><a href="#">Views</a> <sup>[769]</sup></p> <p><a href="#">Add Views</a> <sup>[770]</sup></p>
<p>Packages are the basic containers that create the overall model structure. Packages hold other Packages, elements, diagrams and similar model constructs.</p>	<p><a href="#">Packages</a> <sup>[772]</sup></p> <p><a href="#">Open Package in the Project Browser</a> <sup>[774]</sup></p>
<p>Diagrams are visual representations of how model elements are connected or behaviorally related. They can also display the characteristics of an element, such as attributes, methods, notes and Tagged Values in a convenient visual style.</p>	<p><a href="#">Diagrams</a> <sup>[778]</sup></p> <p><a href="#">Diagram Toolbox</a> <sup>[792]</sup></p> <p><a href="#">The Quick Linker</a> <sup>[896]</sup></p> <p><a href="#">Diagram Context Menu</a> <sup>[778]</sup></p> <p><a href="#">Diagram Tabs</a> <sup>[790]</sup></p> <p><a href="#">Diagram Tasks</a> <sup>[820]</sup></p> <p><a href="#">Layout Diagrams</a> <sup>[874]</sup></p>
<p>Elements are the basic building blocks of models. They represent both structural constructs such as Classes and Interfaces, as well as behavioral</p>	<p><a href="#">Elements</a> <sup>[900]</sup></p> <p><a href="#">Element Context Menu</a> <sup>[939]</sup></p>

constructs such as Activities, Actions and States.	<a href="#">Visual Representation</a> <sup>[953]</sup>
Connectors are the various kinds of relationships between elements within a model - including behavioral relationships, associations, taxonomic relations and similar.	<a href="#">Connectors</a> <sup>[1102]</sup> <a href="#">Connector Management Options</a> <sup>[1102]</sup> <a href="#">Connector Tasks</a> <sup>[1108]</sup> <a href="#">Connector Properties</a> <sup>[1126]</sup>
Tagged Values are customizable, profile or user defined properties which are generally related to a Stereotype and define additional attributes and characteristics of an element. They are one of the fundamental means of extending UML into more domain specific areas.	<a href="#">Tagged Values</a> <sup>[1134]</sup> <a href="#">Quick Start - Add Tagged Value To Elements</a> <sup>[1136]</sup> <a href="#">Modify Tagged Values</a> <sup>[1138]</sup>
Notes are the standard common language based descriptions of what an element, diagram, feature or relationship is for and how it is used within the model. Notes are often used as a first step to eliciting the meaning and use of an element - which is later refined into something more concrete and precisely specified.	<a href="#">Notes</a> <sup>[1142]</sup> <a href="#">Notes Toolbar</a> <sup>[1143]</sup>
Reference Information - a variety of basic types and information that is used across a particular model - for example, Stereotypes, Tagged Values and similar.	<a href="#">Reference Data</a> <sup>[1146]</sup> <a href="#">UML Types</a> <sup>[1146]</sup>

### Learn more

- For information on UML, see the [Standard UML Models](#) <sup>[1179]</sup> topic
- The [Quick Start Tutorial](#) <sup>[47]</sup> topic briefly shows you how to create a diagram within a Package, containing elements and connectors
- Sparx Systems also provide a [Demonstration of quickly developing a Use Case model](#) (Online Resource)

## 5.1 Modeling



Enterprise Architect is a comprehensive UML analysis and design tool. It provides a library of UML data structures that you can use and extend to develop your models.

You begin to create your projects and models using the **Start Page** or **File** menu, which provide templates on which to base your models. From these you create your Packages and diagrams using a range of toolbars and menus, and you then populate these structures with elements and connectors, using the **Diagram Toolbox**. You can also create new structures through the **Project Browser**, and re-use existing structures using the Project Browser, **Model Views**, **Package Browser** and **Model Search**.

Building models requires the use of various UML data structures and Enterprise Architect tools, as above, to graphically represent a business process or software system; the resulting model can be used to emphasize a certain aspect of the system being represented and to record and communicate its detail.

An extremely useful Enterprise Architect tool is the **Relationship Matrix**, through which you can select, visualize and amend the relationships in the organization of structures within the model. This provides particular support for:

- Requirements Management and
- Modeling the business process, an essential part of any software development process

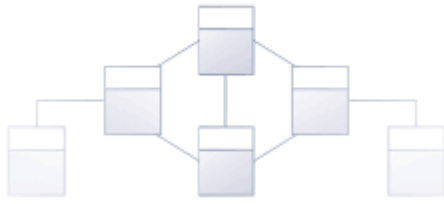
You can extend the scope of your models using **UML Stereotypes**, **Profiles** and **Patterns**, and **MDG Technologies**.

### Learn more

- [The Start Page](#) <sup>[68]</sup>
- [File Menu](#) <sup>[79]</sup>
- [Model Wizard](#) <sup>[753]</sup>
- [Workspace Toolbars](#) <sup>[134]</sup>
- [Main Menu](#) <sup>[77]</sup>
- [Diagram Toolbox](#) <sup>[792]</sup>
- [Project Browser](#) <sup>[646]</sup>
- [Model Views](#) <sup>[686]</sup>
- [Package Browser](#) <sup>[673]</sup>
- [Model Search](#) <sup>[700]</sup>
- [Relationship Matrix](#) <sup>[727]</sup>
- [Modeling Fundamentals](#) <sup>[750]</sup>
- [Defining a Modeling Language](#) <sup>[1483]</sup>
- [Requirements](#) <sup>[1726]</sup>
- [Standard UML Models](#) <sup>[1179]</sup>
- [Extending UML Models](#) <sup>[1477]</sup>



## 5.2 Models



In Enterprise Architect a *model* is a special type of Package, being the top level entry point to a project file. You can develop a project with one model, or with several. Each model is a root node of a hierarchy of Model Packages and Views and, below them, Packages that contain the diagrams, elements, relationships and associated metadata that define the structure and function of a system or process. These components are organized through the Package hierarchy, which helps to group and manage related components; by iterating through all models, you can access all the elements within the project.

You can create the model or models when you first create the project, or you can add and develop new models later. When you first create a project, Enterprise Architect automatically displays the Model Wizard, which you use to generate the required models and subordinate structures from a wide selection of Model Patterns. However, you can also create a model as a single project root node with no content, and build up the model structure yourself, using the menu option:

**Project Browser context menu | Add | Add Model (root node)**

You can also delete a model, but be aware that everything contained in the model is deleted as well.

**Right-click | Delete Project Root**

### Learn more

- [Modeling Basics](#) <sup>750</sup>
- [Model Wizard](#) <sup>753</sup>

### 5.2.1 Model Wizard

The **Model Wizard** provides **Patterns** from a wide range of **technologies**, which you can use to quickly **generate new models** in your project both when you first create the project and over time.

The Model Wizard has three tabs:

- **Model Patterns** - from which you can create basic **models** that you can develop further as components of your project
- **Application Patterns** - from which you can generate starter **projects** including model information, code and build scripts for one of several basic application types
- **VEA Examples** - through which you can **import** complete sample models (Packages), including all necessary model information, code and build scripts, to explore and try out the **Visual Execution Analyzer**

This description covers the use of the **Model Patterns** tab. See *Learn more*, below, for links to information on the other two tabs.

**Access** **Automatically displays when creating a new project; otherwise**  
**Right-click on a Project Browser Package node | Add | Add a New Model using Wizard**

(Ctrl+Shift+M)

Project Browser toolbar | Generate a New Model

Field/Button/ Option	Action	See also
<b>Technology</b>	<p>Lists the technologies integrated with or imported into Enterprise Architect, which provide Model Patterns specific to the technology.</p> <p>If you are a Technology Developer, you can create and import your own MDG Technology and <b>distribute custom Model Patterns</b> from that technology, through the Model Wizard; the name of your technology displays in this list for your users to select to access the technology Patterns.</p> <p>You can also use the Model Wizard to <b>create</b> these MDG Technologies, by selecting the MDG Technology Builder technology. This creates a Model Package containing templates that you can use to generate the technology and a set of Profiles.</p> <p>Click on the required technology to list the associated Model Patterns in the <b>Name</b> panel.</p> <p>The first three technologies support core UML modeling:</p> <ul style="list-style-type: none"> <li>• <b>Basic UML 2 Technology</b> - provides Model Patterns for general purpose, 'diagram type' UML modeling, generating extensive model structures</li> <li>• <b>Simple UML Views</b> - provides Model Patterns for 'view' oriented modeling, generating an appropriate Package with an empty diagram, eliminating the need to delete unwanted sections of boilerplate structure</li> </ul> <p>These are slightly different from the basic Views that you can create instead of Model Packages, under the model root node</p> <ul style="list-style-type: none"> <li>• <b>Core Extensions</b> - provides Model Patterns for the extended application of UML modeling</li> </ul> <p>There is some overlap between these three categories.</p>	<p><a href="#">Using the Profile Helpers</a> <sup>[1528]</sup></p> <p><a href="#">Generate MDG Technology File</a> <sup>[1545]</sup></p> <p><a href="#">Views</a> <sup>[769]</sup></p>
<b>Name</b>	<p>Lists the Model Patterns provided by the selected technology.</p> <p>Click on the check box for each model type you want to create in your project.</p>	
<b>All</b>	<p>Click on this button to select all of the Model Patterns listed in the <b>Name</b> panel.</p>	

Field/Button/ Option	Action	See also
<b>None</b>	Click on this button to clear all Model Patterns selected in the <b>Name</b> panel.	
<b>OK</b>	Click on this button to generate the model structures for your project.  The structures are added to the Project Browser and can be expanded and worked on immediately.	
<b>Cancel</b>	Click on this button to abort the creation of model structures.	
<b>Help</b>	Click on this button to display this Help topic.	

#### Learn more

- [Model Templates](#) <sup>[755]</sup>
- [Application Patterns](#) <sup>[2077]</sup>
- [Visual Execution Analyzer Samples](#) <sup>[2529]</sup>
- [Custom Templates](#) <sup>[1576]</sup>

## 5.2.2 Model Templates

**Model templates** generated by Enterprise Architect are designed to assist in the creation of projects and models for both new and experienced users.

The templates are provided through the **Model Wizard**, for the model types available under the Basic UML 2 technology and Core Extensions technology.

Each template provides a **framework** of Packages, diagrams, representative elements and connectors on which to build a model appropriate to your purpose. These include overview diagrams that provide an introduction to the terminology and icons used in the model templates, and give a quick guide to the UML concepts important to the templates and how they are applied in Enterprise Architect. This information is provided either in Note elements or via hyperlinks to the Enterprise Architect Help and other Sparx Systems website resources.

#### Templates

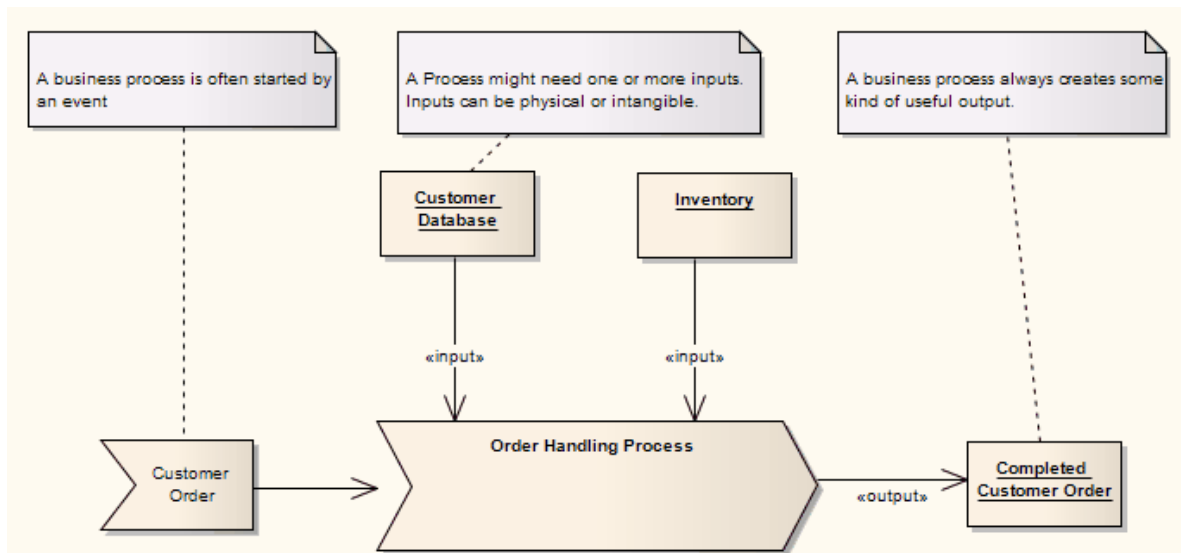
Template	Detail	See also
<b>Business Process Model</b>	The Business Process model describes both the behavior and the information flows within an organization or system.  As a model of business activity, it captures the significant events, inputs, resources, processing and outputs associated with relevant business processes.	<a href="#">The Business Process Model</a>  <a href="#">Business Models</a> <sup>[1804]</sup>  <a href="#">Analysis Diagram</a> <sup>[1801]</sup>  <a href="#">Business Modeling/</a>

Template	Detail	See also
		<a href="#">Interaction</a> <sup>[1805]</sup>
<b>Requirements Model</b>	<p>The Requirements model is a structured catalogue of end-user requirements and the relationships between them.</p> <p>The Requirements Management facilities built into Enterprise Architect can be used to define Requirement elements, connect Requirements to other model elements, connect Requirements into a hierarchy and report on Requirements.</p>	<a href="#">Requirements Management</a> <sup>[1726]</sup> <a href="#">Requirements Management in Enterprise Architect Packages</a> <sup>[772]</sup>
<b>Use Case Model</b>	<p>The Use Case model describes a system's functionality in terms of Use Cases.</p> <p>Each Use Case represents a single repeatable interaction that a user or 'actor' experiences when using the system, emphasizing the user's perspective of the system and interactions.</p>	<a href="#">Use Case</a> <sup>[1352]</sup> <a href="#">Use Case Diagram</a> <sup>[1201]</sup> <a href="#">The Use Case Model</a>
<b>Domain Model</b>	<p>A Domain model is a high-level conceptual model, defining physical and abstract objects in an area of interest to the Project.</p> <p>The Domain model can be used to document relationships between and responsibilities of conceptual classes (that is, classes that represent the concept of a group of things rather than Classes that define a programming object).</p> <p>This model is also useful for defining the terms of a domain.</p> <p>A Domain model shows:</p> <ul style="list-style-type: none"> <li>• The physical and organizational units of the domain; for example, <b>Employee</b> and <b>Flight</b></li> <li>• The relationships between these units; for example, Employee is <b>assigned</b> to Flight</li> <li>• The multiplicity of those relationships; for example, <b>one</b> employee can be assigned to <b>no</b> flights, <b>one</b> flight or <b>many</b> flights (represented by <b>1*</b> at the ends of the relationship)</li> </ul>	<a href="#">Multiplicity</a> <sup>[1130]</sup>
<b>Class Model</b>	<p>The Class model is a rigorous, logical model of the software system under construction.</p> <p>Classes generally have a direct relationship to source code or other software artifacts that can be grouped together into executable components.</p>	<a href="#">Classes</a> <sup>[1363]</sup> <a href="#">Class Diagram</a> <sup>[1184]</sup> <a href="#">The Logical Model</a>
<b>Database Model</b>	<p>The Database model describes the data that must be stored and retrieved as part of the overall system design.</p> <p>Typically, this means relational database models that describe the tables and data in detail and enable generation of DDL scripts to create and set up databases.</p>	<a href="#">Data Models</a> <sup>[1937]</sup> <a href="#">UML Database Modeling</a>

Template	Detail	See also
<b>Component Model</b>	<p>The Component model defines how Classes, Artifacts and other low level elements are collected into high level components, and describes the interfaces and connections between them.</p> <p>Components are compiled software artifacts that work together to provide the required behavior within the operating constraints defined in the Requirements model.</p>	<a href="#">Components</a> <sup>[1370]</sup> <a href="#">Component Diagram</a> <sup>[1194]</sup> <a href="#">The Component Model</a>
<b>Deployment Model</b>	<p>The Deployment model describes how and where a system is to be deployed.</p> <p>Physical machines and processors are represented by Nodes, and the internal construction can be depicted by embedding Nodes or Artifacts.</p> <p>As Artifacts are allocated to Nodes to model the system's deployment and roll out, the allocation is guided by the use of deployment specifications.</p>	<a href="#">Nodes</a> <sup>[1378]</sup> <a href="#">Artifacts</a> <sup>[1358]</sup> <a href="#">Deployment and Roll Out</a> <sup>[192]</sup> <a href="#">Deployment Diagram</a> <sup>[1191]</sup> <a href="#">Compartments - Tagged Values</a> <sup>[955]</sup> <a href="#">The Physical Model</a>
<b>Testing Model</b>	<p>The Testing model describes and maintains a catalogue of tests, test plans and results that are executed against the current model.</p>	<a href="#">Testing</a> <sup>[2604]</sup> <a href="#">Test Case</a> <sup>[2010]</sup> <a href="#">Show Test Script Compartments</a> <sup>[2616]</sup> <a href="#">Testing Support in Enterprise Architect</a>
<b>Maintenance Model</b>	<p>The Maintenance model enables visual representation of issues arising during and after development of a software product. The model can be enhanced with the integration of change elements and testing.</p>	<a href="#">Maintenance</a> <sup>[2619]</sup> <a href="#">Color Code External Requirements</a> <sup>[1776]</sup>
<b>Project Management Model</b>	<p>The Project Management model details the overall project plan, phases, milestones and resource requirements for the current project.</p> <p>Project Managers can use Enterprise Architect to assign resources to elements, measure risk and effort and to estimate project size. Change control and maintenance are also supported.</p>	<a href="#">Project Management</a> <sup>[509]</sup> <a href="#">Project Manager (Online Resource)</a>
<b>User Interface Model</b>	<p>The User Interface Model is a high level, logical mapping of forms, web pages, dialogs and other screens and controls that form part of the proposed system.</p> <p>The template does not automatically include web interface or WIN32 UI design elements, but you can incorporate these in your subsequent development.</p>	<a href="#">User Interface Models</a> <sup>[1990]</sup> <a href="#">Web Stereotypes</a> <sup>[1995]</sup> <a href="#">Win32 UI Technology</a> <sup>[1996]</sup>

### 5.2.2.1 Business Process Model Template

The **Business Process model** describes both the behavior and the information flows within an organization or system. As a model of business activity, it captures the significant events, inputs, resources, processing and outputs associated with relevant business processes.



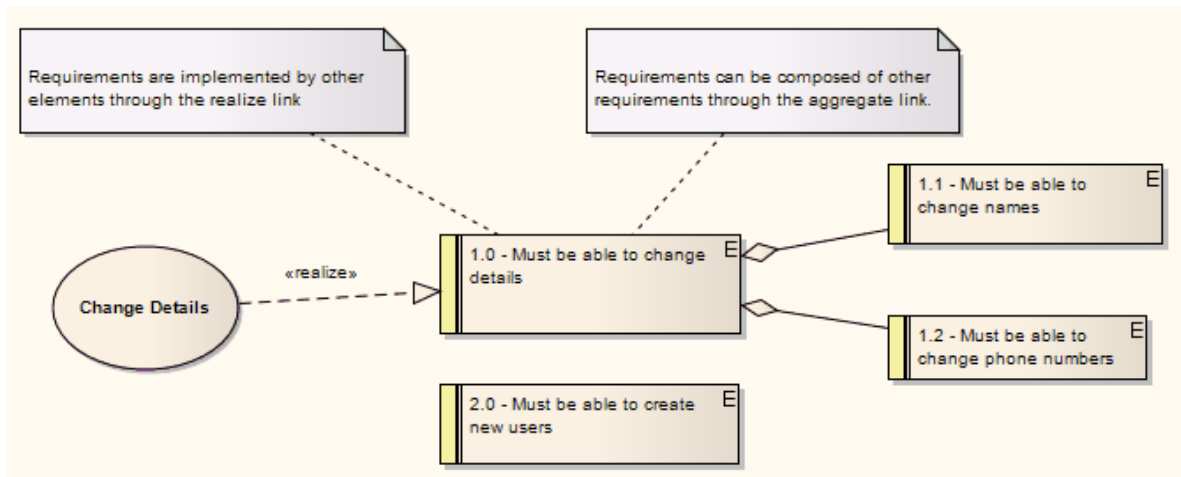
#### Learn more

- [The Business Process Model](#)
- [Business Models](#) <sup>[1804]</sup>
- [Analysis Diagram](#) <sup>[1801]</sup>
- [Business Modeling/Interaction](#) <sup>[1805]</sup>

### 5.2.2.2 Requirements Model Template

The **Requirements model** is a structured catalogue of end-user requirements and the relationships between them.

The **Requirements Management** facilities built into Enterprise Architect can be used to define Requirement elements, connect Requirements to other model elements, connect Requirements into a hierarchy and report on Requirements.



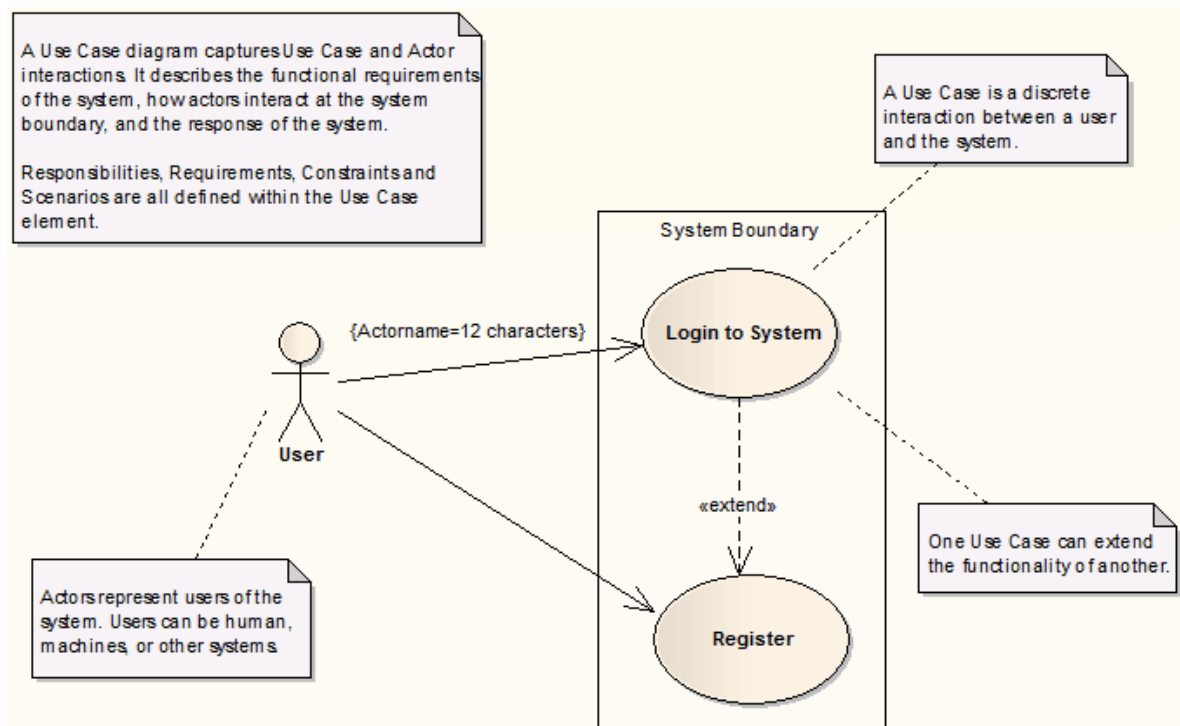
#### Learn more

- [Requirement Models](#) <sup>[1728]</sup>
- [Requirements Management in Enterprise Architect](#)
- [Packages](#) <sup>[772]</sup>

### 5.2.2.3 Use Case Model Template

The **Use Case model** describes a system's functionality in terms of **Use Cases**.

Each Use Case represents a single repeatable interaction that a user or 'actor' experiences when using the system, emphasizing the user's perspective of the system and interactions.



#### Learn more

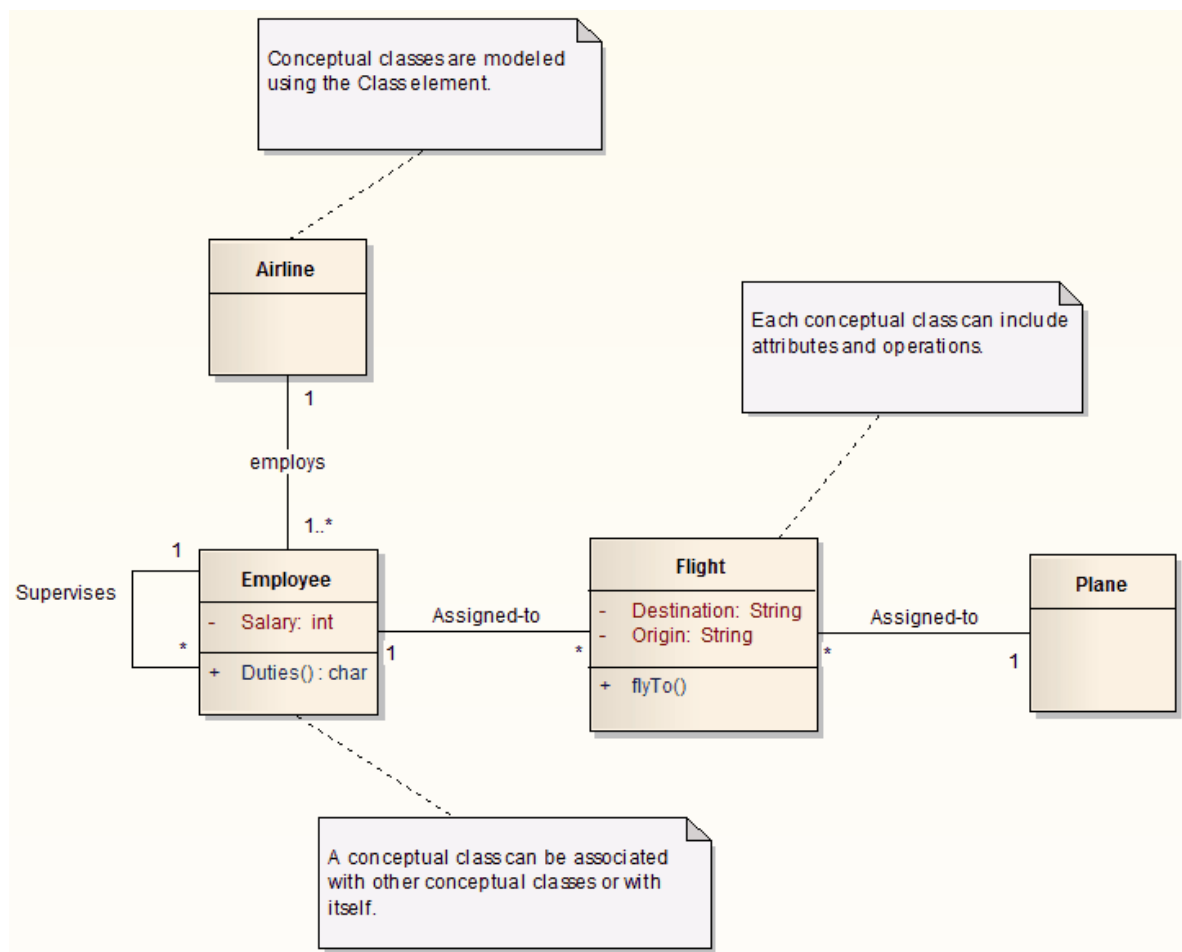
- [Use Case](#) <sup>[1352]</sup>
- [Use Case Diagram](#) <sup>[1201]</sup>
- [The Use Case Model](#)

#### 5.2.2.4 Domain Model Template

A **Domain model** is a high-level conceptual model, defining physical and abstract **objects** in an area of interest to the Project.

The Domain model can be used to document relationships between and responsibilities of conceptual classes (that is, classes that represent the concept of a group of things rather than Classes that define a programming object). It is also useful for defining the terms of a domain.





A Domain model shows:

- The physical and organizational units of the domain; for example, *Employee* and *Flight*
- The relationships between these units; for example, *Employee* is **assigned to** *Flight*
- The multiplicity of those relationships; for example, **one** employee can be assigned to **no** flights, **one** flight or **many** flights (represented by the 1 and the \* at the ends of that relationship)

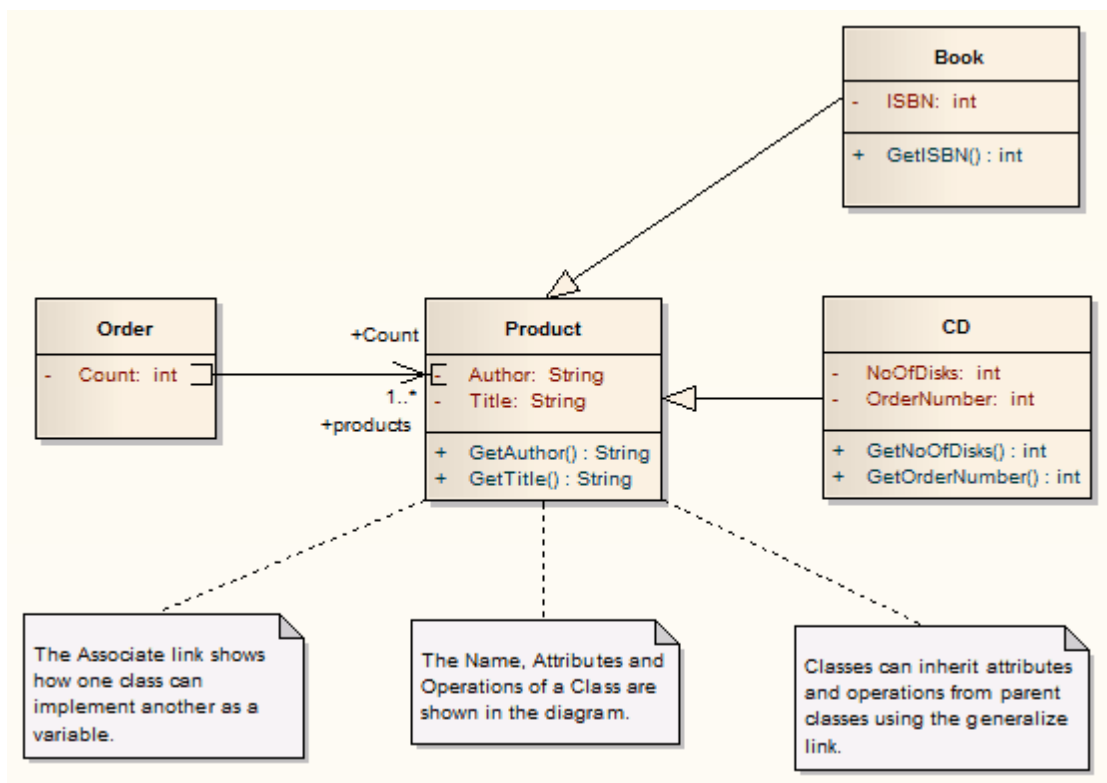
#### Learn more

- [Multiplicity](#) 

### 5.2.2.5 Class Model Template

The **Class model** is a rigorous, logical model of the software system under construction.

**Classes** generally have a direct relationship to **source code** or other software artifacts that can be grouped together into executable components.

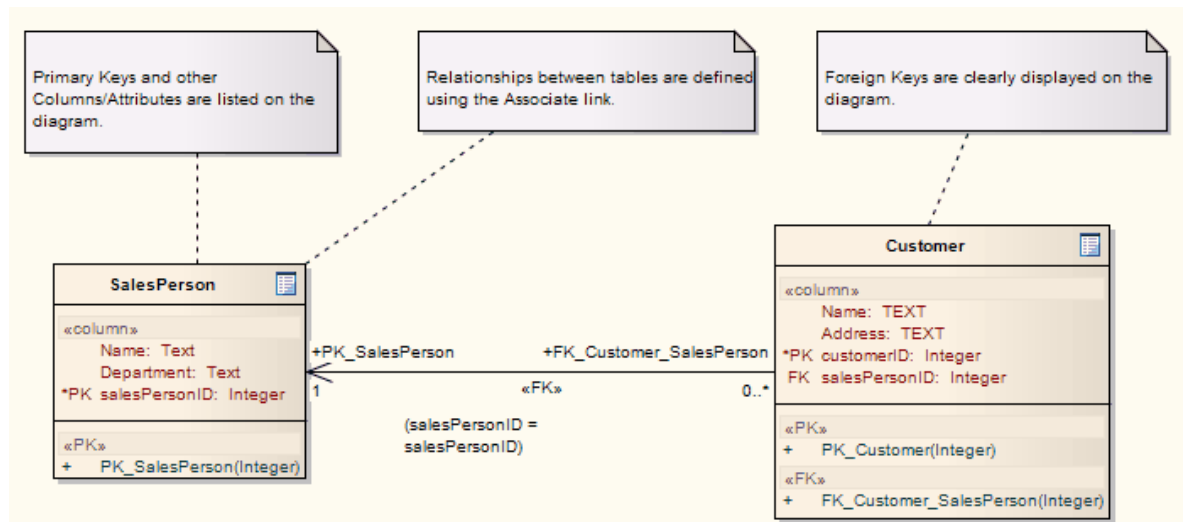


#### Learn more

- [Classes](#) <sup>[1363]</sup>
- [Class Diagram](#) <sup>[1184]</sup>
- [The Logical Model](#)

#### 5.2.2.6 Database Model Template

The **Database model** describes the data that must be stored and retrieved as part of the overall system design. Typically this means relational database models that describe the tables and data in detail and enable generation of DDL scripts to create and set up databases.



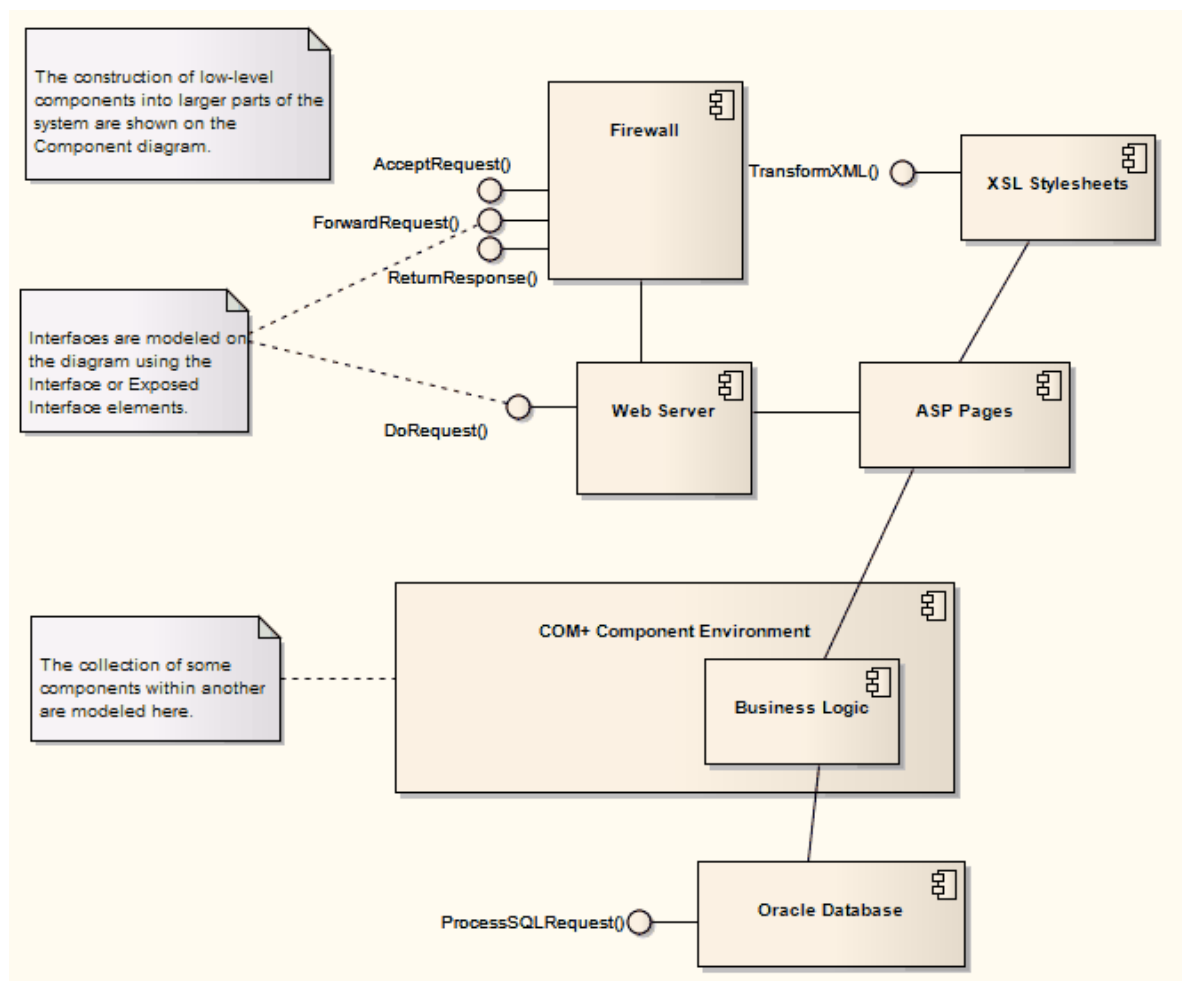
#### Learn more

- [Data Models](#) 1937
- [UML Database Modeling](#)

### 5.2.2.7 Component Model Template

The **Component model** defines how Classes, Artifacts and other low-level elements are collected into high-level components, and describes the interfaces and connections between them.

Components are compiled software artifacts that work together to provide the required behavior within the operating constraints defined in the Requirements model.

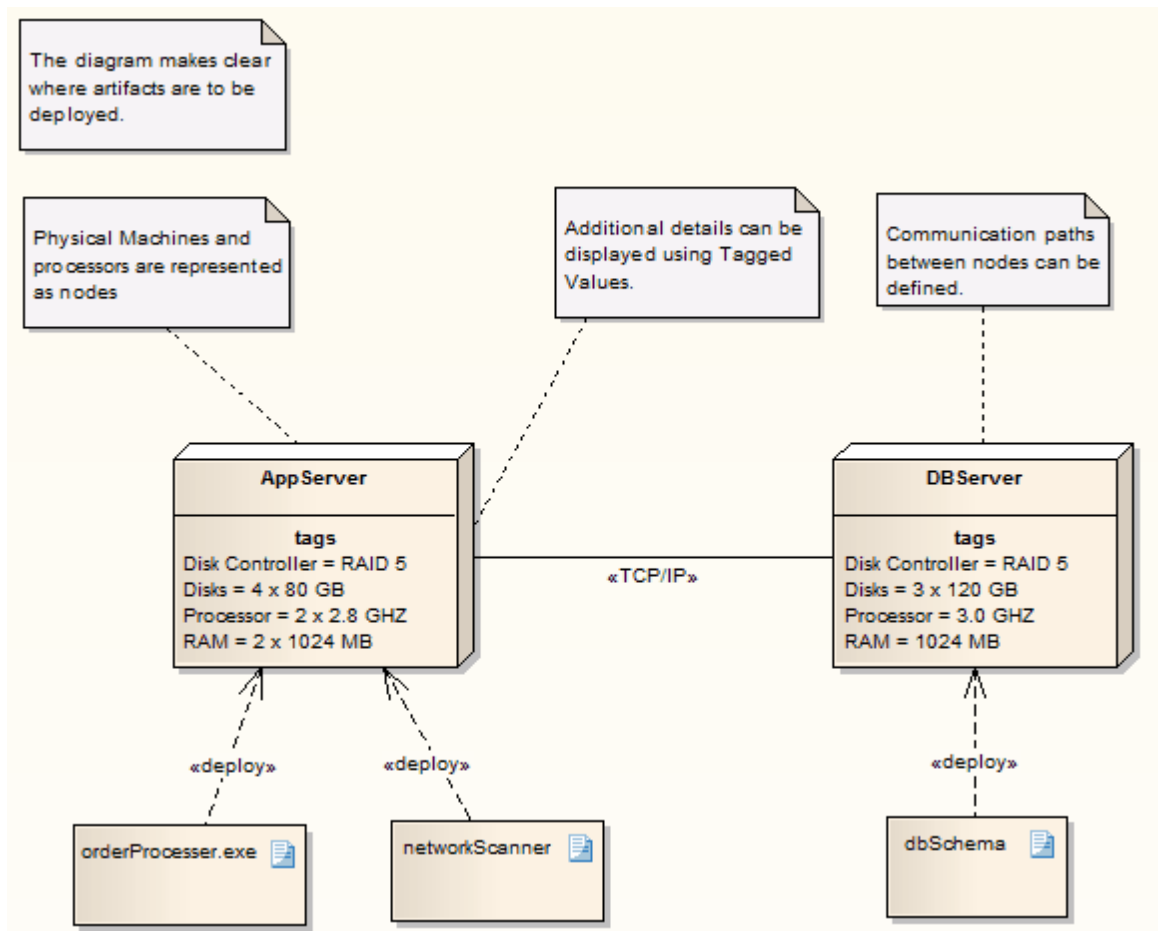


#### Learn more

- [Components](#) <sup>[1370]</sup>
- [Component Diagram](#) <sup>[1194]</sup>
- [The Component Model](#)

#### 5.2.2.8 Deployment Model Template

The **Deployment model** describes how and where a system is to be deployed. Physical machines and processors are represented by Nodes, and the internal construction can be depicted by embedding Nodes or Artifacts. As Artifacts are allocated to Nodes to model the system's deployment and roll out, the allocation is guided by the use of deployment specifications.

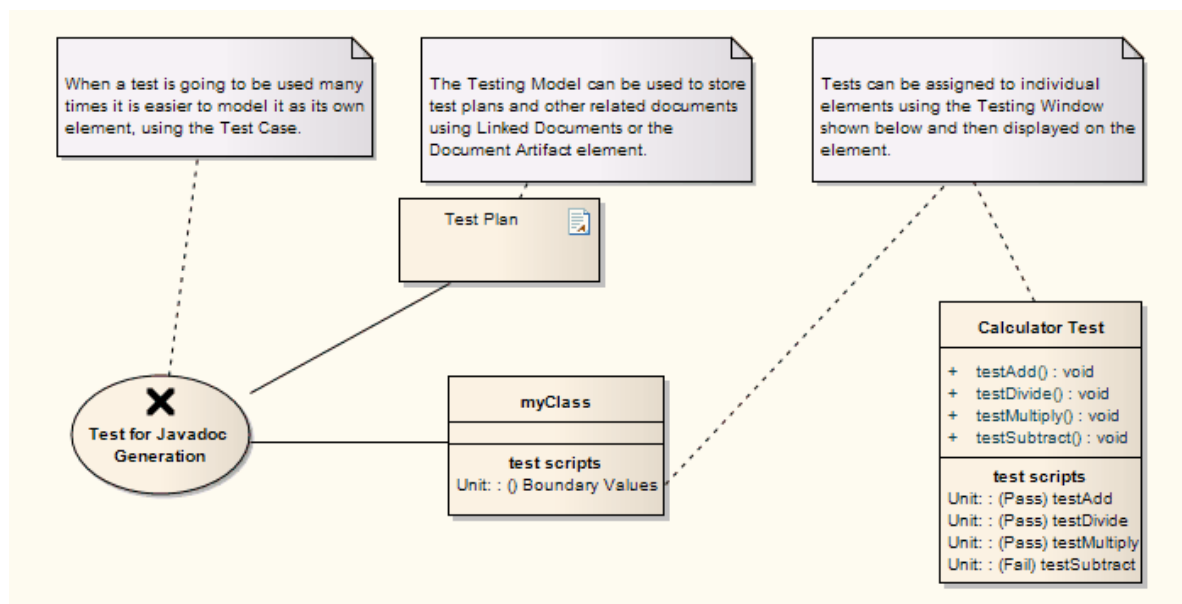


#### Learn more

- [Node](#)<sup>[1378]</sup>
- [Artifact](#)<sup>[1358]</sup>
- [Implementation Managers](#)<sup>[192]</sup>
- [Deployment Diagram](#)<sup>[1191]</sup>
- [Compartments - Tags](#)<sup>[955]</sup>
- [The Physical Model](#)

#### 5.2.2.9 Testing Model Template

The **Test model** describes and maintains a catalogue of tests, test plans and results that are executed against the current model.

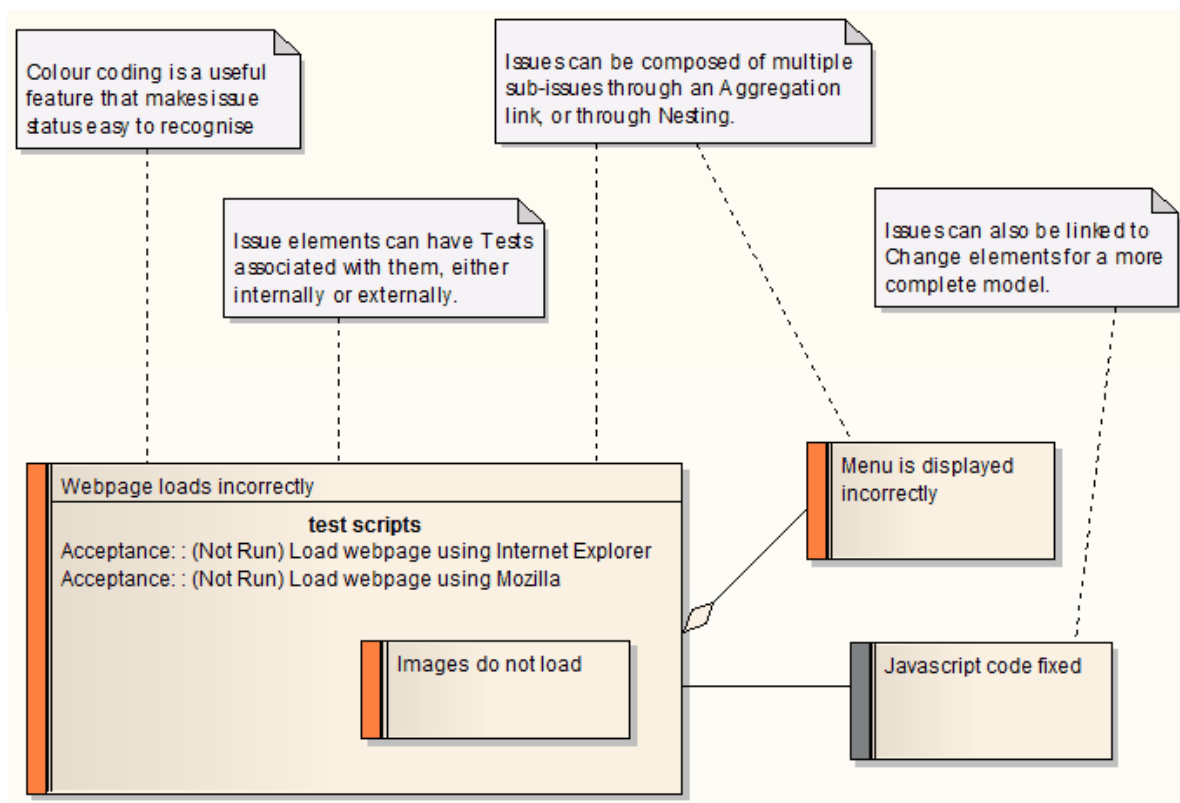


#### Learn more

- [Testing](#) <sup>[2604]</sup>
- [Test Case](#) <sup>[2010]</sup>
- [Show Test Script Compartments](#) <sup>[2616]</sup>
- [Testing Support in Enterprise Architect](#)

#### 5.2.2.10 Maintenance Model Template

The **Maintenance model** is used to create a visual representation of issues arising during and after development of a software product. The model can be enhanced with the integration of Change elements and testing.



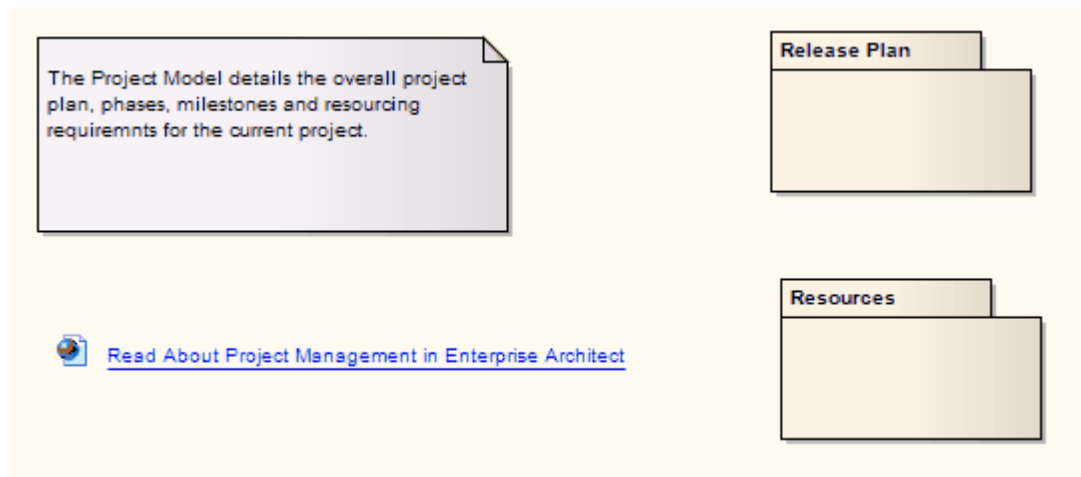
#### Learn more

- [Maintenance](#)<sup>[2619]</sup>
- [Color Code External Requirements](#)<sup>[1776]</sup>

#### 5.2.2.11 Project Model Template

The **Project model** depicts the overall project plan, phases, milestones and resource requirements for the current project.

Project Managers can use Enterprise Architect to assign resources to elements, measure risk and effort and to estimate project size. Change control and maintenance are also supported.



#### Learn more

- [Project Management](#)<sup>[509]</sup>
- [Project Manager](#) (Online Resource)



## 5.3 Views

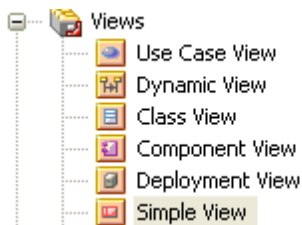
The top-level Packages in a model (below the project root nodes) can be created as **Views**, which are used simply to structurally partition and extend the model according to specific requirements and modeling techniques, such as Component View or Dynamic View. These Views have the same concept as the Simple UML Views generated as Model Packages; however, unlike the Model Package views, basic Views:

- Do not have *any* automatically-generated components
- Can be created only under a root node
- Are top-level Packages with special Package icons
- Are not directly available for Requirements, Analysis or Testing, although you can develop these Views from Simple Views or within Use Case Views (see below)

### Types of View

There are **six** main types of View, each with their own Package icon:

- **Use Case View** - to contain, for example, Use Case diagrams and Analysis diagrams
- **Dynamic View** - to contain, for example, Activity diagrams, Communication diagrams, Sequence diagrams and State Machine diagrams
- **Class View** - to contain, for example, Class diagrams, Code Engineering and Data Models
- **Component View** - to contain, for example, Component diagrams
- **Deployment View** - to contain, for example, Deployment diagrams
- **Simple View** - to customize your own type of view



You can work within the first five standard View types, or devise your own type based on the Simple View. You can create Views, rename them, move them into a different order or delete them, by right-clicking on the selected View and choosing the appropriate context menu option.

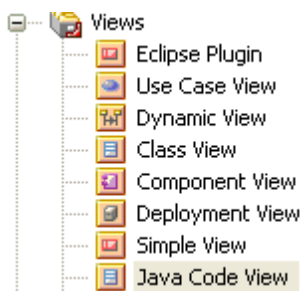
### Learn more

- [Use Case Diagrams](#) <sup>[1201]</sup>
- [Analysis Diagrams](#) <sup>[1801]</sup>
- [Activity Diagrams](#) <sup>[1199]</sup>
- [Communication Diagrams](#) <sup>[1259]</sup>
- [Sequence Diagrams](#) <sup>[1249]</sup>
- [State Machine Diagrams](#) <sup>[1203]</sup>
- [Class Diagrams](#) <sup>[1184]</sup>
- [Code Engineering](#) <sup>[2075]</sup>

- [Data Models](#) <sup>[1937]</sup>
- [Component Diagrams](#) <sup>[1194]</sup>
- [Deployment Diagrams](#) <sup>[1191]</sup>
- [Add Views](#) <sup>[770]</sup>
- [Rename Views](#) <sup>[771]</sup>
- [Delete Views](#) <sup>[771]</sup>

### 5.3.1 Add Views

This example shows a customized view called *Java Code View*, which has been **added** to the end of the Views list.



**Access** [Project Browser root node context menu | Add | New View](#)

#### Create a View

Field/Option/Button	Action	See also
<b>Name</b>	Type the name of the View (in the example, this is <b>Java Code View</b> ).	
<b>Set View Icon Style</b>	Click on the appropriate radio button to set the icon to display next to the View name (the Java Code View has a <b>Class</b> icon).	
<b>Add to Version Control</b>	If the model root node is under version control, this checkbox displays, defaulted to selected.  If you do not want the new View to also be under version control, deselect the checkbox.	<a href="#">Version Control</a> <sup>[383]</sup>
<b>OK</b>	Click on this button to add the View to the Project Browser.	

#### Learn more

- [Views](#) <sup>[769]</sup>
- [Rename Views](#) <sup>[771]</sup>
- [Delete Views](#) <sup>[771]</sup>

### 5.3.2 Rename Views

If you decide to **change the name** of a View, you can do so very easily.

#### Rename a view

Step	Action	See also
1	Right-click on the View in the Project Browser, and select the <b>Properties</b> context menu option.  The Package Properties dialog displays.	
2	In the <b>Name</b> field, type the new name and click on the <b>OK</b> button.	

#### Learn more

- [Views](#) <sup>769</sup>

### 5.3.3 Delete Views

You can use Views as the main divisions of your project, or as temporary holding structures that you **delete** after you have redistributed any content you want to keep.

#### Delete a View

Step	Action	See also
1	In the Project Browser, right-click on the View to delete and select the <b>Delete &lt;viewname&gt;</b> context menu option.  A warning that child elements are to be deleted is displayed.	
2	To delete the View and its contents, click on the <b>Yes</b> button.  To cancel the deletion, click on the <b>No</b> button.	

#### Notes

- If you delete a View, its contents are deleted at the same time; they cannot be restored

#### Learn more

- [Views](#) <sup>769</sup>
- [Move Objects Between Packages](#) <sup>58</sup>

## 5.4 Packages



A **Package** is a container of model components, represented in the Project Browser by a 'folder' icon.

You can perform a number of tasks on Packages, including:

- Open a Package
- Add a Package
- Rename a Package
- Copy a Package
- Drag a Package onto a diagram
- Show or hide a Package
- Delete a Package
- List the elements in a Package, with their properties, in the Package Browser

### Notes

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you need **Update Element** permission to update or delete a Package

### Learn more

- [Open a Package](#)<sup>[845]</sup>
- [Add a Package](#)<sup>[772]</sup>
- [Rename a Package](#)<sup>[774]</sup>
- [Copy a Package](#)<sup>[775]</sup>
- [Drag a Package onto a diagram](#)<sup>[776]</sup>
- [Show or hide Package Contents](#)<sup>[777]</sup>
- [Delete a Package](#)<sup>[777]</sup>
- [Package Browser](#)<sup>[673]</sup>
- [Update Element](#)<sup>[329]</sup>

### 5.4.1 Add a Package

Adding a Package to a model is a simple but versatile process. It is possible to add a new Package to a Root Package, View, Model Package, lower-level Package or element. In the process you can automatically:

- Create a child diagram for the Package
- Open the diagram as soon as it is created, and
- Add the Package to the version control system that its parent component is under

**Access** **Project Browser | Right-click on Package | Add a Package ( Ctrl+W )**

#### Add a Package to the model hierarchy

Field/Option/ Button	Instruction	See also
<b>Name</b>	Type in the name of the new Package.	
<b>Automatically add new diagram</b>	Select this checkbox if you want to immediately create a child diagram for the Package.  Otherwise, leave the checkbox unselected.	
<b>Open new diagram</b>	If you have selected to create a new diagram, select this checkbox if you want to open the diagram as soon as the Package and diagram are created.  Otherwise, leave the checkbox unselected.	
<b>Add to Version Control</b>	If you are adding a Package to a parent Package that is under version control, this option displays with the checkbox selected automatically.  If you want to exclude the new Package from version control, deselect it.	<a href="#">Version Control</a> [383]
<b>OK</b>	Click on this button to create the Package in the Project Browser.  The new Package is inserted into the tree at the current location and, if you selected to create a new diagram, the New Diagram dialog displays. Provide a name for the diagram.  If you have selected to put the Package under version control, the Package Control Options dialog displays. Set up the version control details for the Package	<a href="#">New Diagrams</a> [822]  <a href="#">Configure Packages</a> [488]

#### Notes

- You can also add a new Package element by dragging the **Package** icon from the Toolbox into a diagram; in this case the Package is created under the diagram's owning Package, and is created with a default diagram of the same type as that in which the Package is created
- In a multi-user environment, other users do not see the change until they reload their project

#### Learn more

- [Refresh View/Model](#) [308]

### 5.4.2 Open Package in the Project Browser

When you access a model in Enterprise Architect, you can explore the contents of any Package in the Project Browser by clicking on the **white** arrow to the left of the Package name, to expand the Package structure. If the model has a default diagram (which displays automatically when you access the model) the diagram's parent Package is already expanded.

To display the Package's properties, double-click on the Package name; the element's Properties dialog displays.

#### Notes

- Package contents are arranged alphabetically, and elements can be dragged from one Package to another using the computer mouse
- To collapse the Package structure again, click on the **black** arrow to the left of the Package name in the Project Browser

#### Learn more

- [The Project Browser](#)<sup>[646]</sup>
- [Set the Default Diagram](#)<sup>[844]</sup>
- [Properties Dialog](#)<sup>[956]</sup>

### 5.4.3 Rename a Package

If you want to correct or change a Package name, it is very simple to do so using the Properties dialog for the Package.

**Access**    **Project Browser | Right-click on Package | Properties**

#### Rename a Package

Step	Instruction	See also
1	In the <b>Name</b> field, type the new name.	
2	Click on the <b>OK</b> button.	

#### Notes

- Alternatively, highlight the Package to rename, and press **F2**
- In a multi-user environment, other users do not see the change until they reload their project

#### Learn more

- [Refresh View/Model](#)<sup>[308]</sup>

### 5.4.4 Copy a Package

In developing your model, you might want to duplicate the structure and organization of a section so that, for example, you can replicate that part of the model in another model, or make it easier to cross-refer the contents of duplicate sections. You can quickly do this by copying a complete Package, including its child Packages, elements and diagrams, and then pasting it under:

- The same parent Package
- One or more other Packages in the same model or project, or
- One or more other Packages in another model or project

You would tend to use this procedure for copying sections of a model within the project rather than reproducing an entire model or project, although copying these larger structures is equally feasible.

#### Duplicate a Package

Action	Detail	See also
<b>Copy the Package</b>	<p>In the Project Browser, right-click on the Package and select the <b>Copy / Paste   Copy Package to Clipboard</b> context menu option (or click on the Package and press <b>Ctrl+C</b>).</p> <p>The Copy Package to Clipboard dialog briefly displays until the copy operation completes.</p>	
<b>Paste the Package</b>	<p>In the Project Browser, right-click on the target Package into which to paste the copied Package, and select the <b>Copy / Paste   Paste Package from Clipboard</b> context menu option (or click on the Package and press <b>Ctrl+V</b>).</p> <p>The Paste Package from Clipboard dialog briefly displays until the paste operation completes.</p> <p>In the Project Browser, the target Package is expanded to show the pasted Package; if you are pasting the Package within the same model as the copied source, the source parent Package is also collapsed.</p> <p>If the target Package already contains:</p> <ul style="list-style-type: none"> <li>• A Package with the same name as the pasted Package, the pasted Package name has the suffix - <b>Copy</b></li> <li>• A Package with the same name as the pasted Package <b>including</b> the - <b>Copy</b> suffix, the suffix becomes - <b>Copy1</b> (or - <b>Copy2</b>, - <b>Copy3</b> and so on, as copies of the Package accumulate in the target parent Package)</li> </ul> <p>You can keep the same Package name as the source, or you can rename the Package by either:</p> <ul style="list-style-type: none"> <li>• Clicking twice on it and editing the name in the Project Browser, or</li> <li>• Double-clicking on it and editing the name in the Properties dialog</li> </ul>	

**Notes**

- A copy of a Package does not have the external cross references of the source Package; that is, the following connectors are discarded:
  - Connectors coming from Packages and elements outside the Package being copied, into the Package being copied
  - Connectors going to Packages and elements outside the Package being copied, from the Package being copied
- You cannot paste a Package into a parent Package that is locked by another user or that is checked in; the **Paste...** option is grayed out in the context menu
- This procedure is effectively the same as exporting and importing the Package XML file, with the **Strip GUIDs** checkbox selected
- The paste operation can be a single transaction or a series of transactions to paste each data item separately, depending on the setting of the **Import using single transaction** checkbox in the XML Specifications page of the Options dialog
- You can also **move** a Package to a **different part of the model**, in the same way as moving an element

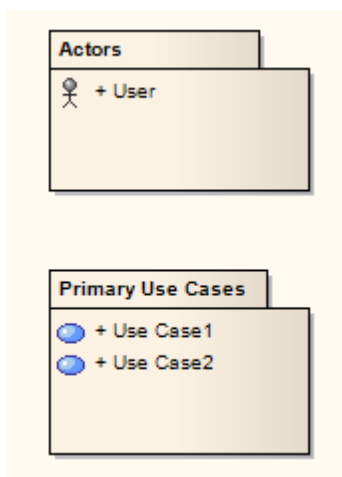
**Learn more**

- [Package XML files](#) <sup>[506]</sup>
- [Copy Packages Between Projects](#) <sup>[506]</sup>
- [XML Specifications](#) <sup>[639]</sup>
- [Locking Packages](#) <sup>[337]</sup>
- [Checking In/Checking Out Packages](#) <sup>[422]</sup>
- [Move Elements Between Packages](#) <sup>[913]</sup>

**5.4.5 Drag a Package Onto a Diagram**

You can drag a Package element from the Project Browser onto the current diagram, to show an icon of the Package containing a list of the Package contents. This is a useful feature to help organize the display and documentation of models.

In this illustration, note the child Actor and Use Case icons.





Learn more

- [Show or Hide Package Contents](#)<sup>[777]</sup>

### 5.4.5.1 Show or Hide Package Contents

If you have Package elements displayed in an open diagram, you can toggle between showing the Package contents within the elements, or leaving them clear.

**Access**    **Diagram | Properties > Elements**

Step	Instruction	See also
1	Select the <b>Package Contents</b> checkbox to list the content of each Package within its element.  Clear the <b>Package Contents</b> checkbox to hide the content of each Package within its element.	
2	Click on the <b>OK</b> button.	

### 5.4.6 Delete a Package

If you do not want to keep a particular Package in the model, it is very simple to remove it.

**Access**    **Project Browser | Right-click on Package | Delete '<package name>'**

A prompt displays to confirm the deletion; click on the **Yes** button.

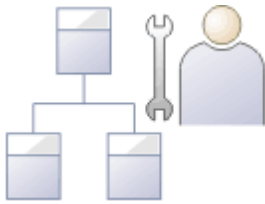
Notes

- Alternatively, click on the Package name and press **Ctrl+Delete**
- Deleting a Package also deletes all contents of the Package, including sub-Packages and elements; make very sure that you really want to do this before proceeding
- In a multi-user environment, other users do not see the change until they reload their project

Learn more

- [Refresh View/Model](#)<sup>[308]</sup>

## 5.5 Diagrams



**Diagrams** are collections of project elements laid out and inter-connected to depict a structure or process.

Enterprise Architect supports all of the UML-defined diagrams, as well as several custom extensions. Together with the Enterprise Architect elements and connectors, these form the basis of the model. Diagrams are stored in Packages and can have a parent object (optional). Diagrams can be moved from Package to Package.

### Notes

- If the diagram display is too small to read comfortably, click on the diagram, press and hold ( **Ctrl** ) and use the mouse wheel to temporarily expand or reduce the display magnification

### Learn more

- [Diagram Context Menu](#) <sup>[778]</sup>
- [Diagram Tasks](#) <sup>[820]</sup>

### 5.5.1 Diagram Context Menu

You can perform a wide range of tasks and operations on a diagram and its contents. To access and initiate these operations, you can select options from the **diagram context menu**.

**Access**    **Open diagram | Right-click on diagram background**

### Options

Option	Action	Shortcut	See also
<b>Extensions</b>	<p>Select this option to access a submenu of enabled Technologies on the system (for example, ArcGIS or TOGAF), each of which provides options for performing technology-specific operations on the diagram.</p> <p>This option is available in the Professional, Corporate and Extended editions of Enterprise Architect.</p>		<a href="#">Domain Based Models</a> <sup>[1789]</sup>
<b>Scripts</b>	<p>Select this option to list the <b>scripts</b> available for execution directly from the diagram.</p> <p>The option is not listed if there are no diagram scripts available.</p>		<a href="#">Script Group Properties</a> <sup>[2794]</sup>

Option	Action	Shortcut	See also
<b>Properties</b>	Select this option to display and/or edit the diagram <b>properties</b> , using the Diagram Properties dialog.		<a href="#">Diagram Properties</a> <sup>[823]</sup>
<b>New Element or Connector</b>	Select this option to <b>add an element or connector</b> to the diagram, by selection from a context-sensitive list.		<a href="#">Insert Elements &amp; Connectors</a> <sup>[782]</sup>
<b>Insert Other Element</b>	Select this option to <b>locate an existing element</b> using the Select Element dialog, and insert it into the diagram as a linked element.		<a href="#">Select &lt;Item&gt; Dialog</a> <sup>[994]</sup>
<b>Synchronize Structural Elements</b>	Select this option to show, on a SysML <b>Internal Block diagram</b> , all structural elements (such as Ports and Parts) relating to the Block that owns this diagram.  This command will also generate Properties defined by existing Association connectors.		<a href="#">Generate Properties From Block Associations</a> <sup>[2304]</sup>
<b>Hide Diagram Frame</b> <b>Show Diagram Frame</b>	On a SysML <b>Internal Block diagram</b> , the elements are automatically enclosed in a <b>Diagram Frame</b> having the name of the Block element that owns the diagram.  Click on these options to hide and redisplay this Diagram Frame.		
<b>Advanced</b>	Select this option to access the options for more advanced operations, such as comparing the diagram to a baseline or creating a working set artifact.		<a href="#">Advanced Diagram Submenu</a> <sup>[783]</sup>
<b>Paste Element(s) as Link</b> <b>Paste Element(s) as New</b>	Select this option to <b>paste copied element(s)</b> into the diagram as a link or as new elements.		<a href="#">Paste copied element</a> <sup>[836]</sup>
<b>Paste Image from Clipboard</b>	Select this option to <b>paste an image</b> held on the clipboard into the diagram.	<b>Ctrl+Shift+ Insert</b>	<a href="#">Paste an image</a> <sup>[82]</sup>
<b>Switch to List View</b>	Select this option to display the diagram contents as a <b>Diagram List</b> instead of as a pictorial diagram.		<a href="#">Diagram List</a> <sup>[684]</sup>
<b>Switch to Gantt</b>	Select this option to display the diagram contents		<a href="#">The Gantt View</a> <sup>[594]</sup>

Option	Action	Shortcut	See also
<b>View</b>	as a <b>Gantt chart</b> instead of as a pictorial diagram.		
<b>Statechart Editor</b>	For a State Machine diagram, select this option to switch the current format of the diagram to: <ul style="list-style-type: none"> <li>• <b>Diagram</b></li> <li>• <b>Table (State-Next State)</b></li> <li>• <b>Table (State-Trigger)</b> or</li> <li>• <b>Table (Trigger-State)</b></li> </ul>		<a href="#">State Machine Table</a> <sup>[1212]</sup>
<b>Swimlanes, Matrix and Kanban</b>	Select this option to add or modify <b>Swimlanes</b> or a <b>Swimlanes Matrix</b> on the diagram, or apply the <b>Kanban</b> facilities.		<a href="#">Swimlanes</a> <sup>[852]</sup> <a href="#">Swimlanes Matrix</a> <sup>[854]</sup> <a href="#">Kanban Facilities</a> <sup>[857]</sup>
<b>Context Filtering</b>	Select this option to toggle <b>context filtering</b> within the diagram view. By default, this will highlight the currently selected element and related elements within the diagram while making all other elements appear faded into the background.  Context Filtering by name or alias can also be applied using the Format Toolbar.		<a href="#">Context Filter a Diagram</a> <sup>[789]</sup>  <a href="#">Format Toolbar</a> <sup>[785]</sup>
<b>Make all Elements Selectable</b>	Select this option to make all the elements on the diagram <b>selectable</b> .  If an element is: <ul style="list-style-type: none"> <li>• Selectable, you can move it around the diagram and perform right-click context-menu operations</li> <li>• Unselectable, you cannot move it around the diagram and the only right-click operation available is to make the element selectable</li> </ul> This option has no effect on double-click operations on the element, such as displaying child diagrams.		
<b>Modify Element Z Order</b>	Select this option to <b>re-sequence overlapping objects</b> in the diagram.  This option both: <ul style="list-style-type: none"> <li>• Numbers the objects in the diagram by Z-order, in red, and</li> </ul>		<a href="#">Z Order Elements</a> <sup>[844]</sup>

Option	Action	Shortcut	See also
	<ul style="list-style-type: none"> <li>Lists the objects by Z-order in the <b>Modify Z Order</b> dialog</li> </ul> <p>In the dialog, you select the required object and move it up or down in the Z-order using the <b>Up</b> and <b>Down</b> arrow keys, each click on an arrow key changes the Z-order by 1.</p> <p>When you select the object in the list, that object is also selected on the diagram and its Z-order number is shown in blue.</p> <p>Click on the <b>OK</b> button to save your changes.</p> <p>On the diagram, you can move the focus from element to element according to Z Order, by pressing <b>Shift+Tab</b>. Each time you press these keys, the element with the next-lowest Z Order number is selected.</p>		
<b>Lock Diagram</b>	<p>Select this option to lock the <b>diagram</b> to protect it from inadvertent changes.</p> <p>This does not apply in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions if security is enabled, in which case you lock the model <b>elements</b>.</p>		<a href="#">Lock a Diagram</a> <sup>[872]</sup>  <a href="#">Lock Model Elements</a> <sup>[336]</sup>
<b>Find in Project Browser</b>	Select this option to <b>locate</b> the current diagram in the <b>Project Browser</b> .	<b>Shift+Alt+G</b>	
<b>Import from source file(s)</b>	Select this option to import, or <b>reverse engineer, source code</b> (the option is not available in the Desktop edition).		<a href="#">Import, or reverse engineer, source code</a> <sup>[2138]</sup>
<b>Import DB schema from ODBC</b>	Select this option to import <b>database tables</b> from an ODBC data source (the option is not available in the Desktop edition).		<a href="#">Import database tables from an ODBC data source</a> <sup>[2376]</sup>
<b>Save Current Changes</b>	Select this option to <b>save any changes</b> to the current diagram.	<b>Ctrl+S</b>	
<b>Execute Simulation</b>	<p>Select this option to list and select from three further options for performing a model simulation on a behavioral diagram:</p> <ul style="list-style-type: none"> <li><b>Interpreted simulation</b> - Perform dynamic execution of a simulation (Corporate and Extended Editions)</li> <li><b>Manual simulation</b> - Step through a</li> </ul>		<a href="#">Model Simulation</a> <sup>[2463]</sup> <a href="#">How It Works</a> <sup>[2466]</sup>  <a href="#">Set Up Simulation</a>

Option	Action	Shortcut	See also
	simulation manually (the only option available in the Professional edition) <ul style="list-style-type: none"> <li>• <b>Using Script</b> - If simulation scripts exist for the diagram, list the scripts available; if no scripts exist, the option does not display</li> </ul>		<a href="#">Script</a> <sup>[2472]</sup>
<b>Help</b>	Select this option to view the Help on the this type of diagram.		

#### Notes

- Not all the menu options described appear on the context menu of every diagram

### 5.5.1.1 *Insert Elements and Connectors*

When you click on the **Diagram | New Element or Connector** context menu option, a list of elements and connectors displays. This is the **Toolbox Shortcut** menu.

#### Structure

The Toolbox Shortcut menu is divided into sections:

- **Other** - expands to offer options to select elements and connectors from diagram types other than either the current diagram type or pinned Toolbox pages
- The expanded list of elements and connectors for the current diagram type
- Collapsed lists of elements and connectors for pages that have been pinned in the Toolbox; if an MDG Technology:
  - is active
  - automatically pins Toolbox pages, and
  - has pages that redefine UML or Extended pages

The MDG Technology pages override the UML or Extended pages, which are not shown

- At the end:
  - **Common** - expands to display a list of the common elements and connectors, and
  - **Artifact** - Expands to display a list of the different types of Artifact element you can create

#### Notes

- The list of elements and connectors also displays on the diagram if you press **Ctrl** or **Shift** as you right-click the mouse, or if you press **Spacebar** or **Insert** while the cursor is on the diagram

#### Learn more

- [Toolbox Shortcut Menu](#)<sup>[799]</sup>

- [Common Page](#)<sup>[800]</sup>
- [Artifact](#)<sup>[1358]</sup>

### 5.5.1.2 Advanced Diagram Submenu

You can perform a number of advanced tasks and operations on a diagram and its contents. To access and initiate these operations, you can select options from the **Advanced diagram submenu**.

**Access**    **Open diagram | Right-click on diagram background | Advanced**

#### Options

Option	Action	Shortcut	See also
<b>Compare Diagram to Baseline</b>	Select this option to perform a comparison of the diagram <b>layout</b> against a version from a previously saved baseline.  This option can be used to roll back individual changes made to the diagram since the baseline was created.		<a href="#">Check Visual Changes to Diagrams</a> <sup>[466]</sup> <a href="#">Manage Baselines</a> <sup>[467]</sup>
<b>Save as Profile</b>	Select this option to save the current diagram <b>as a Profile</b> .		<a href="#">Export a Profile</a> <sup>[1523]</sup>
<b>Create Working Set Artifact</b>	Select this option to create a Working Set Artifact on the diagram, to define a Working Set that includes an instruction to open the diagram.		<a href="#">Artifact</a> <sup>[1358]</sup> <a href="#">Working Sets</a> <sup>[567]</sup>
<b>Set Theme</b>	Select this option to set the <b>Theme</b> to apply to all diagrams that you display in the model.  The Options dialog displays, at the <b>Diagram   Themes</b> page.		<a href="#">Diagram Theme Options</a> <sup>[617]</sup>

#### Learn more

- [Diagram Context Menu](#)<sup>[778]</sup>






### 5.5.2 Print Preview

Before you **print** an Enterprise Architect diagram or document, you can display a **preview** of its output appearance.

**Access**    **File | Print Preview**

#### Notes

- The display initially shows the first two pages on one screen, with no scroll bar

- To toggle between the two-page display and a single-page display, click on the  button in the preview screen toolbar; in either mode, you can use the  and  buttons in the toolbar to scroll through the pages of the diagram or document
- To display more than two pages on one screen, up to a maximum of ten pages, click on the  button in the Preview screen toolbar; the screen now includes the vertical scroll bar, which you can also use to scroll through the pages of the diagram
- Having checked the preview, you can immediately print the diagram or document by clicking on the  **Print** button; otherwise click on the **Close** button

### 5.5.3 Diagram View

If you have created diagrams in the Project Browser, you can select to open them and then add elements and connectors to them within the **Diagram View**. This is the main workspace for developing the structure of your model components, dragging new objects onto the diagram from the Diagram Toolbox and existing objects from the Project Browser. You can also add elements and connectors using the Quick Linker and other context menus within the Diagram View, and move, format and edit them.

Most work on elements and connectors is carried out in the Diagram View, so understanding how it works and how to manipulate elements is essential. You can use the **EAExample** project supplied with the system, or your own 'sandpit' models, to explore the capabilities and behavior of the Diagram View.

**Access**    **Project Browser | double-click diagram name**

#### Diagram sections

Section	Description	See also
<b>Caption Bar</b>	<p>Across the top of a diagram is the diagram caption bar, which includes the following:</p> <ul style="list-style-type: none"> <li>• Icon and text label for the diagram type</li> <li>• Diagram name</li> <li>• A 'chevron' icon to display or hide the Format Toolbar (see below)</li> <li>• A drop-down arrow that lists the currently-open diagrams; click on: <ul style="list-style-type: none"> <li>• A diagram name to switch to that diagram</li> <li>• Options to redisplay the hidden Start Page, Model Default diagram (if defined) or User Default diagram (if defined)</li> </ul> </li> <li>• The window 'close cross'; click on this to close the displayed diagram</li> </ul>	<p><a href="#">The Start Page</a><sup>[68]</sup></p> <p><a href="#">Diagram Advanced Menu</a><sup>[95]</sup></p>
<b>Format Toolbar</b>	<p>Underneath the Caption Bar is the Format Toolbar, which you can use to apply a wide range of functions to the format and appearance of the elements in your diagram.</p> <p>You can hide or reveal this toolbar by clicking on the 'chevron' icon in the Caption Bar (third icon from the right).</p>	<p><a href="#">Format Toolbar</a><sup>[785]</sup></p>



Section	Description	See also
<b>Diagram View</b>	<p>The larger area underneath the Caption bar is the work area within which you can create and view the diagram structures. The borders of the printable diagram page are represented by pale broken lines; structures created beyond these borders will print on separate pages.</p> <p>The Diagram View area is also where the diagram and object context menus operate, and where the representation of the model structure can be changed to a Gantt chart or diagram list format.</p>	<p><a href="#">The Gantt View</a><sup>[594]</sup></p> <p><a href="#">Diagram List</a><sup>[684]</sup></p>

### Notes

- Diagrams can contain hyperlinks to other diagrams, which are opened when you click on the links
- It is possible to open several diagrams at once, but **within the Diagram View** you can only **view** one at a time
- You **can** view and work on several diagrams at once, by making the tab containing a diagram into a **floating window**, using the **Window** menu
- You can also work on elements using the **Diagram List**
- Additional information on the diagram, or a selected element within the diagram, is displayed in the system **Status bar**

### Learn more

- [Diagram Tabs](#)<sup>[790]</sup>
- [Window Menu](#)<sup>[121]</sup>
- [The Project Browser](#)<sup>[646]</sup>
- [Diagram List](#)<sup>[684]</sup>
- [Status Bar](#)<sup>[150]</sup>

#### 5.5.3.1 Format Toolbar

As you add elements and connectors to a diagram, you might want to group or differentiate these objects by adjusting their appearance or position on the diagram, or by filtering the diagram to show only certain elements. You can make a number of changes to an element's appearance and then capture these changes as a named style, which you can then apply to other elements on the same diagram or on a different diagram. All of these options are available through the diagram **Format Toolbar**, which automatically displays at the top of every diagram.



You can hide or recover this toolbar by clicking on the or icons in the **Caption Bar** at the top of the Diagram View.

You can also temporarily display an abbreviated Format Toolbar beside an element or connector. For:

- Elements, select one or more elements on the diagram and click on the paintbrush icon beside the element (and, for multiple selected elements, the icon)
- Connectors, right-click on the connector; the Format Toolbar appears above the context menu

Options

Option	Action	Shortcut	See also
<b>Set Font</b>	Select this icon to set the text font, style, size and effects for the selected element(s) using the Font dialog.		<a href="#">Set Element Font</a> <sup>[949]</sup>
<b>Text Color</b>	Click on the drop-down arrow to set the text color for the selected element(s).  This option overrides the color setting of the <b>Set Font</b> option.		
<b>Fill Color</b>	Click on the drop-down arrow to set the fill color of the selected element(s).  This setting takes precedence over all other fill color settings, such as default appearance, Template element appearance, stereotype default color and Shape Script color.  If you set the fill color to <b>Default</b> and the element has a stereotype with a fill color defined, the element takes the stereotype default rather than the Standard Color default.		<a href="#">Stereotype Settings</a> <sup>[1147]</sup>  <a href="#">Standard Colors</a> <sup>[616]</sup>
<b>Line Color</b>	Click on the drop-down arrow to set the line color of the selected connector or the border of the selected element.  This setting overrides any line color set through the element Default Appearance or Connector Appearance dialogs.		
<b>Line Width</b>	Use the 'spinner' arrows to set the line width of the selected connector or the border of the selected element; choose a value between <b>1</b> (thinnest) and <b>5</b> (thickest).		
<b>Apply Style</b>	Click on this icon to apply a style to the selected elements or connector.  The style is either: <ul style="list-style-type: none"> <li>• Captured using the <b>Get Style</b> option (see below) or</li> <li>• The style currently displayed in the <b>Style List</b> field (see below)</li> </ul>		
<b>Get Style</b>	Select this icon to capture the current style of the selected element or connector.		
<b>Style List</b>	Click on the drop-down arrow and select a style from the list, to immediately apply to the selected elements or		

	connector.  To save a style to this list, use the <b>Save Style</b> option (below).		
<b>Save Style</b>	<p>Having selected an element or connector, click on the down arrow and select the appropriate option:</p> <ul style="list-style-type: none"> <li>• <b>Save as New Style</b> - save the style of the element or connector to the <b>Style List</b> as a new style; type the name of the style when prompted to do so</li> <li>• <b>Save Style</b> - save the (updated) style of the selected element or connector under its current style name in the <b>Style List</b></li> <li>• <b>Refresh List</b> - refresh the <b>Style List</b> with any changes to the existing styles</li> <li>• <b>Delete Current Style</b> - delete the defined style of the selected element or connector from the <b>Style List</b></li> </ul>		
<b>Align Left</b>	<p>(Activated when multiple elements are selected.)</p> <p>Click on this icon to align the selected elements on the left border of the last-selected element.</p>	<b>Ctrl+Alt+Left Arrow</b>	
<b>Align Right</b>	<p>(Activated when multiple elements are selected.)</p> <p>Click on this icon to align the selected elements on the right border border of the last-selected element.</p>	<b>Ctrl+Alt+Right Arrow</b>	
<b>Align Tops</b>	<p>(Activated when multiple elements are selected.)</p> <p>Click on this icon to align the selected elements on the top edge of the last-selected element.</p>	<b>Ctrl+Alt+Up Arrow</b>	
<b>Align Bottoms</b>	<p>(Activated when multiple elements are selected.)</p> <p>Click on this icon to align the selected elements on the bottom edge of the last-selected element.</p>	<b>Ctrl+Alt+Down Arrow</b>	
<b>Bring to Top</b>	Click on this icon to move the selected element(s) in front of other elements; that is, to the top of the Z order.		<a href="#">Diagram Context Menu</a> [778]
<b>Send To Back</b>	Click on this icon to move the selected element(s) behind other elements; that is, to the bottom of the Z order.		
<b>Layout Diagram</b>	<p>Click on the drop-down arrow and select a layout to auto-layout the diagram in that arrangement.</p> <p>You can also filter a standard layout to arrange the elements according to a selected characteristic. To do this, click on the <b>Configure Layouts</b> option in the drop-down list.</p>		<a href="#">Layout Diagrams</a> [874]

<b>Diagram Properties</b>	Click on this icon to display the Diagram Properties dialog for the diagram.	<b>F5</b>	<a href="#">Set Diagram Properties</a> [823]
<b>Delete Selected</b>	Click on this icon to delete the selected element(s).	<b>Ctrl+D</b>	
<b>Filter diagram elements by Name</b>  <b>Context Filter</b>	<p>Use this field and icon to filter the elements in the diagram simply according to name or alias. (A more comprehensive diagram filtering facility is also available, through the Diagram Filters window.)</p> <p>To activate the field, click on the <b>Context Filter</b> icon. If this icon is not selected, the field remains grayed out. If you:</p> <ul style="list-style-type: none"> <li>Do not type any text in the field, you can filter the diagram to show all elements related to a selected element</li> <li>Type text, the diagram is filtered according to the text and settings as described below</li> </ul> <p>Type the first few characters or all of the text string that forms the name of the elements to show normally. All elements that do not have names that match this string are grayed out.</p> <p>For example, if you type <b>REQ</b>, all elements with a name beginning with REQ are visible, whilst all other elements are grayed out.</p> <p>By typing in the full name of an element, you can mute everything on the diagram except that one element, thereby showing exactly where it is on the diagram.</p> <p>The filter <b>also</b> searches on aliases if the <b>Use Alias if Available</b> option is set on the diagram properties Diagram tab. If the <b>Alias only</b> option is set on the Options &gt; Diagram Behavior page, the filter is on the alias text <b>instead of</b> the name, and if the <b>Alias and Name</b> option is set, the filter shows elements that have the text string in <b>either</b> the alias <b>or</b> name.</p>		<a href="#">Diagram Filters</a> [718]  <a href="#">Context Filter a Diagram</a> [789]  <a href="#">Configure Diagram Display</a> [825]  <a href="#">(Diagram) Behavior</a> [625]
<b>Add or Remove Buttons</b>	<p>Click on the drop-down arrow at the far right of the toolbar to:</p> <ul style="list-style-type: none"> <li>Display or hide icons currently defined for the toolbar, or</li> <li>Customize the toolbar to include other icons, to initiate commands found on other toolbars or totally new commands</li> </ul> <p>(As an example, ten options on the Format toolbar are also found on the Diagram toolbar, which contains other options that could be duplicated here).</p>		<a href="#">The Customize Dialog</a> [152]  <a href="#">Diagram Toolbar</a> [138]

### Notes

- To set the global appearance of all elements throughout a model, use the Options dialog; select the **Tools | Options** menu option, then select **Standard Colors** and **Diagram | Appearance** from the options tree
- To override the global appearance and define a default appearance of a selected element (or several selected elements) on all diagrams on which it occurs, set the Default Appearance for that element; right-click the element and choose the **Appearance | Default Appearance** context menu option

### Learn more

- [Diagram Appearance Options](#)<sup>[622]</sup>
- [Set An Element's Default Appearance](#)<sup>[927]</sup>

## 5.5.3.2 Context Filter a Diagram

As you review a modeling diagram, it is possible to click on any element on the diagram and show only those elements that are directly related to the selected element. All other elements are grayed out until you click on the diagram background.

**Access** **Right-click diagram background | Context Filtering**  
**Diagram Filters window | right-click background | Context Filtering, or**  
**Diagram Format Toolbar | Context Filter icon**

### Notes

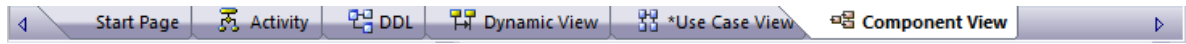
- When you set the option and click on an element, any elements on the diagram that are **not** directly related to the selected element are grayed out
- Setting the context filter applies to all diagrams that are open; you can select an element on each opened (floating) diagram and examine their relationships together
- To clear the filtering (but not turn Context Filtering off) press **Esc**
- To select the next element in a direct **chain** or **group** of relationships on a diagram, press the **Tab** key; the **Tab** key does not take you out of the element group
- To select the next element on the diagram, and display **that** element's related elements, use the **arrow** keys to move up, down, left or right to the next element in that direction; when moving left or right, the selection tends to be upwards rather than downwards
- When you enable the option on the diagram background or Diagram Filters window, it overrides the Diagram Filters settings; to use the Diagram Filters again, toggle the **Context Filter** menu option off
- When you enable the option using the diagram Format Toolbar, it activates the **Filter Diagram elements by Name** field; if you type text in that field the diagram is filtered according to the text value, but if you leave the field blank the diagram can be filtered according to the relationships of each selected element
- Context Filtering can be made even more useful by highlighting the incoming and outgoing relationships in different colors, by pressing and holding the **L** key on the selected element

### Learn more

- [Format Toolbar](#)<sup>[785]</sup>
- [Diagram Filters](#)<sup>[718]</sup>
- [Connect Elements](#)<sup>[1109]</sup>

### 5.5.4 Diagram Tabs

*Diagram tabs* indicate which diagrams and views (such as the Start Page, document reports, code editors, the Model Search and the Relationship Matrix) are currently open in the work area, by showing the view name and icon. The currently displayed view is indicated by the diagram name being in bold, and an asterisk against the tab name indicates that the diagram or view contains unsaved changes.



You open the view that you want by clicking on its tab, and you can select options for saving, reloading, moving and closing the views from the tab context menu. You can also review the tabs and select one using the View Tab Switcher window. The tabs are located, by default, at the bottom of the Diagram View; if you prefer, you can move them to the top of the Diagram View, above the Status Bar, using the General Settings page of the Options dialog.

**Access** Open a diagram or view

#### Diagram tabs context menu options

Option	Usage	See also
<b>Save Changes to '&lt;tab name&gt;'</b>	Save the changes made to the diagram or view.	
<b>Save All</b>	Save the model.	
<b>Reload '&lt;tab name&gt;'</b>	Reopen the diagram or view without the unsaved changes; that is, revert to the state before any changes were made.  Refresh the diagram or view from the repository, to show any changes made by other users in a shared model.	<a href="#">Refresh the diagram</a> <sup>[308]</sup>
<b>Float</b>	Converts the diagram or view into a floating window that you can move around your screen(s).  An alternative is to simply click on the selected tab and drag it off the Diagram Tabs row.  You could use this option to display and work on several diagrams at the same time.  To restore a floating view to a tab in the Diagram View, right-click on its Caption Bar and select the <b>Tabbed Document</b> option.	<a href="#">Window Menu</a> <sup>[121]</sup>
<b>Close '&lt;tab name&gt;'</b>	Close the selected diagram or view; Enterprise Architect prompts you to save any changes.  You can also close the tab by clicking the <i>middle</i> mouse button/ scroller.	
<b>Close All</b>	Close all open diagrams and views; Enterprise Architect prompts you to save any unsaved changes.	

Option	Usage	See also
<b>Close All Except '&lt;tab name&gt;'</b>	Close all diagrams and views except for '<tab name>'; Enterprise Architect prompts you to save any unsaved changes.	
<b>Close All &lt;view type&gt;</b>	(Where several views of the same type can be opened at the same time, such as diagrams, report documents, or text editors.)  Close all views of the same type as the selected tab, leaving views of other types still open.	
<b>Hide</b>	(Start Page only) Hide the Start Page.  To redisplay the Start Page, use the drop-down menu at the top right of the Diagram View, or select the <b>Window   Show Start Page</b> menu option.	

**Notes**

- The only context menu option available for the Start Page tab is the **Hide** option
- You can change the sequence of tabs by clicking on each tab and dragging it to a different position, as required; however, you cannot change the position of the Start Page tab - when displayed, it always sits at the left-hand end of the sequence

**Learn more**

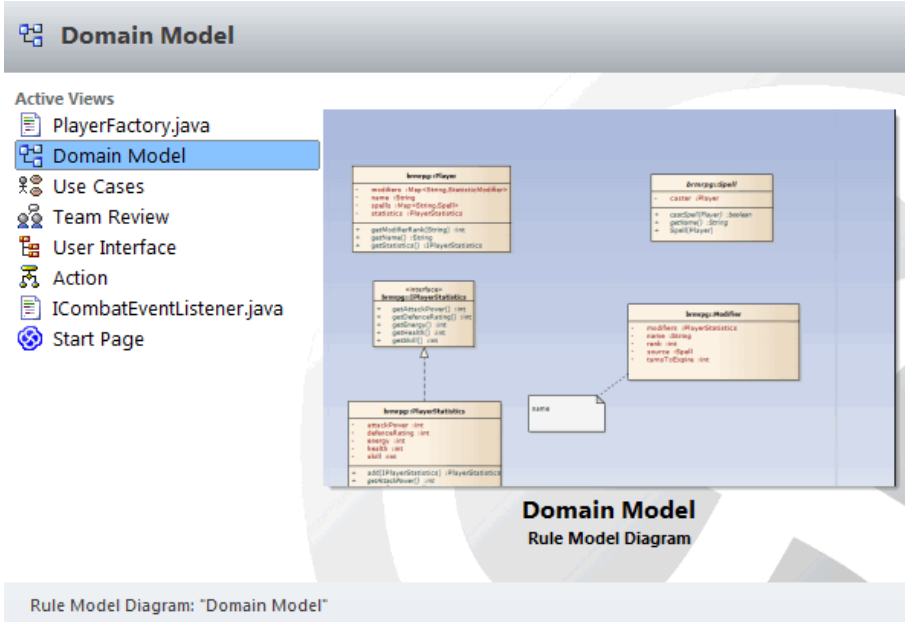
- [View Tab Switcher](#)<sup>[79]</sup>
- [General Options](#)<sup>[60]</sup>

**5.5.4.1 Switch Between Views**

When there are a number of open views available in the Diagram View (such as separate diagrams, editors and the Model Search screen), you can review them and switch between them using the **View Tab Switcher**.

**Access** Click on a view tab, press and hold ( **Ctrl** ) and press ( **Tab** )

**Switching between views**

Action	Detail	See also
<b>Selecting a different view</b>	 <p>The View Tab Switcher lists the currently open views, in order of most recently activated to least recently activated; the selection cursor is on the name of the view that was activated just before the currently active one.</p> <p>Details of the selected view are displayed on the right hand side of the screen, and include:</p> <ul style="list-style-type: none"> <li>• A preview thumbnail of the view, and</li> <li>• A description of the view and its contents</li> </ul> <p>Continue holding the ( <b>Ctrl</b> ) key to keep the window open; to move the selection cursor:</p> <ul style="list-style-type: none"> <li>• Down the list of views, press the ( <b>Tab</b> ) key</li> <li>• Up the list, press the ( <b>Shift+Tab</b> ) keys</li> </ul> <p>Alternatively, use the arrow keys.</p>	
<b>Activating a view</b>	<p>To activate a view in the Diagram View, either release the ( <b>Ctrl</b> ) key or press the ( <b>Enter</b> ) key; alternatively, click the left mouse button on the required view.</p> <p>Activating a view moves it to the top of the <b>Active Views</b> list, whereas closing a view removes it from the list completely.</p>	

### 5.5.5 Diagram Toolbox

The **Diagram Toolbox** is a panel of icons that you use to create elements and connectors on a diagram.

Within the Toolbox, related elements and connectors are organized into pages, each page containing the elements or connectors used for a particular type of diagram. The diagrams include standard UML diagrams, Enterprise Architect Extended diagrams, and diagrams customized for any MDG Technologies

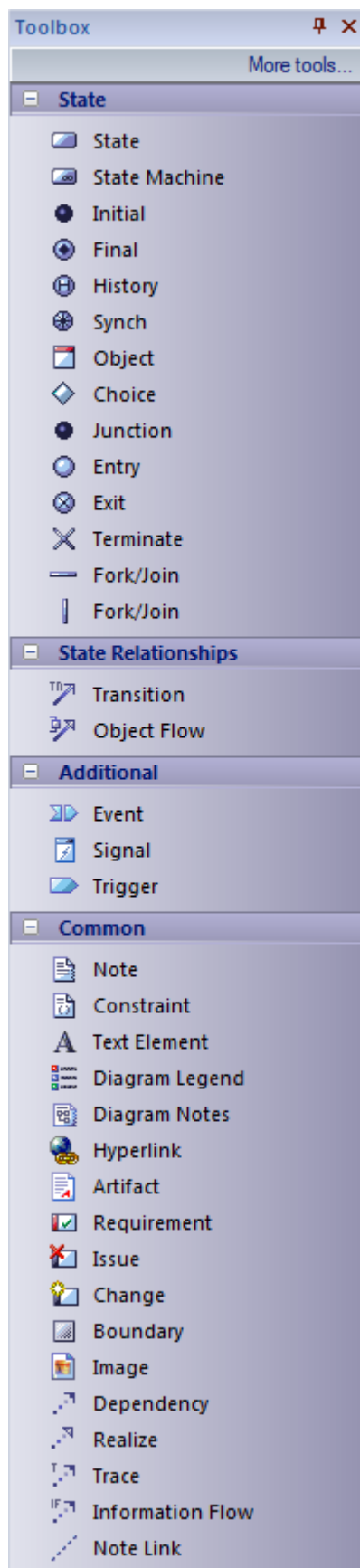


and Profiles that you have added to Enterprise Architect.

When you open a diagram, the Toolbox automatically provides the element and relationship pages corresponding to the diagram type. This does not prevent you using elements and connectors from other pages in a given diagram, though some combinations might not represent valid UML.

**Access** **Diagram | Diagram Toolbox ( Alt+5 )**

**Example**



Options


Option	Detail	See also
<b>Display specific pages</b>	<p>To display specific pages in the Toolbox, click on the <b>More Tools</b> button at the top of the Toolbox and select the appropriate UML, Extended or customized diagram type from the menu.</p> <p>In most cases, three pages display:</p> <ul style="list-style-type: none"> <li>• &lt;type&gt; elements</li> <li>• &lt;type&gt; Relationships and</li> <li>• Common</li> </ul> <p>If you select the &lt;default&gt; option, you display only the Common page.</p>	
<b>Customize the Toolbox</b>	You can customize the Toolbox display by pinning pages within the Toolbox, or by adding MDG Technologies and UML Profiles to the Toolbox.	<a href="#">Toolbox Appearance Options</a> <sup>[797]</sup> <a href="#">MDG Technologies</a> <sup>[1477]</sup> <a href="#">Using UML Profiles</a> <sup>[1472]</sup>

Create elements and connectors from the Toolbox

Step	Instruction	See also
<b>1</b>	<p>In the Project Browser, double-click on the icon against the required diagram.</p> <p>The diagram opens with the appropriate Toolbox pages for that diagram type.</p> <p>If you want a different set of elements and connectors, click on <b>More tools</b> and select the appropriate diagram type as explained above.</p>	
<b>2</b>	Click on the required item; for example, the Class element or Associate relationship.	
<b>3</b>	For element items, click anywhere on the diagram to place the new element.	
<b>4</b>	<p>For connector items, drag the cursor between the source and target elements on the diagram; the solid border of the elements changes to a hatched border as you pass the cursor over them, indicating the source and potential target elements.</p> <p>To add bends to the connector, press ( <b>Shift</b> ) as you change the drag direction of the cursor.</p>	<a href="#">Quick-linker</a> <sup>[897]</sup>

Step	Instruction	See also
	Alternatively, drag from the source element to an empty area of the diagram; the Quick-linker enables you to create the target element.	
5	Edit the element properties or connector properties, as required.	<a href="#">Properties Dialog</a> <sup>[956]</sup> <a href="#">Connector Properties</a> <sup>[1126]</sup>

### Notes

- Dropping a Package element from the Toolbox into a diagram creates a new Package in the Project Browser, and a default diagram of the same type as the current diagram
- If you are creating several elements of one type, after creating the first just press **(Shift+F3)** or **(Ctrl) + click** to create the next element of that type; for connectors, click on the source element and press **(F3)** to create another connector of the same type
- You can change an unstereotyped element to one of its stereotyped elements by dragging the stereotyped element from the Toolbox onto the unstereotyped element in the diagram; for example, you can stereotype a Class by dragging a Table element or a Profiled Class element onto it
- As you drag the stereotyped element, you can press **(Ctrl)** to apply the element stereotype to the unstereotyped element, or **(Shift)** to add the stereotyped element as a new element; otherwise a small context menu displays offering these actions as options
- If the diagram element already has the stereotype, you can also drag the Toolbox element onto it to synchronize the element's stereotype Tagged Values
- The Toolbox can be docked on either side of the diagram, or free floated on top of the diagram to expose more surface for editing
- You can also hide and show the whole Toolbox using the  **(Show Element Toolbox)** button on the Workspace Views toolbar
- Enterprise Architect provides Toolbox pages for a wide range of MDG Technologies, such as ArchiMate, BPEL, BPMN, Data Flow Diagrams, ICONIX and Mind Mapping, as part of the initial install

### Learn more

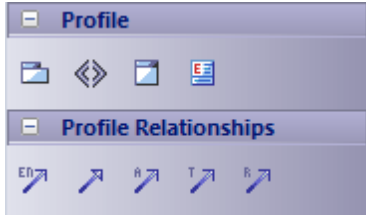
- [Stereotype Dialog](#) <sup>[1452]</sup>
- [Synchronize Tags and Constraints](#) <sup>[1473]</sup>
- [ArchiMate](#) <sup>[1926]</sup>
- [BPEL](#) <sup>[1870]</sup>
- [BPMN](#) <sup>[1845]</sup>
- [Data Flow Diagrams](#) <sup>[1797]</sup>
- [ICONIX](#) <sup>[2282]</sup>
- [Mind Mapping](#) <sup>[1794]</sup>

### 5.5.5.1 Toolbox Appearance Options

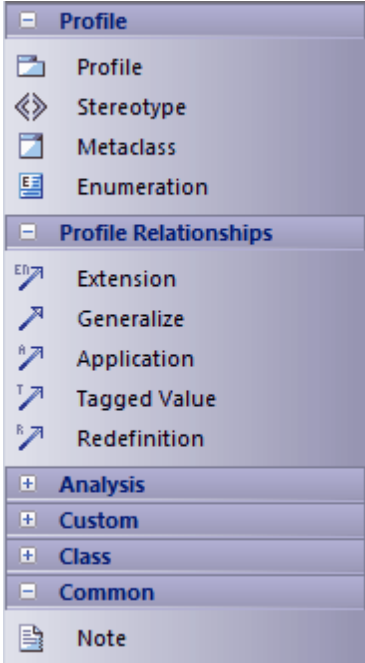
You can modify the appearance of the Diagram Toolbox pages in several ways, through the Toolbox context menu.

**Access** Right-click on the Toolbox page

#### Options

Option	Detail	See also
<b>Hide or Show element or relationship labels</b>	<p>Select the <b>Hide Labels</b> (or <b>Show Labels</b>) context menu option.</p> <p>The icons in the page then 'wrap' within the page, without text labels.</p>  <p>When you hide the labels, you can display the label of an individual element or relationship by moving the cursor over the icon.</p>	
<b>Pinning pages to display all the time</b>	<p>Select the <b>Pin in Toolbox</b> context menu option (not available on the Common page, which displays all the time anyway).</p> <p>For example, if you 'pinned' the Class elements page, and switched to the Communication pages, the Toolbox would include a collapsed Class elements page below the Communication pages.</p> <p>You can also pin a <b>SysML Common</b> page to the Toolbox, simply by making the SysML 1.3 Technology the <b>Active</b> technology for your work login.</p>	<a href="#">SysML Model Elements Toolbox</a> <sup>[2310]</sup>
<b>Unpinning pages</b>	<p>To unpin the page so that it displays only in its own Toolbox group, select the <b>Unpin from Toolbox</b> context menu option.</p>	
<b>Collapsing pages</b>	<p>To collapse a page to just show the heading (&lt;type&gt; elements, &lt;type&gt; Relationships or Common), click on the 'minus' box at the left of the page heading; alternatively, right-click on the page and select the <b>Collapse</b> context menu option.</p> <p>To expand the page again, click on the heading.</p>	

**Tailor Toolbox to list all required pages at once**

Step	Instruction	See also
1	Click on the <b>Set Toolbox Visibility</b> context menu option. The Visible Toolbox Pages dialog displays.	
2	By default, the dialog lists the element pages only, in the order: UML pages, Extended pages, MDG Technology pages.  To include the corresponding relationship pages, select the <b>Show Relationship Pages</b> checkbox at the bottom of the dialog.	
3	For each page to display on the Toolbox, select the <b>Visible</b> checkbox. Deselect the checkbox if you no longer require a page to be displayed.	
4	When you have defined the list of pages to display, click on the <b>OK</b> button. The pages you have selected are pinned to the Toolbox in a collapsed state, underneath the current diagram-type pages.  	
5	To expand a page, click on the heading.  To remove an individual page, expand it, right-click on it, and select the <b>Unpin from Toolbox</b> context menu option.	

#### Notes

- On a Toolbox page for an MDG Technology or UML Profile, if you right-click directly on a profile element

an additional option - **Synchronize Stereotype** - is available at the end of the context menu; this enables you to synchronize the Tagged Values and constraints for all elements created from the selected profile element

- MDG Technologies can impose their own Toolbox page visibility; for example, if ICONIX is the active technology, all six ICONIX pages are automatically exposed in the Toolbox
- If the active Technology pages duplicate UML or Extended pages (as the ICONIX pages do), the pinned Technology pages override and replace the pinned UML and Extended pages; for example, if ICONIX is active and you have pinned the Extended Analysis page, the Analysis page in the list is the *ICONIX-defined page*, not the Extended Analysis page

#### Learn more

- [Synchronize Tags and Constraints](#) 

### 5.5.5.2 Toolbox Shortcut Menu

When you add elements and connectors into a diagram, you can do so using the **Toolbox shortcut menu** instead of employing the full graphical Toolbox. The menu provides options to select:

- Elements specific to the current diagram type
- Relationships specific to the current diagram type
- Elements and relationships from any pages pinned in the Toolbox
- Common elements and relationships
- Elements and connectors for other diagram types

The advantage of using the shortcut menu is that it provides an increased amount of the workspace to be used for diagramming rather than displaying fixed (instead of pop-up) menus.

#### Use the Toolbox shortcut menu

Step	Action	See also
1	Open a diagram.	
2	<p>Either:</p> <ul style="list-style-type: none"> <li>• Click on the diagram background and press <b>Insert</b> or <b>Spacebar</b></li> <li>• Press and hold <b>Ctrl</b> or <b>Shift</b> and right-click on the diagram background</li> <li>• Right-click on the diagram background and select the <b>New Element or Connector</b> context menu option</li> </ul> <p>The shortcut menu displays, listing the current diagram-type elements and connectors.</p>	
3	If necessary, select the <b>Other</b> option or a pinned Toolbox page option to list elements and connectors for a different diagram type.	
4	Select the element or connector to include in the diagram; the object is added to the diagram.	

Step	Action	See also
	If you select the <b>Other</b> context menu option, the final menu in the sequence offers the <b>Activate &lt;Type&gt; Toolbox</b> option at the top; this opens and activates the corresponding page in the Toolbox, if the Toolbox is visible.	

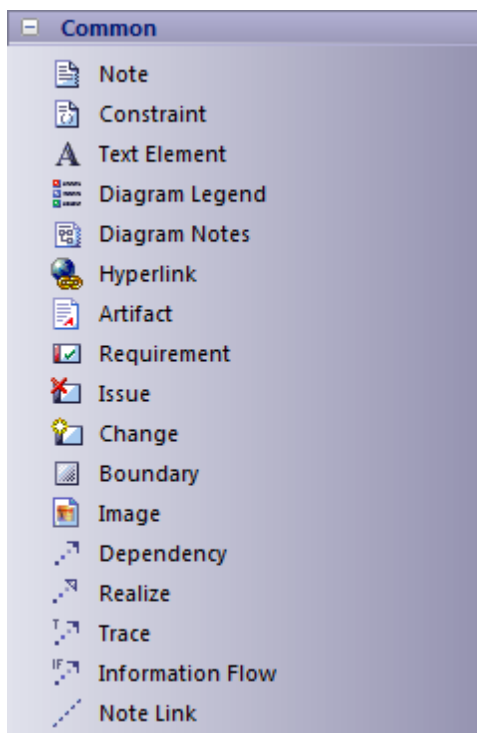
### Notes

- As with the Toolbox itself, if an MDG Technology:
  - Is active
  - Automatically pins Toolbox pages, and
  - Duplicates UML or Extended pages

The pinned Technology pages override and replace the pinned UML or Extended pages in the initial **Toolbox** shortcut menu

### 5.5.5.3 Common Page





















The **Common** page of the Diagram Toolbox provides the elements and relationships that can be used on any diagram. It is displayed at the bottom of every other set of pages in the Toolbox.



### Common Elements and Connectors

Click on an icon below for more information on the corresponding element or connector.



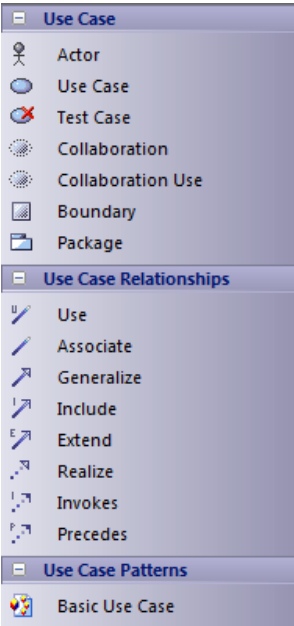
Elements	Connectors
 Note	 Dependency
 Constraint	 Realize
 Text Element	 Trace
 Diagram Legend	 Information Flow
 Diagram Notes	 Note Link
 Hyperlink	
 Document	
 Artifact	
 Requirement	
 Issue	
 Change	
 Boundary	
 Image	
 Working Set	
 Chart Object	

#### 5.5.5.4 Use Case Toolbox

The Use Case Toolbox is used to model the system functionality from the perspective of a **system user**; **Use Case** objects are used to build the Use Case models, which describe the **functionality** of the system to be built, the requirements, the constraints and how the user interacts with the system.

**Sequence diagrams** can be associated with Use Cases to capture work flow and system behavior.

**Use Case Toolbox pages**

Image	Detail	See also
	<p>A system user is called an <b>Actor</b> and is drawn as a stick figure, although this could represent another computer system or similar.</p> <p>A <b>Use Case</b> element defines a discrete piece of functionality the system provides for the user to either perform some piece of work or generate something of value using the system.</p> <p>Examples of Use Cases are: login, open account, transfer funds, check balance and logout; each of these implies some purposeful and discrete functionality the system is to provide to a user.</p> <p>A <b>Test Case</b> describes what must be set up in order to test a particular feature.</p> <p>The connectors available include:</p> <ul style="list-style-type: none"> <li>• <b>Associate</b> (an Actor uses a Use Case)</li> <li>• <b>Extend</b> (one Use Case can extend another)</li> <li>• <b>Include</b> (one Use Case can include another) and</li> <li>• <b>Realize</b> (the source Use Case might realize some business requirement)</li> <li>• <b>Invokes</b> (stereotyped <b>Dependency</b> relationship defined by the Open Modeling Language (OML), indicating that Use Case A, at some point, causes Use Case B to happen)</li> <li>• <b>Precedes</b> (stereotyped <b>Dependency</b> relationship defined by the Open Modeling Language (OML), indicating that Use Case C must complete before Use Case D can begin)</li> </ul> <p>The <b>Basic Use Case</b> icon adds the Basic Use Case Pattern to your diagram as a starting point for developing the diagram (drag the icon onto the diagram to open the Add Pattern &lt;name&gt; to Diagram dialog and click on the <b>OK</b> button).</p>	<p><a href="#">Actor</a> <sup>[1284]</sup></p> <p><a href="#">Use Case</a> <sup>[1352]</sup></p> <p><a href="#">Test Case</a> <sup>[2010]</sup></p> <p><a href="#">Add Pattern Dialog</a> <sup>[1469]</sup></p>

### Notes

- To add an **element** to the currently-open diagram, click on the appropriate icon and drag it into position on the diagram; set an element name and other properties as prompted
- To add a **relationship**, click on the appropriate icon, then click on the start element in the diagram and drag to the end element

### Learn more

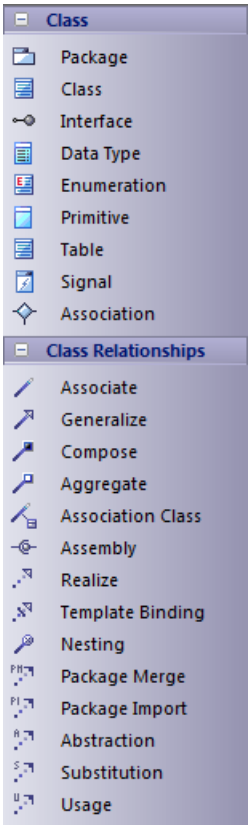


- [Use Case Diagram](#) <sup>[1201]</sup>

### 5.5.5.5 Class Toolbox

The **Class** Toolbox can be used for **Package** diagrams, **Class** diagrams and **Object** diagrams: those diagrams that usually display elements concerned with the **logical** structure of the system, such as Objects, Classes and Interfaces.

Logical models can include **domain** models (high level business driven object model) through to strict development **Class** models (define inheritance, attributes, operations).




#### Class Toolbox pages

Image	Detail	See also
 <p>The screenshot shows the Class Toolbox with two main sections: 'Class' and 'Class Relationships'. The 'Class' section includes icons for Package, Class, Interface, Data Type, Enumeration, Primitive, Table, Signal, and Association. The 'Class Relationships' section includes icons for Associate, Generalize, Compose, Aggregate, Association Class, Assembly, Realize, Template Binding, Nesting, Package Merge, Package Import, Abstraction, Substitution, and Usage.</p>	<p>The Class group is used for creating <b>Class models</b> and <b>Data models</b>.</p> <p>Class modeling uses the <b>Class</b> and <b>Interface</b> elements, with occasional use of the <b>Object</b> element to model Class instances; you can add <b>Association</b> or <b>Aggregation</b> relationships.</p> <p>Use the <b>Table</b> element to insert a stereotyped Class for use in data modeling.</p> <p>A <b>Primitive</b> element can be used to support the Meta-Object Facility (MOF) specification.</p>	<p><a href="#">Data Modeling</a> </p> <p><a href="#">MOF</a> </p>

#### Notes

- To add an **element** to the currently-open diagram, click on the appropriate icon and drag it into position on the diagram; set an element name and other properties as prompted
- To add a **relationship**, click on the appropriate icon, then click on the start element in the diagram and drag to the end element

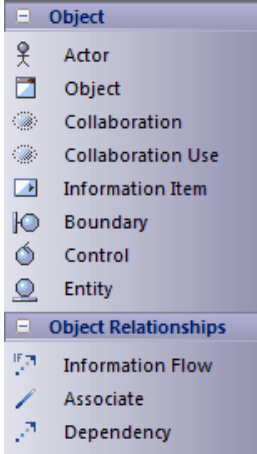
#### Learn more

- [Package Diagrams](#) 
- [Class Diagrams](#) 
- [Object Diagrams](#) 

### 5.5.5.6 Object Toolbox

The **Object** Toolbox is used to create Object diagrams, which reflect multiplicity and the roles that instantiated Classes could serve. They are useful in creating different cases in which relationships and Classes are applied.

#### Object Toolbox pages

Image	Detail	See also
	<p>The user is called an <b>Actor</b> and is drawn as a stick figure, although this could represent another computer system or similar.</p> <p>An <b>Object</b> is an instance of a <b>Class</b>.</p>	<p><a href="#">Actor</a> <small>[1284]</small></p> <p><a href="#">Object</a> <small>[1379]</small></p>

#### Notes

- To add an **element** to the currently-open diagram, click on the appropriate icon and drag it into position on the diagram; set an element name and other properties as prompted
- To add a **relationship**, click on the appropriate icon, then click on the start element in the diagram and drag to the end element

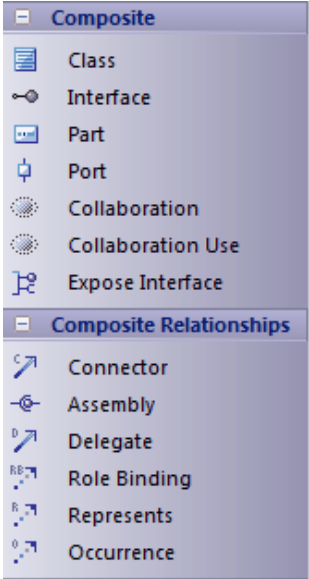
#### Learn more

- [Object Diagrams](#) [1188]

### 5.5.5.7 Composite Toolbox

The **Composite** Toolbox is used to create Composite Structure diagrams. These reflect the internal collaboration of Classes, Interfaces or Components to describe a functionality or to express run-time architectures, usage patterns and the participating elements' relationships, which static diagrams might not show.

#### Composite Toolbox pages

Image	Detail	See also
 <p>The <b>Composite</b> toolbox contains the following elements:</p> <ul style="list-style-type: none"> <li>Class</li> <li>Interface</li> <li>Part</li> <li>Port</li> <li>Collaboration</li> <li>Collaboration Use</li> <li>Expose Interface</li> <li>Composite Relationships</li> <li>Connector</li> <li>Assembly</li> <li>Delegate</li> <li>Role Binding</li> <li>Represents</li> <li>Occurrence</li> </ul>	<p>The <b>Class</b> elements form the basic structure of the diagram, with <b>Interface</b>, <b>Part</b>, <b>Port</b> and <b>Collaboration</b> elements defining the functionality or use of the Classes.</p>	<p><a href="#">Composite Structure Diagrams</a> <sup>[1168]</sup></p>

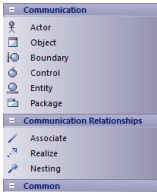
### Notes

- To add an **element** to the currently-open diagram, click on the appropriate icon and drag it into position on the diagram; set an element name and other properties as prompted
- To add a **relationship**, click on the appropriate icon, then click on the start element in the diagram and drag to the end element

### 5.5.5.8 Communication Toolbox

The **Communication** Toolbox is used to model dynamic interactions between elements at run-time.

### Communication Toolbox pages

Image	Detail	See also
 <p>The <b>Communication</b> toolbox contains the following elements:</p> <ul style="list-style-type: none"> <li>Actor</li> <li>Object</li> <li>Boundary</li> <li>Control</li> <li>Entity</li> <li>Package</li> <li>Communication Relationships</li> <li>Associate</li> <li>Realize</li> <li>Nesting</li> <li>Common</li> </ul>	<p>Communication diagrams are used to model work flow and sequential passing of messages between elements in real time; they are often placed beneath <b>Use Case</b> elements to further expand on Use Case behavior over time.</p> <p>The <b>Actor</b> element models a user of the system, while the other elements model things within the system, including standard elements (<b>Object</b>), user interface component (<b>Boundary</b>), <b>Controller</b> and <b>Entity</b>.</p>	<p><a href="#">Communication Diagrams</a> <sup>[1259]</sup></p>

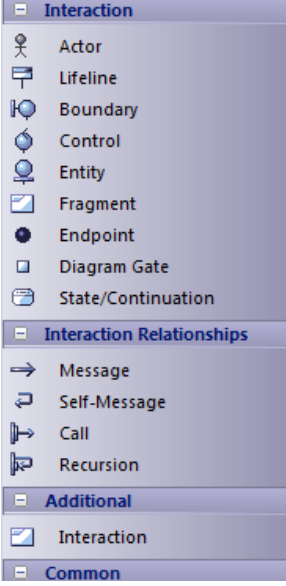
### Notes

- To add an **element** to the currently-open diagram, click on the appropriate icon and drag it into position on the diagram; set an element name and other properties as prompted
- To add a **relationship**, click on the appropriate icon, then click on the start element in the diagram and drag to the end element

### 5.5.5.9 Interaction Toolbox

The **Interaction** Toolbox is used for Interaction diagrams (**Sequence**, **Timing**, **Communication** or **Interaction Overview**), which are used to model work flow and sequential passing of messages between elements in real time.

#### Interaction Toolbox pages

Image	Detail	See also
	<p>The Interaction group is used to model dynamic interactions between elements at run-time; Interaction diagrams are often placed beneath <b>Use Case</b> elements to further expand on Use Case behavior over time.</p> <p>The <b>Actor</b> element models a user of the system, while the other elements model things within the system, including standard elements (<b>Lifeline</b>), user interface components (<b>Boundary</b>), <b>Controller</b> and <b>Entity</b>.</p> <p>The <b>Message</b> (sequence) relationship is used to model the flow of information and processing between elements.</p>	<p><a href="#">Sequence diagram</a> [1249]</p>

#### Notes

- Messages can be simple or recursive calls
- To add an **element** to the currently-open diagram, click on the appropriate icon and drag it into position on the diagram; set an element name and other properties as prompted
- To add a **relationship**, click on the appropriate icon, then click on the start element in the diagram and drag to the end element

#### Learn more

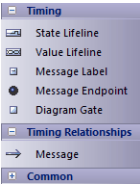
- [Timing](#) [1225]
- [Communication](#) [1259]
- [Interaction Overview](#) [1262]

### 5.5.5.10 Timing Toolbox

The **Timing** Toolbox is used solely for Timing diagrams, which use a time-scale to define the behavior of objects. The time-scale visualizes how the objects change state and interact over time.

Timing diagrams can be used for defining hardware-driven or embedded software components, and time-driven business processes.

Timing Toolbox pages

Image	Detail	See also
	<p>A <b>Lifeline</b> is the path an object takes across a measure of time, indicated by the x-axis.</p> <p>A <b>State Lifeline</b> follows discrete transitions between states, which are defined along the y-axis of the timeline; any transition has optional attributes of timing constraints, duration constraints and observations.</p> <p>A <b>Value Lifeline</b> shows the lifeline's state across the diagram, with parallel lines indicating a steady state; a cross between the lines indicates a transition or change in state.</p>	<p><a href="#">State Lifeline</a> <small>f1335</small></p> <p><a href="#">Value Lifeline</a> <small>f1355</small></p>

Notes

- To add an **element** to the currently-open diagram, click on the appropriate icon and drag it into position on the diagram; set an element name and other properties as prompted
- To add a **relationship**, click on the appropriate icon, then click on the start element in the diagram and drag to the end element

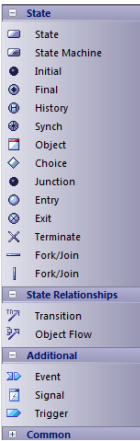
Learn more

- [Timing Diagrams](#) f1225

**5.5.5.11 State Toolbox**

The **State** Toolbox is used for State Machine diagrams to show the enableable states a Class or other element might be in and the transitions from one state to another. These diagrams are often placed under a Class element in the Project Browser to illustrate how a particular element changes over time.

State Toolbox pages

Image	Detail	See also
	<p>The State Toolbox provides elements common to State Machine diagrams; basically the <b>State</b>, <b>Entry</b> and <b>Exit</b> nodes and the <b>Object Flow</b> relationship.</p> <p>State Machine diagrams are used to model the states or conditions that elements might be in at runtime, such as active, inactive, idle, accelerating or braking.</p> <p>States can have substates; for example, <i>Accelerate</i> and <i>Brake</i> might be substates of <i>Active</i>.</p>	<p><a href="#">State Machine Diagrams</a> <small>f1203</small></p>

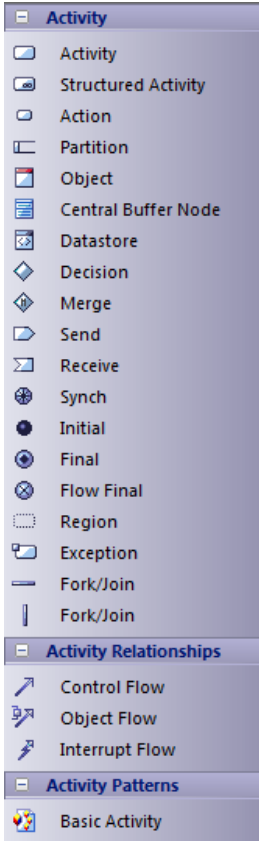
Notes

- To add an **element** to the currently-open diagram, click on the appropriate icon and drag it into position on the diagram; set an element name and other properties as prompted
- To add a **relationship**, click on the appropriate icon, then click on the start element in the diagram and drag to the end element

### 5.5.5.12 Activity Toolbox

The **Activity** Toolbox is used to model system dynamics from a number of viewpoints in **Activity** diagrams and **Interaction Overview** diagrams.

#### Activity Toolbox pages

Image	Detail	See also
	<p><b>Activity</b> elements enable you to describe the dynamics of the system from the point of view of activities and flows between them.</p> <p>Activities can be stereotyped as a process to display a business process icon; an Activity is some work that is carried out, which might overlap several <b>Use Cases</b> or form only a part of one Use Case.</p> <p><b>Send</b> and <b>Receive</b> events are included as triggers.</p> <p>A <b>Decision</b> element marks a point where processing might split based on some outcome or value.</p> <p>The <b>Flow</b> relation models an active transition and <b>Synch</b> points are used to split and rejoin periods of parallel processing.</p> <p>The <b>Basic Activity</b> icon adds the Basic Activity Pattern to your diagram as a starting point for developing the diagram (drag the icon onto the diagram to open the Add Pattern &lt;name&gt; to Diagram dialog and click on the <b>OK</b> button).</p>	<p><a href="#">Add Pattern Dialog</a> <sup>[1469]</sup></p>

#### Notes

- To add an **element** to the currently-open diagram, click on the appropriate icon and drag it into position on the diagram; set an element name and other properties as prompted
- To add a **relationship**, click on the appropriate icon, then click on the start element in the diagram and drag to the end element

#### Learn more

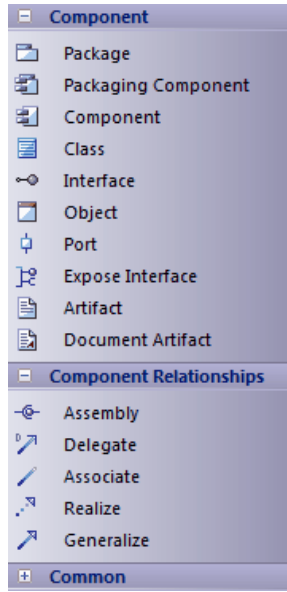
- [Activity Diagrams](#) <sup>[1199]</sup>
- [Interaction Overview Diagrams](#) <sup>[1262]</sup>



### 5.5.5.13 Component Toolbox

The **Component** Toolbox provides model elements that you can use to model the physical components of your system in a **Component diagram**.

#### Component Toolbox pages

Image	Detail	See also
	<p>The Component pages contain elements and connectors related to the actual building of the system - the components that make up the system (such as ActiveX DLLs or Java beans), the Interfaces they expose and the dependencies between those elements.</p> <p>A component is a piece of hardware or software that makes up the system; for example, a DLL or a Web Server are components that might be deployed on a Windows 2000 Server (Node).</p>	<p><a href="#">Deployment Diagram</a> <sup>1191</sup></p>

#### Notes

- To add an **element** to the currently-open diagram, click on the appropriate icon and drag it into position on the diagram; set an element name and other properties as prompted
- To add a **relationship**, click on the appropriate icon, then click on the start element in the diagram and drag to the end element

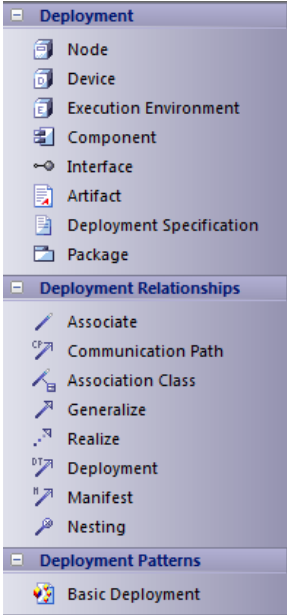
#### Learn more

- [Component Diagram](#) <sup>1194</sup>

### 5.5.5.14 Deployment Toolbox

The **Deployment** Toolbox provides model elements that you can use to model the physical components and deployment structure of your system in a **Deployment diagram**.

#### Deployment Toolbox pages

Image	Detail	See also
	<p>The Deployment group contains elements related to the actual building of the system - the components that make up the system (such as ActiveX DLLs or Java beans) and the nodes those components run on, including the physical connections between nodes.</p> <p>A Component is a piece of hardware or software that makes up the system, and a Node is a physical platform on which the component is to exist; for example, DLLs or Web Servers are Components that could be deployed on a Windows 2000 Server (Node).</p> <p>The <b>Basic Deployment</b> icon adds the Basic Deployment Pattern to your diagram as a starting point for developing the diagram (drag the icon onto the diagram to open the Add Pattern &lt;name&gt; to Diagram dialog and click on the <b>OK</b> button).</p>	<p><a href="#">Deployment Diagram</a> <sup>[119]</sup></p> <p><a href="#">Add Pattern Dialog</a> <sup>[1469]</sup></p>

### Notes

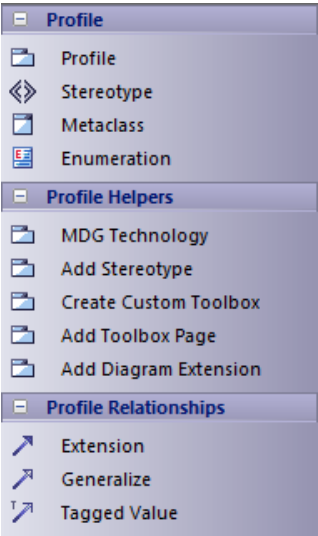
- To add an **element** to the currently-open diagram, click on the appropriate icon and drag it into position on the diagram; set an element name and other properties as prompted
- To add a **relationship**, click on the appropriate icon, then click on the start element in the diagram and drag to the end element

### 5.5.5.15 Profile Toolbox

The Profile Toolbox is used in creating **Profiles** for extending UML models. These Profiles can define custom **Stereotypes**, **Tagged Values**, **Diagram Types** and **Toolboxes**.

The Toolbox also contains a set of **Profile Helpers** that provide templates and dialogs to help you create UML Profiles, Toolbox Profiles and Diagram Profiles within an MDG Technology.

### Profile Toolbox pages

Image	Detail	See also
	<p>Profiles are used to provide a generic extension mechanism. The <b>Profile</b> Package element acts as the container for the diagrams, elements and connectors that define the Profile itself.</p> <p>A <b>Stereotype</b> provides a mechanism for varying the behavior and type of a model element.</p> <p>A <b>Metaclass</b> represents the base UML element or connector type that you are extending.</p> <p>An <b>Enumeration</b> is used to define an enumeration Tagged Value, which can be attached to one or more Stereotypes that have been defined in the Profile.</p> <p>Profile Helpers are templates and dialogs that help you create, respectively:</p> <ul style="list-style-type: none"> <li>• An MDG Technology Package to contain Profile definitions</li> <li>• Stereotypes and Metaclasses within a UML Profile</li> <li>• A customized Toolbox Profile</li> <li>• Toolbox pages within the customized Toolbox</li> <li>• A Diagram Profile to define an extended diagram type</li> </ul> <p>An <b>Extension</b> relationship shows that a Stereotype <b>extends</b> a Metaclass; all Stereotypes must extend either:</p> <ul style="list-style-type: none"> <li>• One or more Metaclasses, or</li> <li>• Another Stereotype that extends a Metaclass; you can create a <b>chain</b> of extending Stereotypes that ultimately extend the Metaclass</li> </ul> <p>A <b>Generalize</b> relationship shows that one Stereotype specializes a more general Stereotype; the more general Stereotype must still extend a Metaclass.</p> <p>A <b>Tagged Value</b> relationship is used to define a Tagged Value which is a reference (that is, RefGUID) to an element with a specific stereotype.</p>	<p><a href="#">Add Stereotypes and Metaclasses</a> <sup>[1488]</sup></p> <p><a href="#">Add an Enumeration to a Stereotype</a> <sup>[1493]</sup></p> <p><a href="#">Using the Profile Helpers</a> <sup>[1528]</sup></p> <p><a href="#">Use the Tagged Value Connector</a> <sup>[1498]</sup></p>

### Notes

- To add an **element** to the currently-open diagram, click on the appropriate icon and drag it into position on the diagram; set an element name and other properties as prompted
- To add a **relationship**, click on the appropriate icon, then click on the start element in the diagram and drag to the end element












### Learn more

- [Developing Profiles](#) <sup>[1485]</sup>
- [Using UML Profiles](#) <sup>[1472]</sup>

### 5.5.5.16 Metamodel Toolbox

The **Metamodel** Toolbox provides model elements that you can use to create **Metamodel diagrams**, with support for **Meta-Object Facility (MOF)** diagrams.

#### Metamodel Toolbox pages

Image	Detail	See also
 <b>Metamodel</b>  Package  Class  Enumeration  Primitive  <b>Metamodel Relationships</b>  Generalize  Associate  Compose  Aggregate  Common	<p>A <b>Package</b> is a namespace as well as an element that can be contained in other Package's namespaces.</p> <p>A <b>Class</b> is a representation of objects, that reflects their structure and behavior within the system.</p> <p>An <b>Enumeration</b> is used to provide a list of named values as the range of a particular type.</p> <p>A <b>Primitive</b> supports the MOF specification (<b>deprecated</b> - use the <b>UML Primitive</b> in the <b>Class</b> group).</p>	<p><a href="#">Package</a> <sup>[1382]</sup></p> <p><a href="#">Class</a> <sup>[1363]</sup></p> <p><a href="#">Enumeration</a> <sup>[1374]</sup></p> <p><a href="#">Class</a> <a href="#">Toolbox</a> <sup>[803]</sup></p>

#### Notes

- To add an **element** to the currently-open diagram, click on the appropriate icon and drag it into position on the diagram; set an element name and other properties as prompted
- To add a **relationship**, click on the appropriate icon, then click on the source element in the diagram and drag to the target element

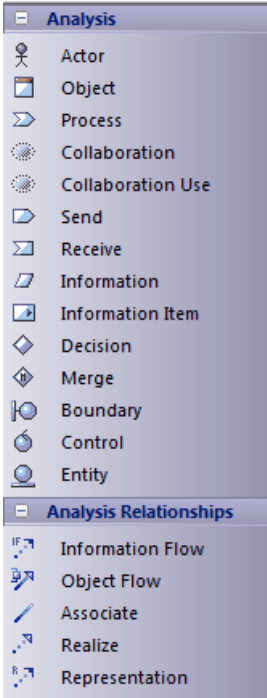
#### Learn more

- [MOF](#) <sup>[2456]</sup>

### 5.5.5.17 Analysis Toolbox

**Analysis-type elements** are used early in modeling to capture business processes, activities, collaborations and general domain information. They are generally used in **Analysis diagrams**.

#### Analysis Toolbox pages

Image	Detail	See also
	<p>You can use <b>stereotyped Activities</b> to model business processes, or stereotyped elements to capture standard UML business process modeling extensions such as worker, case worker, entity, and controller.</p>	

### Notes

- To add an **element** to the currently-open diagram, click on the appropriate icon and drag it into position on the diagram; set an element name and other properties as prompted
- To add a **relationship**, click on the appropriate icon, then click on the start element in the diagram and drag to the end element

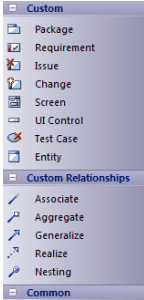
### Learn more

- [Analysis Diagrams](#) <sup>[1801]</sup>

#### 5.5.5.18 Custom Toolbox

The **Custom** Toolbox contains **extended** UML elements that can be of use in modeling or designing your system in a **Custom diagram**.

### Custom Toolbox pages

Image	Detail	See also
	<p>A <b>Package</b> is a namespace as well as an element that can be contained in other Package's namespaces.</p> <p>A <b>Requirement</b> is a custom element used to capture requirements external to standard UML elements; a Requirement expresses required system behavior that can cross several Use Cases.</p> <p>You can connect Requirements to other elements using the <b>Realize</b> connector to express the implementation of a requirement and hence the traceability from user requirements to what is being built.</p> <p>An <b>Issue</b> element is a structured comment that contains information about defects and issues relating to the system/model; affected elements are connected by <b>Trace</b> connectors.</p> <p>A <b>Change</b> element is a structured comment that contains information about changes requested to the system/model; affected elements are connected by <b>Trace</b> connectors.</p> <p>A <b>Screen</b> provides a stereotyped Class element that displays a GUI type screen; this can be used to express application GUI elements and flows between them.</p> <p>A <b>UI Control</b> likewise can be used to express GUI controls.</p> <p>A <b>Test Case</b> element defines what must be set up in order to test a particular feature; it enables you to define a set of tests once for a number of elements, and provides greater visibility for tests.</p> <p>An <b>Entity</b> is a stereotyped element that represents any general thing not captured by the element or Class type elements (for example a trading partner); use of this element is <b>deprecated</b> - it was originally intended to take the role now occupied by a <b>Table</b> element.</p>	<p><a href="#">Package</a> <sup>[1382]</sup></p> <p><a href="#">Requirements</a> <sup>[1763]</sup></p> <p><a href="#">Traceability</a> <sup>[723]</sup></p> <p><a href="#">Issues (Defects)</a> <sup>[2631]</sup></p> <p><a href="#">Trace</a> <sup>[1446]</sup></p> <p><a href="#">Changes and Issues</a> <sup>[2631]</sup></p> <p><a href="#">Screen</a> <sup>[1991]</sup></p> <p><a href="#">UI Control</a> <sup>[1992]</sup></p> <p><a href="#">Test Case</a> <sup>[2010]</sup></p> <p><a href="#">Working on Test Records</a> <sup>[2605]</sup></p> <p><a href="#">Table</a> <sup>[1942]</sup></p>

### Notes

- To add an **element** to the currently-open diagram, click on the appropriate icon and drag it into position on the diagram; set an element name and other properties as prompted
- To add a **relationship**, click on the appropriate icon, then click on the start element in the diagram and drag to the end element

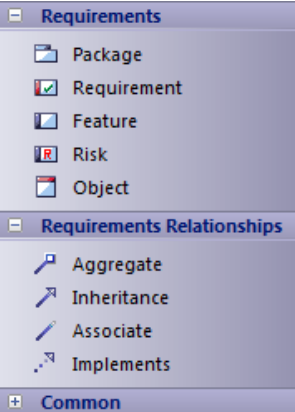
### Learn more

- [Custom Diagram](#) <sup>[1796]</sup>

## 5.5.5.19 Requirements Toolbox

As an analysis step, often it is desirable to capture simple system **requirements**. These are eventually **realized** by **Use Cases**.

### Requirements Toolbox pages

Image	Detail	See also
	<p>A <b>Package</b> is a namespace as well as an element that can be contained in other Package's namespaces.</p> <p>A <b>Requirement</b> defines a requirement of a system. Note that there are several different requirement types, such as:</p> <ul style="list-style-type: none"> <li>• Display</li> <li>• Functional</li> <li>• Performance</li> <li>• Printing</li> <li>• Report</li> <li>• Testing</li> <li>• Validate</li> </ul> <p>A <b>Feature</b> is a small client-valued function expressed as a requirement. Features are the primary requirements-gathering artifact of the <b>Feature-Driven Design (FDD)</b> methodology.</p> <p>A <b>Risk</b> element represents an identified risk to the project.</p> <p>An <b>Object</b> is an instance of a <b>Class</b>.</p>	<p><a href="#">Package</a> <sup>[1382]</sup></p> <p><a href="#">Requirement</a> <sup>[1763]</sup></p> <p><a href="#">Feature</a> <sup>[1769]</sup></p> <p><a href="#">Risk</a> <sup>[2009]</sup></p> <p><a href="#">Object</a> <sup>[1379]</sup></p>

#### Notes

- To add an **element** to the currently-open diagram, click on the appropriate icon and drag it into position on the diagram; set an element name and other properties as prompted
- To add a **relationship**, click on the appropriate icon, then click on the start element in the diagram and drag to the end element

#### 5.5.5.20 Maintenance Toolbox

The **Maintenance** elements are Issues (Defects), Changes and Tasks.

#### Maintenance Toolbox pages

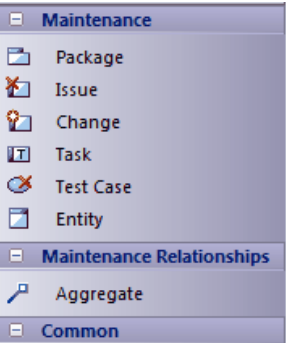
Image	Detail	See also
	<p>A <b>Package</b> is a namespace as well as an element that can be contained in other Package's namespaces.</p> <p>An <b>Issue</b> element is a structured comment that contains information about defects and issues relating to the system/model. Affected elements are connected by <b>Trace</b> connectors.</p> <p>A <b>Change</b> element is a structured comment that contains information about changes requested to the system/model. Affected elements are connected by <b>Trace</b> connectors.</p> <p>A <b>Task</b> element defines a task attached to an element, that enables resources to be assigned specifically to the task rather than just to the parent element. You can monitor these</p>	<p><a href="#">Package</a> <sup>[1382]</sup></p> <p><a href="#">Issues (Defects)</a> <sup>[2631]</sup></p> <p><a href="#">Trace</a> <sup>[1446]</sup></p> <p><a href="#">Changes and Issues</a> <sup>[2631]</sup></p> <p><a href="#">Task</a> <sup>[2010]</sup></p>

Image	Detail	See also
	<p>tasks on diagram Gantt charts.</p> <p>A <b>Test Case</b> describes what must be set up in order to test a particular feature.</p> <p>An <b>Entity</b> is a stereotyped element that represents any general thing not captured by the element or Class type elements (for example a trading partner). Use of this element is <b>deprecated</b>: it was originally intended to take the role now occupied by a <b>Table</b> element.</p>	<p><a href="#">Test Case</a> <sup>[2010]</sup></p> <p><a href="#">Table</a> <sup>[1942]</sup></p>

### Notes

- To add an **element** to the currently-open diagram, click on the appropriate icon and drag it into position on the diagram; set an element name and other properties as prompted
- To add a **relationship**, click on the appropriate icon, then click on the start element in the diagram and drag to the end element

### Learn more

- [Maintenance](#) <sup>[2619]</sup>

## 5.5.5.21 User Interface Toolbox

The **User Interface** Toolbox provides a wide range of icons that you can use to create graphical user interface diagrams.

### User Interface Toolbox pages

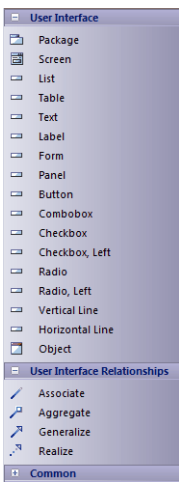
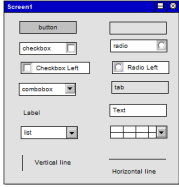
Image	Detail	See also
	<p>A <b>Package</b> is a namespace as well as an element that can be contained in other Packages' namespaces.</p> <p>A <b>Screen</b> element represents a graphical user interface. You can place GUI elements onto the screen element (see below).</p> <p><b>UI Control</b> elements are placed onto the screen element to build up a graphical user interface diagram. There are different stereotyped elements such as buttons and combo boxes.</p> <p>An <b>Object</b> is an instance of a Class.</p>	<p><a href="#">Package</a> <sup>[1382]</sup></p> <p><a href="#">Screen</a> <sup>[1991]</sup></p> <p><a href="#">UI Control</a> <sup>[1992]</sup></p> <p><a href="#">Object</a> <sup>[1379]</sup></p>



Image	Detail	See also
	<p>The diagram on the left illustrates all of the GUI elements from the Toolbox, within a <b>Screen</b> element.</p>	

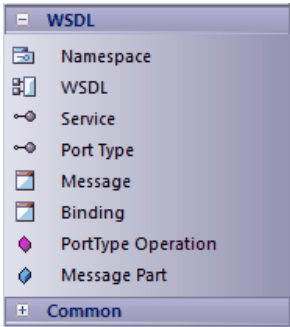
### Notes

- To add an **element** to the currently-open diagram, click on the appropriate icon and drag it into position on the diagram; set an element name and other properties as prompted
- To add a **relationship**, click on the appropriate icon, then click on the start element in the diagram and drag to the end element

### 5.5.5.22 WSDL Toolbox

The **WSDL** Toolbox provides elements you can use to rapidly develop a W3C Web Service Definition Language (WSDL) model, from which you can automatically generate WSDL documents.

#### WSDL Toolbox page

Image	Detail	See also
	<p>A <b>Namespace</b> represents the top-level container for the WSDL model. Drag this element onto an open diagram to create the necessary model structure for WSDL documents.</p> <p>A physical <b>WSDL</b> document is represented as a UML <b>Component</b>. Its interfaces represent the WSDL <b>Services</b>.</p> <p>A WSDL <b>Port Type</b> is modeled as a UML <b>Interface</b>. Its <b>Port Type Operations</b> are realized by <b>Binding</b> elements.</p> <p>Each of the operation parameters is derived from the <b>Message</b> elements defined in the Messages Package.</p>	<p><a href="#">WSDL Namespace</a> <sup>[2426]</sup></p> <p><a href="#">WSDL document</a> <sup>[2443]</sup></p> <p><a href="#">WSDL Services</a> <sup>[2441]</sup></p> <p><a href="#">WSDL Port Type</a> <sup>[2433]</sup></p> <p><a href="#">Port Type Operations</a> <sup>[2434]</sup></p> <p><a href="#">WSDL Binding</a> <sup>[2437]</sup></p> <p><a href="#">WSDL Messages</a> <sup>[2430]</sup></p>

### Notes

- To add an **element** to the currently-open diagram, click on the appropriate icon and drag it into position on the diagram; set an element name and other properties as prompted
- To add a **relationship**, click on the appropriate icon, then click on the start element in the diagram and drag to the end element

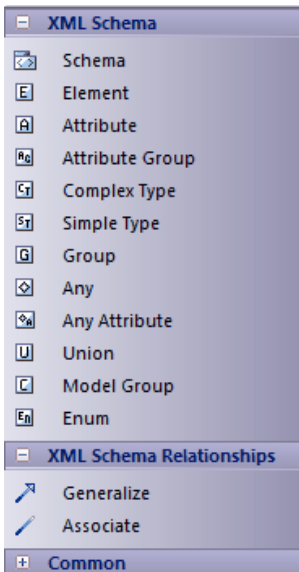
### Learn more

- [Model WSDL](#) <sup>[2426]</sup>
- [Generate WSDL](#) <sup>[2445]</sup>

### 5.5.5.23 XML Schema Toolbox

The **XML Schema** Toolbox provides model elements that you can use to model W3C XSD schema files. This Toolbox implements the constructs provided by the UML Profile for XML Schema. Having created your model, you can automatically generate the W3C XSD schema files from it.

#### XML Schema Toolbox pages

Image	Detail	See also
 <p>The screenshot shows the XML Schema Toolbox with the following items:</p> <ul style="list-style-type: none"> <li><b>XML Schema</b> (expanded) <ul style="list-style-type: none"> <li>Schema</li> <li>Element</li> <li>Attribute</li> <li>Attribute Group</li> <li>Complex Type</li> <li>Simple Type</li> <li>Group</li> <li>Any</li> <li>Any Attribute</li> <li>Union</li> <li>Model Group</li> <li>Enum</li> </ul> </li> <li><b>XML Schema Relationships</b> (expanded) <ul style="list-style-type: none"> <li>Generalize</li> <li>Associate</li> </ul> </li> <li><b>Common</b></li> </ul>	<p>A <b>Schema</b> corresponds to a UML Package, which contains the type and element definitions for a particular <b>targetNamespace</b>. Drag this item onto an open diagram to create the Package to contain your schema model elements. The Package is stereotyped as <b>XSDschema</b>.</p> <p>Open the logical diagram created under the XSDschema Package and add additional schema elements to define the model.</p>	

#### Notes

- To add an **element** to the currently-open diagram, click on the appropriate icon and drag it into position on the diagram; set an element name and other properties as prompted
- To add a **relationship**, click on the appropriate icon, then click on the start element in the diagram and drag to the end element

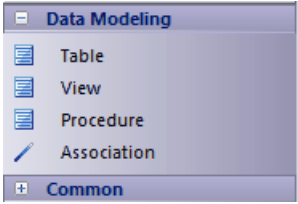
#### Learn more

- [XML Schema - XSD](#) <sup>[2387]</sup>
- [Model XSD](#) <sup>[2387]</sup>
- [Generate XSD](#) <sup>[2417]</sup>

### 5.5.5.24 Data Modeling Toolbox

The **Data Modeling** Toolbox provides a basic set of elements for database modeling and database design, in conjunction with the **UML Data Modeling Profile**.

#### Data Modeling Toolbox page

Image	Detail	See also
	<p>The <b>Table</b> element defines a table on the data model.</p> <p>The <b>View</b> element represents <b>database views</b> in the data model.</p> <p>The <b>Procedure</b> element represents stored procedures in the data model.</p>	<p><a href="#">Table</a> <sup>[1942]</sup></p> <p><a href="#">Views</a> <sup>[2371]</sup></p> <p><a href="#">Stored Procedures</a> <sup>[2364]</sup></p>

### Notes

- To add an **element** to the currently-open diagram, click on the appropriate icon and drag it into position on the diagram; set an element name and other properties as prompted
- To add a **relationship**, click on the appropriate icon, then click on the start element in the diagram and drag to the end element

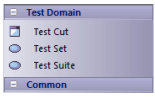
### Learn more

- [Database Modeling](#) <sup>[1937]</sup>

## 5.5.5.25 Test Domain Toolbox

The **Test Domain** Toolbox provides a basic set of elements that you can use to create Test Domain diagrams, used in the **Testpoint** facility. A Test Domain diagram provides specific elements that can aid in the logical composition of tests into **Test Sets** and **Test Suites**.

### Test Domain Toolbox page

Image	Detail	See also
	<p>A <b>Test Cut</b> is a group of operations that represent a specific behavior (such as <i>Print</i>) defined for a Class, separate from other constraints defined for the Class.</p> <p>A <b>Test Set</b> aggregates one or more groups of methods (Test Cuts) that perhaps span multiple Classes, into a single task.</p> <p>A <b>Test Suite</b> aggregates one or more groups of tasks (Test Sets).</p>	<p><a href="#">Test Cut</a> <sup>[2588]</sup></p> <p><a href="#">Test Set</a> <sup>[2588]</sup></p> <p><a href="#">Test Suite</a> <sup>[2589]</sup></p>

### Notes

- To add an **element** to the currently-open diagram, click on the appropriate icon and drag it into position on the diagram; set an element name and other properties as prompted
- To add a **relationship**, click on the appropriate icon, then click on the start element in the diagram and drag to the end element

### Learn more

- [Testpoint Management](#) <sup>[2577]</sup>

### 5.5.6 Diagram Tasks

This topic introduces many of the common tasks associated with managing diagrams.

Topic	Link
Add New Diagrams	<a href="#">Add New Diagrams</a> <sup>[822]</sup>
Set Diagram Properties	<a href="#">Set Diagram Properties</a> <sup>[823]</sup>
Add Elements and Connectors From Toolbox	<a href="#">Add Elements and Connectors From Toolbox</a> <sup>[792]</sup>
Paste from the Project Browser	<a href="#">Paste from the Project Browser</a> <sup>[833]</sup>
Copy And Paste Diagram Element	<a href="#">Copy And Paste Diagram Element</a> <sup>[836]</sup>
Place Related Elements on Current Diagram	<a href="#">Place Related Elements on Current Diagram</a> <sup>[838]</sup>
Delete a Diagram	<a href="#">Delete a Diagram</a> <sup>[839]</sup>
Rename a Diagram	<a href="#">Rename a Diagram</a> <sup>[839]</sup>
Change Diagram Type	<a href="#">Change Diagram Type</a> <sup>[840]</sup>
Diagram Navigation Hotkeys	<a href="#">Diagram Navigation Hotkeys</a> <sup>[841]</sup>
Copy Image to Disk	<a href="#">Copy Image to Disk</a> <sup>[841]</sup>
Copy Image to Clipboard	<a href="#">Copy Image to Clipboard</a> <sup>[842]</sup>
Duplicate a Diagram	<a href="#">Duplicate a Diagram</a> <sup>[842]</sup>
Z Order Elements	<a href="#">Z Order Elements</a> <sup>[844]</sup>
Set Default Diagram	<a href="#">Set Default Diagram</a> <sup>[844]</sup>
Open a Package	<a href="#">Open a Package</a> <sup>[845]</sup>

Topic	Link
Feature Visibility	<a href="#">Feature Visibility</a> <sup>[845]</sup>
Insert Diagram Properties Note	<a href="#">Insert Diagram Properties Note</a> <sup>[848]</sup>
Manage Legend Elements	<a href="#">Manage Legend Elements</a> <sup>[849]</sup>
Autosize Elements	<a href="#">Autosize Elements</a> <sup>[851]</sup>
Swimlanes	<a href="#">Swimlanes</a> <sup>[852]</sup>
Swimlanes Matrix	<a href="#">Swimlanes Matrix</a> <sup>[854]</sup>
Apply Kanban facilities to the diagram	<a href="#">Kanban Facilities</a> <sup>[857]</sup>
Using the Image Manager	<a href="#">Using the Image Manager</a> <sup>[860]</sup>
Show Realized Interfaces for a Class	<a href="#">Show Realized Interfaces for a Class</a> <sup>[865]</sup>
Label Menu Section	<a href="#">Manage Object Labels</a> <sup>[866]</sup>
Pan and Zoom a Diagram	<a href="#">Pan and Zoom a Diagram</a> <sup>[868]</sup>
Move Diagram Sections	<a href="#">Move Diagram Sections</a> <sup>[869]</sup>
View Last and Next Diagram	<a href="#">View Last and Next Diagram</a> <sup>[869]</sup>
Set Diagram Page Size	<a href="#">Set Up Diagram Page</a> <sup>[870]</sup>
Scale Image to Page Size	<a href="#">Scale Image to Page Size</a> <sup>[871]</sup>
Lock Diagram	<a href="#">Lock Diagram</a> <sup>[872]</sup>
Lay Out a Diagram	<a href="#">Lay Out a Diagram</a> <sup>[886]</sup>
Undo Last Action	<a href="#">Undo Last Action</a> <sup>[873]</sup>

Topic	Link
Redo Last Action	<a href="#">Redo Last Action</a> <sup>[873]</sup>
Present Diagrams in a Model Views Slideshow	<a href="#">Present Diagrams in a Model Views Slideshow</a> <sup>[695]</sup>

### Notes

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Update Element** permission to update or delete items on a diagram, and **Manage Diagram** permission to create, copy or delete diagrams

### Learn more

- [List of Available Permissions](#) <sup>[329]</sup>

## 5.5.6.1 Add New Diagrams

This topic explains how to add a UML diagram, Extended diagram or MDG Technology diagram to a model in Enterprise Architect.

### How to

Step	Action	See also
1	In the Project Browser, select the appropriate Package or element under which to place the diagram.	
2	<p>Do one of the following:</p> <ul style="list-style-type: none"> <li>In the Project Browser toolbar click on the <b>New Diagram</b> icon</li> <li>Right-click to open the context menu and select the <b>Add   Add Diagram</b> or <b>Add   Add &lt;type&gt; Diagram</b> menu option</li> <li>Press <b>Insert</b> and select the <b>Add   Add Diagram</b> or <b>Add   Add &lt;type&gt; Diagram</b> menu option, or</li> <li>Select the <b>Project   New Diagram</b> menu option</li> </ul> <p>The New Diagram dialog displays.</p>	
3	The <b>Name</b> field defaults to the name of the selected Package or element; if necessary, type a different name for the new diagram.	
4	<p>In the Select From panel, click on the appropriate diagram category for the diagram.</p> <p>The Diagram Types panel displays a list of the diagram types within the selected</p>	<a href="#">UML Diagrams</a> <sup>[1183]</sup>

Step	Action	See also
	category.	
5	In the Diagram Types panel, click on the type of diagram to create.	
6	Click on the <b>OK</b> button to create your new diagram.  Enterprise Architect checks whether there is a template diagram for this diagram type in the Templates Package and, if so, applies the display characteristics defined in that template.	<a href="#">Set Element Templates Package</a> <sup>[929]</sup>

### Notes

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Diagrams** permission to create new diagrams
- The diagram type determines the default toolbar associated with the diagram and whether it can be set as a child of another element in the Project Browser (for example, a Sequence diagram under a Use Case)

### Learn more

- [Manage Diagrams](#)<sup>[329]</sup>

## 5.5.6.2 Set Diagram Properties

You can set several properties of a diagram using the diagram Properties dialog. Some properties influence the display and some are logical attributes that appear in the documentation.

There are several options for opening the diagram Properties dialog for a given diagram:

- Select the **Diagram | Properties** menu option to open the Properties dialog for the currently active diagram
- Right-click on the required diagram in the Project Browser and select the **Properties** context menu option
- Right-click on the background of the open diagram and select the **Properties** context menu option
- Double-click in the background of the open diagram

In the Diagram Properties dialog you can set properties including name, author and version information, zoom factor, paper size and layout, diagram notes and various appearance attributes.

### Notes

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Update Diagrams** permission to change diagram properties
- You can also set the default diagram background color and color gradients for the diagram background on the Diagram Gradients and Backgrounds page of the Options dialog

Learn more

- [Update Diagrams](#) <sup>[329]</sup>
- [Diagram Gradients and Backgrounds](#) <sup>[613]</sup>
- [General Tab](#) <sup>[824]</sup>
- [Configure Diagram Display](#) <sup>[825]</sup>
- [Define Element Characteristics](#) <sup>[828]</sup>
- [Features Tab](#) <sup>[830]</sup>
- [Connectors Tab](#) <sup>[831]</sup>

**5.5.6.2.1 General Tab**

You use the General tab of the diagram Properties dialog to define characteristics of the overall diagram, such as its title, version and modification date.

**Access**    **Diagram | Properties > General**

Set Diagram Properties

Field	Action	See also
<b>Name</b>	Defaults to the name of the parent Package, as the name of the diagram.  If necessary, overwrite the field with a different diagram name.	
<b>Author</b>	Defaults to the name of the person who created the diagram.  If necessary, overwrite the field with a different author name, or click on the drop-down arrow and select a name from the list. You might do this if the diagram becomes the responsibility of another person.	
<b>Version</b>	Defaults to <b>1.0</b> as the version number of the diagram.  If necessary, overwrite with a different version number.	
<b>Stereotype</b>	Type in the name of a diagram stereotype to apply, or click on the drop-down arrow and select a stereotype from the list.  You can define stereotypes to select here using the <b>Settings   UML Types</b> menu option, selecting the Stereotypes <sup>[9006]</sup> tab and creating stereotypes with a <b>Base Class</b> of <b>Diagram</b> .	<a href="#">Stereotypes Tab</a> <sup>[1461]</sup>
<b>Created</b>	Automatically displays the date the diagram was created.  If necessary, overwrite with an amended creation data, or click on the drop-down arrow and select the date.	
<b>Modified</b>	Displays the date and time on which the diagram was last modified.	



Field	Action	See also
<b>Notes</b>	Type in any additional information on the diagram. You can format the notes using the Notes toolbar at the top of the field.	<a href="#">Notes Toolbar</a> <small>[1143]</small>
<b>OK</b>	If you have finished updating the diagram properties, click on this button to save the changes and close the Properties dialog.	
<b>Cancel</b>	If you do not want to save the changes you have made, click on this button to discard the changes and close the dialog.	
<b>Help</b>	Click on this button to display this Help topic.	

### Notes

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have Update Diagrams permission to update diagram properties

### Learn more

- [Permissions List](#)  
[329]

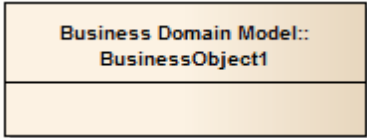
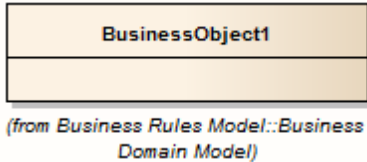
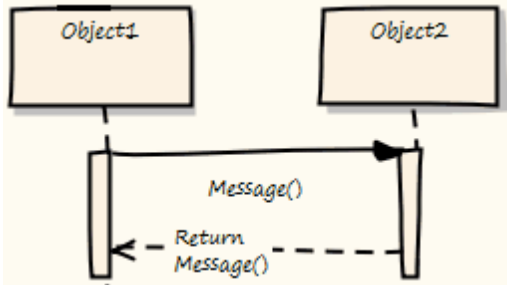
#### 5.5.6.2.2 *Configure Diagram Display*

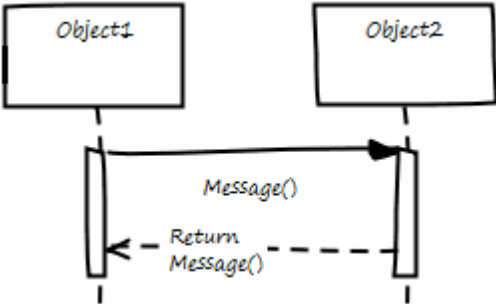
A diagram can show or hide information, and represent it in different ways depending on how you configure the display characteristics. You can perform this display configuration on the Diagram tab of the diagram Properties dialog.

**Access**    **Diagram | Properties > Diagram**

### Set the diagram display options

Field	Action	See also
<b>Use Alias if Available</b>	Display the element alias as the name, if the alias is specified.	
<b>Show Additional Parents</b>	Show the name of all parents not in the current diagram, for all Classes and interfaces.	
<b>Hide Page Border (All Diagrams)</b>	Disable the display of page borders on <b>all</b> diagrams in the model.	

Field	Action	See also
<b>Hide Page Border (Current Diagram)</b>	Disable the display of the page border on the <b>current</b> diagram.	
<b>Show Diagram Details</b>	Show diagram details in a note in the top left corner of the diagram. Deselect the option to hide the diagram details.	
<b>Show Sequence Notes</b>	Show the Sequence Notes on the current diagram.	
<b>Show Namespace</b>	Show the namespace of each element on the diagram, as part of the element name.  	
<b>Fully Qualified Namespace</b>	(If <b>Show Namespace</b> is selected) show the fully qualified namespace of each element on the diagram, under the element.  	
<b>Hand Drawn</b>	Display the diagram contents as if they had been roughly drawn by hand.  	
<b>Whiteboard Mode</b>	Displays the diagram as if it had been drawn on a whiteboard, with white fill and background.	

Field	Action	See also
	 <p>(This illustration also has Hand Drawn mode enabled.)</p>	
<b>Disable fully scoped object names</b>	Prevent the display of fully-qualified object names on the diagram.	
<b>Page Setup</b>	Adjust the scaling of the image according to the size of the page it is to be printed on.	<a href="#">Scale Image to Page Size</a> <sup>[87]</sup>
<b>Print Page Header and Page Footer</b>	Add page headers and footers to a print-out of the diagram; the headers and footers are generated from the diagram characteristics, such as the name of the creator and the date of modification.	
<b>Always Open as Element List</b>	Always display the diagram contents as a Diagram List rather than as a diagram.	<a href="#">Diagram List</a> <sup>[68]</sup>
<b>Always Open as Gantt</b>	Always display the diagram contents as a Gantt chart.	<a href="#">The Gantt View</a> <sup>[59]</sup>
<b>Set Layout Style</b>	Define the layout of the diagram, to be applied automatically when you select the Layout Diagram options.	<a href="#">Layout Diagram</a> <sup>[89]</sup>
<b>RTF Document Options</b>	Options for generating document reports for a particular diagram.	<a href="#">Document Options</a> <sup>[266]</sup>
<b>Exclude image from RTF documents</b>	Exclude this diagram image from any document report generated on the parent Package or element.	
<b>Document each contained element in RTF</b>	Include documentation on each element in this diagram, in any document report generated on the parent Package or element.	
<b>Divide Diagram into Multiple</b>	Divide each large diagram into separate pages in the document report.	<a href="#">Scaled to page</a> <sup>[87]</sup>

Field	Action	See also
<b>Pages</b>		
<b>Rotate Images</b>	Rotate each diagram image by 90 degrees in the document report.	
<b>OK</b>	If you have finished updating the diagram properties, click on this button to save the changes and close the Properties dialog.	
<b>Cancel</b>	If you do not want to save the changes you have made, click on this button to discard the changes and close the dialog.	
<b>Help</b>	Click on this button to display this Help topic.	

#### Notes



- The **Divide Diagrams into Multiple Pages** option is only effective when the **Scaled Printing** option on the Print Advanced dialog is set to **None**
- **Rotate Images** is only available for bitmap (.bmp) images

#### 5.5.6.2.3 Define Element Characteristics

It is possible to show a number of element characteristics on a selected diagram. You can define which of these characteristics to show, by selecting the corresponding check boxes on the diagram Properties dialog (and, conversely, which to hide by clearing their check boxes). You can also override the model default coding language and set a different language for the elements in the context of this diagram.

**Access**   **Diagram | Properties > Elements**

#### Set Component Display Options

Field	Action	See also
<b>Use Stereotype Icons</b>	For elements that have whole shapes drawn by Enterprise Architect (such as Analysis stereotypes), draw the alternative shape (if defined).  For elements that have an icon displayed in the top right corner, (such as an Artifact element) if <b>Show Element Stereotypes</b> is selected, display the stereotype icon instead of the stereotype text.	<a href="#">Analysis stereotypes</a>  <a href="#">Artifact</a> 
<b>Show Element Stereotypes</b>	For elements that have whole shapes drawn by Enterprise Architect, if <b>Use Stereotype Icons</b> is deselected, display any stereotype on the element.  For elements that have an icon displayed in the top right corner, indicate that a stereotype is present (icon if <b>Use Stereotype Icons</b>	

Field	Action	See also
	is selected, text if not).	
<b>Show Table Owner</b>	Display the Table Owner.	<a href="#">Set Table Owner</a> [2342]
<b>Show Element Property String</b>	Show the advanced property string for all elements; for example, {leaf}.	
<b>Show Compartments</b>	<p>Show or hide the named compartments for all elements using rectangle notation on the diagram:</p> <ul style="list-style-type: none"> <li>• Attributes</li> <li>• Operations</li> <li>• Tags (Tagged Values)</li> <li>• Requirements</li> <li>• Constraints</li> <li>• Testing (Testing Scripts)</li> <li>• Maintenance (Maintenance Scripts)</li> <li>• Package Contents</li> <li>• Notes</li> </ul> <p>If you have added custom compartments to your model display, such as Related Elements, Flow Ports or Parts, <b>Show Compartments</b> checkboxes are also listed for each of them. These checkboxes default to selected to show the compartments on the diagram; deselect the checkboxes to hide the compartments.</p> <p>If the checkboxes do not seem to operate for an element on a diagram, also check the Feature Visibility settings for that <b>element</b>.</p>	<p><a href="#">Show Test Scripts Compartment</a> [2616]</p> <p><a href="#">Show Maintenance Script in Diagram</a> [2630]</p> <p><a href="#">Add Custom Compartments to Element</a> [1604]</p> <p><a href="#">Feature Visibility</a> [845]</p>
<b>Language</b>	<p>Set the language that elements created on this diagram receive, overriding the model default language.</p> <p>If you do not want to override the model default, select the value <b>&lt;none&gt;</b> from the drop-down list.</p>	
<b>OK</b>	If you have finished updating the diagram properties, click on this button to save the changes and close the Properties dialog.	
<b>Cancel</b>	If you do not want to save the changes you have made, click on this button to discard the changes and close the dialog.	
<b>Help</b>	Click on this button to display this Help topic.	

#### 5.5.6.2.4 Features Tab

You use the Features tab of the diagram Properties dialog to define how features (attributes and operations) are displayed on the diagram.

**Access** [Diagram | Properties > Features](#)

#### Define how features are displayed on the diagram

Field	Action	See also
<b>Show Qualifiers and Visibility Indicators</b>	<p>Show or hide the qualifiers and visibility indicators on the diagram.</p> <p><i>Qualifiers</i> include such things as the 'derived' symbol (<i>/</i>) and the public key symbol (<b>PK</b>).</p> <p>Visibility indicators <sup>[2080]</sup> include such things as <b>+</b>, <b>-</b>, <b>#</b> and <b>~</b>, which indicate the scope of access of the item (such as an attribute, operation or role).</p>	<a href="#">Visibility indicators</a> <sup>[2080]</sup>
<b>Show Stereotypes</b>	Show the stereotypes on all features.	
<b>Show Property String</b>	Show the advanced property string for all element features, for example, {readOnly}.	
<b>Show Operation Return Type</b>	Display the return data type of operations.	
<b>Suppress Brackets for Operations Without Parameters</b>	Suppress brackets on operations that have no parameters; that is, <b>Opn</b> ; rather than <b>Opn()</b> .	
<b>Always Show Linked Features</b>	Force display of linked <sup>[1098]</sup> attributes and operations, regardless of the setting of any other option that might hide them.	<a href="#">Linked Attributes and Operations</a> <sup>[1110]</sup>
<b>Visible Class Members</b>	Hide Class members according to their scope and methods that specify properties.	<a href="#">Visible Class Members</a> <sup>[831]</sup>
<b>Show Attribute Detail</b>	Select whether to show both the attribute name and type, or the attribute name only.	
<b>Show Parameter Detail</b>	Control the display of method parameters.	<a href="#">Visible Class Members</a> <sup>[831]</sup>

Field	Action	See also
<b>OK</b>	If you have finished updating the diagram properties, click on this button to save the changes and close the Properties dialog.	
<b>Cancel</b>	If you do not want to save the changes you have made, click on this button to discard the changes and close the dialog.	
<b>Help</b>	Click on this button to display this Help topic.	

#### 5.5.6.2.4.1 Visible Class Members

On the Features tab of the diagram Properties dialog, you use the Visible Class Members panel to hide Class members by their scope and methods that specify properties. Use the check boxes to define the visibility of Class members.

**Access** [Diagram | Properties > Features : Visible Class Members](#)

#### Show Parameter Detail

Option	Effect
<b>None</b>	No details shown.
<b>Type Only</b>	Show the type of parameter only.
<b>Full Details</b>	Show all of the details for parameters.
<b>Name Only</b>	Show the name of the parameter only.

#### 5.5.6.2.5 Connectors Tab

To define the appearance of the connectors on a diagram, you use the Connectors tab of the diagram Properties dialog.

**Access** [Diagram | Properties > Connectors](#)

#### Define connector appearance

Field/Option/ Button	Action	See also
<b>Show Relationships</b>	Show relationships in the current diagram.	
<b>Show Collaboration Numbers</b>	Show numbering in Communication diagrams.	
<b>Show Non-Navigable Ends</b>	Indicate when an Association end is not navigable; a cross is presented at the Association connector.	
<b>Show Connector Property String</b>	Show the property string for connectors.	
<b>Suppress All Connector Labels</b>	Hide all connector labels.	
<b>Show Stereotype Labels</b>	Show (selected) or hide (deselected) any stereotype labels on the connectors on the diagram.	
<b>Connector Notation</b>	Display the required connector notation: <ul style="list-style-type: none"> <li>• <b>UML 2.1</b> - use the standard UML 2.1 notation for connectors</li> <li>• <b>Information Engineering</b> - use the Information Engineering (IE) connection style</li> <li>• <b>IDEFX1</b> - use the Integrated Definition Methods IDEFX1 connection style</li> </ul>	<a href="http://www.agiledata.org/essays/dataModeling101">http://www.agiledata.org/essays/dataModeling101</a>  <a href="http://www.idef.com/IDEF1X.html">http://www.idef.com/IDEF1X.html</a>
<b>OK</b>	If you have finished updating the diagram properties, click on this button to save the changes and close the Properties dialog.	
<b>Cancel</b>	If you do not want to save the changes you have made, click on this button to discard the changes and close the dialog.	
<b>Help</b>	Click on this button to display this Help topic.	



### 5.5.6.3 Paste from Project Browser




As well as creating new elements in the current diagram, you can drag **existing** elements (individually or as a group) from the **Project Browser** into the diagram. To select multiple elements, click on the element names in the Project Browser while pressing and holding:

- **Ctrl** to add single items to the selection of multiple elements, or
- **Shift** to select all the elements between the first and last selected items

When you drag an element onto a diagram from the Project Browser, the system response depends on whether you are dragging a **Package** element or **any other** type of element. For information on dragging a Package element onto a diagram, see the *Paste Packages* topic.

**Access**    **Drag existing element(s) onto the diagram**  
**(If the Paste Element dialog does not display, press ( Ctrl ) while dragging the elements)**

#### Select Paste options

Field/Button	Detail	See also
<b>Paste as</b>	<p>Click on the drop-down arrow and select one of the paste types:</p> <ul style="list-style-type: none"> <li>• <b>Link</b> - Paste the element as a simple link: in this case the <b>one</b> element is represented in the Project Browser and in any diagram to which it has been added; changes to the element are reflected in all diagrams in which it is shown</li> <li>• <b>Instance</b> - Paste a <b>copy</b> of the element as a separate instance of the original element, classified by the original element; this is useful when creating multiple instances of a Class in a Sequence diagram or Communication diagram</li> <li>• <b>Invocation</b> - Paste a copy of a State Machine or Activity element as an invocation of the source element; the resulting element is a State object or Action - respectively - with the source element as its classifier</li> <li>• <b>Property</b> - Paste a <b>copy</b> of the element as a separate Object instance</li> <li>• <b>Attribute</b> - (Available when dragging a classifier onto an element that supports attributes) Create a new attribute classified by the dragged classifier; the attribute is automatically selected in in-place editing so that you can add the attribute name</li> <li>• <b>Child</b> - Paste the element as a child of the copied element - that is, as a new element (which you are prompted to name) with a Generalization connector back to the copied element; this is very useful when you have a Class library or framework from which you inherit new forms - for example, you can paste a Hashtable as "MyHashtable" which automatically becomes a child of the original Hashtable</li> </ul> <p>Used with the Override Parent Operations feature, this is a quick</p>	<p><a href="#">Classifiers and Instances</a>  </p> <p><a href="#">Edit Element Item Name</a>  </p> <p><a href="#">Override Parent Operations</a>  </p>

Field/Button	Detail	See also
	<p>way to create new structures based on frameworks such as the Java SDK and the .NET SDK</p>	
<b>Structural Elements</b>	<p>If the original element has any child elements (for example, if it is a composite element or nested element, or contains embedded elements) this field contains the value <b>Select</b>. Either:</p> <ul style="list-style-type: none"> <li>• Leave this value as is, so that when you click on the <b>OK</b> button the Structural Elements dialog displays, from which you can <b>select</b> the child elements to reproduce with the pasted element, or</li> <li>• Click on the drop-down arrow and select: <ul style="list-style-type: none"> <li>• <b>None</b> to not reproduce any child elements</li> <li>• <b>All</b> to reproduce all child elements on the original element, or</li> <li>• <b>Based on Instance &lt;diagram names&gt;</b> to reproduce the configuration of child elements on an instance of the original element on another diagram, if there are any other instances</li> </ul> </li> </ul> <p>The resulting Object element is created large enough to display the selected child elements laid out neatly within the Object border.</p>	<a href="#">Manage Structural Elements</a> [935]
<b>Copy Connectors</b>	<p>This checkbox is enabled when you select to paste the copied element as an instance, property or invocation.</p> <p>Select the checkbox to reproduce on the diagram any connectors between the original element and any other elements that have also been pasted to this diagram, as connectors between the <b>instances</b>.</p>	
<b>Select for all</b>	<p>(If dragging multiple elements onto the diagram at once.)</p> <p>Select the checkbox to apply the dialog settings to all multiple-selected elements being dragged onto the diagram, and to add the elements together.</p> <p>Leave the checkbox unselected to add the elements to the diagram one at a time, with the opportunity to change the dialog settings for each element.,</p>	
<b>Options</b>	<p>Click on this button to display a short menu from which you can select to:</p> <ul style="list-style-type: none"> <li>• Make your settings on this dialog the default for all drag and drop operations from the Project Browser, and</li> <li>• Display this dialog for drag and drop operations only if you press and hold ( <b>Ctrl</b> ) as you drag the element onto the diagram</li> </ul> <p>If you select the <b>Autosave selection as default</b> option, this deselects the <b>Auto Instance</b> checkbox on the Diagram &gt; Behavior page of the Options dialog. The automatic paste operation becomes whatever you have set on the Paste dialog, and is not overridden by the paste-as-instance imposed by the <b>Auto Instance</b> checkbox.</p> <p>If you select the <b>Use Ctrl + Mouse drag to display this dialog</b> option, you can drag elements from the Project Browser without displaying the dialog;</p>	<a href="#">Diagram Behavior</a> [625]

Field/Button	Detail	See also
	the paste operation complies with whatever settings you used the last time you opened the dialog. To change these settings, hold <b>Ctrl</b> as you drag an element, and edit the field values on the dialog.	
<b>OK</b>	Click on this button to complete the paste operation and close the dialog.	
<b>Cancel</b>	Click on this button to cancel the paste operation and close the dialog. If you have selected multiple elements, the first-selected element is not pasted, but the remainder are.	
<b>Cancel All</b>	(If dragging multiple elements onto the diagram at once.) Click on this button to cancel the insertion of all the multiple-selected elements being dragged onto the diagram.	
<b>Help</b>	Click on this button to display this Help topic.	

### Notes

- Not all paste options apply to all element types; for many types, the only option available is a **Link**
- When an element dragged from the Project Browser is pasted into the diagram, its default style and size is applied rather than any alternative styles and sizes used for instances of the element in other diagrams

### Learn more

- [Paste Packages](#) <sup>[835]</sup>
- [Connect Requirements](#) <sup>[1779]</sup>
- [Create Object From Attribute](#) <sup>[1008]</sup>
- [Make Linked Element A Local Copy](#) <sup>[931]</sup>
- [Copy Elements Between Diagrams](#) <sup>[912]</sup>

#### **5.5.6.3.1 Paste Packages**

When you drag a Package element from the Project Browser onto the current diagram, you can paste it into the diagram as one of a number of different objects, by simple selection from a menu.

**Access**    **Drag existing Package(s) onto the diagram**  
**(If the Place Package on Diagram menu does not display, press ( Ctrl ) while dragging the Package)**

### Select Paste options

Option	Detail	See also
<b>Package element</b>	Paste the Package as an element representing the Package in the Project Browser (a simple link): in this case any changes to the Package are reflected in all diagrams in which it is shown.	<a href="#">Package</a> <sup>[1382]</sup>
<b>Report Specification linked to package</b>	Paste the Package as a Report Specification linked to the Package; a Report Specification is a stereotyped Document Artifact element from which you can generate a report and view the report as a linked document of that element.	<a href="#">Generate Documentation</a> <sup>[2644]</sup> <a href="#">Generate Report to an Artifact Element</a> <sup>[2680]</sup>
<b>Graph based on package contents</b>	Paste the Package as a Pie Chart or Bar Chart that you can edit to graphically represent some aspect of the Package content.	<a href="#">Chart Elements</a> <sup>[2763]</sup>
<b>Matrix profile</b>	Paste the Package as a Matrix Specification object.	<a href="#">Create Matrix Profile</a> <sup>[1744]</sup>
<b>Package as List</b>	Paste the Package as a Model View chart element that lists the elements within the Package and shows the type (by icon), name, Scope, Status and Author of each element.	<a href="#">Define a Model View Chart</a> <sup>[2765]</sup>



#### Learn more

- [Paste from Project Browser](#) <sup>[833]</sup>

### 5.5.6.4 Copy And Paste Diagram Element


#### How to

To copy a diagram element

Step	Instruction	See also
<b>1</b>	Select the element(s) to copy.	
<b>2</b>	For multiple elements: <ul style="list-style-type: none"> <li>• Right-click to open the context menu and select the <b>Copy</b> menu option</li> <li>• Press <b>Ctrl+C</b></li> <li>• Click on the <b>Copy</b> button (  ) in the Default Tools toolbar, or</li> <li>• Click on the <b>Cut</b> button (  ) in the Default Tools toolbar</li> </ul>	
<b>3</b>	For single elements, select the <b>Edit   Copy</b> menu option or alternatively press ( <b>Ctrl+C</b> ).	

Step	Instruction	See also

To paste diagram elements

Step	Instruction	See also
1	Open the diagram to paste into.	
2	<ul style="list-style-type: none"> <li>Click on the <b>Paste</b> button (  ) in the Default Tools toolbar, or</li> <li>Right-click on the diagram background to open the diagram context menu</li> </ul>	
3	Select either the <b>Paste Object(s) as New</b> menu option (completely new element) or the <b>Paste Object(s) as Link</b> menu option (reference to the existing element).	

#### Notes

- The **Date Created** and **Time Created** parameters of a pasted-as-new element are set to the current date and time; the parameters for a linked element remain the same as the copied element


#### 5.5.6.5 Move/Copy Elements To & From Floating Diagrams

If you have converted a tabbed diagram in the Diagram View into a floating diagram, you have a number of context menu options to move or copy elements between that diagram and another. The other diagram can be either another floating diagram or a tabbed diagram.

**Access** Drag element(s) between two diagrams and release mouse button

#### Move or copy selected elements




Option	Function Keys	Description	See also
<b>Move to this diagram</b>	<b>Drag+Shift</b>	Cuts the selected element or elements from the source diagram and pastes them into the target diagram.  If the moved elements are held in a different Package, when the elements are pasted each has a label underneath identifying the parent Package.	
<b>Create link on this diagram</b>	<b>Drag+Alt</b>	Creates an element or elements on the target diagram, each as an instance of a selected element on the source diagram.	
<b>Copy as New</b>	<b>Drag+Ctrl</b>	Creates an element or elements on the target diagram, each	<a href="#">Copy</a>

Option	Function Keys	Description	See also
<b>Object(s)</b>		as a copy of a selected element on the source diagram.  The Paste Element(s) as New dialog displays, on which you enter a new name for each element to identify it as a new, separate element. Click on the <b>Paste</b> button to complete the create task.	<a href="#">Elements Between Diagrams</a> 
<b>Instance (Object)</b>		Creates an element or elements on the target diagram, each as an object instance or invocation of an element on the source diagram. The source element is the classifier for the new object element.	
<b>as New Child (Generalization)</b>		Creates a new child element on the target diagram, as a generalization of the element of the source diagram.  The Element Name dialog displays for each copied element, on which you enter a new name for the element to identify it as a new, separate element.  Click on the <b>OK</b> button to complete the create task.	

#### Notes

- Multiple elements are processed individually, so if one element is 'illegal' the other elements are still copied and/or transferred
- If you select multiple elements to copy as instances or generalizations, those elements must be of the same type and stereotype
- Embedded elements are moved or copied only if they are specifically selected with the parent element

#### Learn more

- [Diagram Tabs](#)  - for information on floating diagrams
- [Window Menu](#)  - for information on floating diagrams
- [Paste from Project Browser](#)  - for information on copying, pasting and moving elements using the Project Browser

### 5.5.6.6 Place Related Elements on Diagram

To find and place related elements on the current diagram, use the Relationships window.

**Access** **Element | Relationships** (Ctrl+Shift+2)

#### Topics

Topic	Detail	See also
<b>Usage</b>	Right-click on any connector in the list to open the context menu.	

Topic	Detail	See also
	<p>If an element is not present in the current diagram, the context menu contains the <b>Place Target Element in Diagram</b> option. This is useful when you are building up a picture of what an element interacts with, especially when reverse engineering an existing code base.</p> <p>Select the <b>Place Target Element in Diagram</b> option.</p> <p>Move the cursor to the required position in the diagram and click to place the element. Alternatively, press ( <b>Esc</b> ) to cancel the action.</p>	

### 5.5.6.7 Delete Diagram

#### Delete a diagram from your model

Step	Action	See also
1	In the Project Browser, right-click on the diagram to delete. The context menu displays.	
2	Select the <b>Delete '&lt;diagram name&gt;'</b> menu option. A confirmation prompt displays.	
3	Click on the <b>OK</b> button to confirm the delete.	

#### Notes

- In Enterprise Architect there is no Undo feature for deleting diagrams, so be certain that you want to delete a diagram before you do so
- When you delete a diagram, you do not delete the elements in the diagram from the model
- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Diagrams** permission to create new diagrams
- You can also delete multiple diagrams from the Project Browser, by holding ( **Ctrl** ) or ( **Shift** ) while you select them, then right-clicking on one of them and selecting the **Delete selected items** context menu option

#### Learn more

- [Permissions List](#)<sup>[329]</sup>

### 5.5.6.8 Rename Diagram

#### How to

To rename a diagram

Step	Action	See also
1	Open the Diagram Properties dialog by double-clicking on the diagram background, or by selecting the <b>Diagram   Properties</b> menu option.	
2	In the <b>Name</b> field on the General tab, type the new name for your diagram.	
3	Click on the <b>OK</b> button to save changes.	

### 5.5.6.9 Change Diagram Type

If necessary, you can change one type of diagram to another type. This is useful if you have either made a mistake in selecting the diagram type to begin with, or if the purpose and nature of a diagram changes during analysis.

**Access** **Diagram | Advanced | Change Type**

#### Change a diagram type

Step	Action	See also
1	Open the diagram to change.	
2	Select the <b>Change Type</b> menu option. The Change Diagram Type dialog displays.	
3	Select the required diagram type.	
4	Click on the <b>OK</b> button to save changes.	

#### Notes

- Some diagram types do not transfer to others; for example you cannot change a Class diagram into a Sequence diagram



### 5.5.6.10 Diagram Navigation Hotkeys

The diagram hotkeys enable you to quickly navigate to and select elements within a diagram. The following table details the key combinations and their functionality.

Hotkey Command	Action
<b>Shift+Arrow</b> , Element(s)selected	Move the selected element(s) by increments.
<b>Arrow</b> , No element selected	Scroll around the diagram.
<b>Esc</b>	Clear the current selection.
<b>Tab</b>	Select the first element in the diagram if none currently selected.
<b>Shift+Tab</b>	Select the element with the next-lowest Z Order number (the next furthest back in the diagram).
<b>Shift</b> +click	Add the clicked element to the current selection.
<b>Ctrl</b> +click	Add the clicked element to the current selection.
<b>Ctrl+Shift</b> +drag	Pan the diagram.
<b>Alt+G</b>	Select the item in the Project Browser and give it focus.

### 5.5.6.11 Copy Image to Disk

You can copy a diagram image to a disk file in the following formats:

- Windows bitmap (256 color bitmap)
- GIF image
- Windows Enhanced Metafile (standard metafile)
- Windows Placeable Metafile (older style metafile)
- PNG format
- JPG
- TGA

#### How to

To copy a diagram image to file

Step	Action	See also
1	Open the diagram to save.	
2	Select the <b>Diagram   Save as Image</b> menu option, or press ( <b>Ctrl+T</b> ).	
3	When prompted, enter a name for the file and select an image format.	
4	Click on the <b>OK</b> button.	


#### Notes

- Enterprise Architect clips the image size to the smallest bounding rectangle that encompasses all diagram elements

### 5.5.6.12 Copy Image to Clipboard

You can copy diagram images onto the MS Windows clipboard and paste them directly into MS Word or other applications.

#### Copy an image to the clipboard

Step	Action	See also
1	Open the diagram to copy.	
2	Select the <b>Diagram   Copy Image to Clipboard</b> menu option, or press ( <b>Ctrl+B</b> ).	
3	Click on the <b>OK</b> button.  The diagram has been copied to the clipboard and can now be pasted into compatible applications or into another diagram.  You can set the clipboard format on the Options dialog ( <b>Tools   Options   General</b> ). Enterprise Architect supports bitmap or metafile format.	<a href="#">General Options</a> 

### 5.5.6.13 Copy (Duplicate) Diagram

As you develop the diagrams in your model, you might find that some of the diagrams use many of the same elements and have a similar format. You can copy and replicate the complete diagrams for minor modification, resulting in a duplicate diagram with either:

- Links back to the original diagram elements, **unless** those elements are Text, Note, Hyperlink, Lifeline or Object elements that are specific to the diagram and that are copied (*shallow mode*) or
- Complete copies of all elements in the diagram, other than Packages (*deep mode*) or
- Complete copies of all elements in the diagram that have the **same parent** as that of the diagram, and

links back to the original diagram elements for all other elements on the diagram (*smart mode*)

**Access** **Project Browser | right-click on diagram | Copy Diagram** then  
**Project Browser | right-click on Package | Paste Diagram**

#### Duplicate a diagram

Step	Action	See also
1	In the Project Browser, select the diagram to copy and select the <b>Copy Diagram</b> context menu option.	
2	Select the Package to host the new diagram, and select the <b>Paste Diagram</b> context menu option. The Copy Diagram dialog displays.	
3	In the <b>Name</b> field, type the name for the new diagram.	
4	In the Type of copy panel, click on the radio button for the type of copy you require; either: <ul style="list-style-type: none"> <li>• <b>Shallow</b></li> <li>• <b>Deep</b>, or</li> <li>• <b>Smart</b></li> </ul>	
5	Click on the <b>OK</b> button. The new diagram is automatically created with linked or new elements arranged as in the original diagram. All connectors are also copied between diagram elements where appropriate.	

#### Notes

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Diagrams** permission to copy diagrams
- You can also paste a copied diagram as a child of a composite element
- For a diagram copied in *shallow mode*, any changes to the properties of the original elements are reflected only in those elements that are copied as simple links in the pasted diagram
- For a diagram copied in *deep mode*, any changes to the properties of the original elements are not reflected in the elements in the pasted diagram
- For a diagram copied in *deep mode*, Package elements are copied as simple links only; you cannot paste as new
- For a diagram copied in *smart mode*, any changes to the properties of the original elements are reflected only in those elements that are copied as simple links in the pasted diagram

- Element position and size are independent in all copy modes

#### Learn more

- [Permissions List](#)<sup>[329]</sup>
- [Composite Elements](#)<sup>[936]</sup>

### 5.5.6.14 Z Order Elements

Z Order refers to an element's depth in the diagram perspective, and thus influences which elements appear in front of others and which appear behind.

#### Set the Z Order of an element

Step	Action	See also
1	Right-click on the element in the diagram.	
2	Select the <b>Z-Order</b> menu option.	
3	The displayed commands enable you to move the selected element back, forward, to the front of all other elements or behind all other elements.	

#### Notes

- This option enables you to change the Z order of a single element approximately in relation to the rest of the diagram; you can also select the **Modify Z Order** option from the *diagram* context menu, to review and adjust the Z Order of each element on the diagram precisely relative to each other element
- On a diagram, you can move the focus from element to element according to Z Order, by pressing **Shift+Tab**; each time you press these keys, the element with the next-lowest Z Order number is selected

#### Learn more

- [Diagram Context Menu](#)<sup>[778]</sup>

### 5.5.6.15 Set the Default Diagram

A project might have a default diagram. If set, this diagram loads when Enterprise Architect first opens the model. It is often convenient to place hyperlinks to other diagrams and resources on the default diagram, thus creating a Home Page for your model.

**Access**   **Diagram | Advanced | Make Model Default**

#### Notes

- Once you have specified a default diagram, the **Home** icon on the Diagram toolbar takes you back to that diagram from your current location in the model
- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of

Enterprise Architect, if security is enabled you must have **Manage Project Settings** permission to set the model default diagram

#### Learn more

- [Diagram Advanced Menu](#)<sup>[95]</sup>
- [Diagram Toolbar](#)<sup>[138]</sup>

### 5.5.6.16 Open Package From Diagram

#### How to

To open a Package from within a diagram follow the steps below:

Step	Action	See also
1	Open a diagram that shows the Package to open.	
2	Right-click on the Package element to open the context menu.	
3	Select the <b>Open Package</b> option. Alternatively, press ( <b>Ctrl+K</b> ).	

#### Notes

- Enterprise Architect finds the Package default diagram and opens it for you; this is the first available diagram in the Package, selected in alphabetical order - for example, a diagram called Alpha in a child Package or element several levels down opens before a diagram called Beta immediately under the selected Package





### 5.5.6.17 Feature Visibility

When you set up the characteristics and features of an element, you can set a collection of options to **display** or **hide the features** of that element on a specific diagram. You can also filter the attributes and operations that are shown by **scope** - such as hiding all protected attributes - or select to hide or show **specific** attributes and operations. The visibility you set applies only to the current diagram, so a Class can appear in one diagram with all features displayed, and in another with features hidden.

It is possible to hide or show inherited attributes, operations, responsibilities, constraints and Tagged Values for elements that support those features. When the system displays **inherited** features, it creates a merged list from all generalized parents and from all realized interfaces. If a child Class redefines something found in a parent, the parent feature is omitted from the list.

**Access** In diagram, right-click element | **Feature and Compartment Visibility**, or click element, then **Element | Feature Visibility** (**Ctrl+Shift+Y**)

#### Customize feature visibility

Field/Button	Action	See also
<b>Attribute Visibility</b>	Select the checkbox against each scope that should be visible, and clear the checkbox against each scope that should not be visible.	<a href="#">General Properties of Attributes</a> 
<b>Operation Visibility</b>	Select the checkbox against each scope that should be visible, and clear the checkbox against each scope that should not be visible.	<a href="#">General Properties of Operations</a> 
<b>Show and Custom</b>	<p>Use these to show or hide specific operations or attributes.</p> <p>If you <b>select</b> the <b>Show</b> checkbox, the <b>Custom</b> button displays the Show Features in Diagram dialog; if you <b>deselect</b> the checkbox, the button displays the Suppress Features in Diagram dialog.</p> <p>The two dialogs are identical. Select the checkboxes of specific features to <i>show</i> or <i>hide</i> on the element in the diagram.</p> <p>You can also use the:</p> <ul style="list-style-type: none"> <li>• <b>Filter by Scope</b> drop-down arrow to filter the list to show only features that have a specific scope</li> <li>• <b>All</b> button to select all operations or attributes in the list</li> <li>• <b>None</b> button to clear the checkboxes against all selected operations or attributes in the list</li> </ul> <p>The <b>Show</b> checkbox, if selected, overrides the selections to display attributes or operations in the element, other than those specifically selected in the Show Features in Diagram dialog.</p>	
<b>When Resizing Elements</b>	<p>Select the appropriate option for resizing the Class, object or table to manage very wide diagram objects.</p> <p>The <b>Resize to longest Feature</b> radio button defaults to selected, so that the minimum width for a diagram object is determined by its longest displayed attribute, operation or other compartment value.</p> <p>If necessary, you can select the <b>Wrap Features</b> radio button (so that any long features are wrapped onto multiple lines) or <b>Truncate Features</b> radio button (so that long features are not displayed in full).</p>	
<b>Inherited Features</b>	If you want to display inherited attributes and/or operations as well as directly owned ones, select one or both of the <b>Show Attributes</b> and <b>Show Operations</b> checkboxes.	<a href="#">Display Inherited Attributes</a>   <a href="#">Display Inherited Operations</a> 
<b>Hide Stereotyped Features</b>	<p>If you do not want to show an object having a particular stereotype, type that stereotype name - exactly as defined - in this field; the field is case-sensitive.</p> <p>If you want to specify more than one stereotype, separate the stereotype names with a comma.</p>	

Field/Button	Action	See also
<b>Show element compartments</b>	<p>Select the checkbox against each compartment to display in the element on the diagram.</p> <p>Some technologies such as SysML define additional compartments in the element, such as for Ports, Parts and Flow Ports. Visibility of these compartments will also be controlled by a checkbox matching the name of the compartment.</p> <p>If you select the <b>Fully Qualified Tags</b> checkbox, the <i>Tags</i> compartment will contain the full provenance of each Tagged Value (if created in Enterprise Architect release 7.1 or later) where the same Tagged Value is used several times in different contexts with different values. The description in the <i>Tags</i> compartment reads:</p> <pre>&lt;Profile&gt;::&lt;Stereotype&gt;::&lt;Tagged Value name&gt;=&lt;Value&gt;</pre> <p>For example:</p> <pre>BPMN::Activity::Activity Type = Task</pre> <p>You can display an <b>informal</b> internal or composite diagram of child elements such as Ports, Parts and Classes within a compartment of the parent element, by selecting the <b>Structured Compartment</b> checkbox. The Structured Compartment acts as a container for the child elements; Classes cannot be dragged out of the compartment, and Ports and Parts are confined to the borders of the compartment.</p> <p>If you want to apply a structure and formal relationships to the child elements, create a proper Composite Structure Diagram containing the elements and select the <b>Show Composite Diagram in Compartment</b> context menu option.</p>	<p><a href="#">Add Custom Compartments to Element</a> <sup>[1604]</sup></p> <p><a href="#">Composite Elements</a> <sup>[936]</sup></p>
<b>Element Notes</b>	<p>Select the <b>Show Notes</b> checkbox to display, in the <i>Notes</i> compartment on the element in the diagram, the text that has been typed into the <b>Notes</b> field of the element Properties dialog.</p> <p>This checkbox also enables the <b>maximum chars</b> field, which defaults to 1000 as the number of characters of notes text that can be displayed. Overtyping this value to display less text or more text, as you need.</p> <p>The setting applies only to the selected elements on the diagram, so you can display full notes for the selected element whilst the other elements on the diagram have no notes text.</p> <p>You can also select the <b>Render Formatted Notes</b> checkbox to display the text on the diagram as formatted using the Notes field toolbar.</p>	<a href="#">Notes Toolbar</a> <sup>[1143]</sup>
<b>Runstate</b>	Select the checkbox if you want to hide the object runstate of the element on the diagram.	<a href="#">Object State</a> <sup>[1381]</sup>
<b>Type</b>	(Available for Ports and Parts only.) Select the checkbox if you want to	

Field/Button	Action	See also
	display the element type.	
<b>OK</b>	Click on this button to save your settings. The diagram redisplay with the appropriate level of feature visibility.	


#### Notes

- To show features for element types that do not have visible compartments, such as Use Cases and Actors, right-click on the diagram object to display the context menu and select the **Advanced | Use Rectangle Notation** option

#### 5.5.6.18 Insert Diagram Properties Note

Properties of a diagram can be displayed on screen within a custom text box. You can move this text box around and change its appearance. You cannot change what the text box says.

#### Topics

Topic	Detail	See also
<b>Usage</b>	<p>To create the note, drag the <b>Diagram Notes</b> icon from the Common page of the Toolbox onto the diagram.</p> <p>Alternatively:</p> <ul style="list-style-type: none"> <li>Select the <b>Diagram   Add Diagram Property Note</b> menu option, or</li> <li>Click on the <b>Diagram Properties Note</b> button on the UML Elements toolbar and click on the diagram</li> </ul> 	

#### Notes

- This is not the same as the diagram details note, which displays in the top left corner of the diagram if the **Show Diagram Details** checkbox is selected on the Diagram Properties dialog; you cannot move the diagram details, nor change the appearance - to hide the diagram details, deselect the checkbox

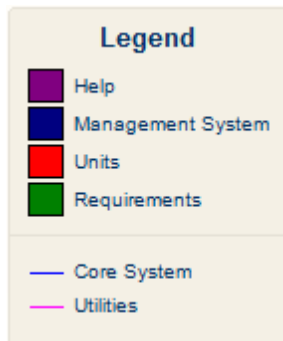
#### Learn more

- [Configure Default Appearance](#)<sup>[927]</sup>
- [Configure Diagram Display](#)<sup>[825]</sup>




### 5.5.6.19 Create Legends

A *Legend* shape identifies colors and styles you have used to group other elements on the diagram. You can use the Legend to assist in distinguishing different elements, connectors or systems on the diagram. For example, the Legend could show that all elements concerned with the management system are shaded in blue, and all outcomes connectors are shown in red. The Legend displays as a key to the diagram, with the filled shape styles first and the lines and connector styles underneath.



You add a Legend to the diagram, then edit it to add Legend elements, which define the colors and styles used in the diagram.

To add a Legend to a diagram, drag the *Diagram Legend* element from the Common page of the Toolbox onto the diagram (or click on the **New Diagram Legend** icon (  ) on the UML Elements toolbar, and click on the diagram).

To edit the Legend either double-click on the Legend or Right-click on the Legend and select the **Properties** context menu option.

#### 5.5.6.19.1 Legend Properties


**Access** Drag the Diagram Legend element from the Common page of the Toolbox onto the diagram

#### **Use to**

- Add a Legend to a diagram

#### **Reference**

Field	Usage	See also
<b>Name</b>	Indicates a name for an element displayed within a legend.	
<b>Fill Color</b>	Indicates a fill color for an element within a legend.	
<b>Line Color</b>	Indicates a line color for an element within a legend.	
<b>Line Thickness</b>	Denotes the line thickness of an element within a legend.	

<b>Style Options</b>	Allows further alteration options to the legend appearance.	<a href="#">Style Options</a> 
<b>Fill</b>	Indicates that a fill element is to be created/modified.	
<b>Line</b>	Indicates that a line element is to be created/modified.	
<b>Hand (Up)</b>	If more than one element is present within a legend, this button moves a highlighted element up one entry within the description list.	
<b>Hand (Down)</b>	If more than one element is present within a legend, this button moves the highlighted element down one entry within the description list.	
<b>New</b>	Add a new element to the legend.	
<b>Save</b>	Saves an element and adds it to the description list.	
<b>Delete</b>	Deletes an element from the description list.	
<b>Description List</b>	Specifies the current elements created within the legend.	

#### Notes

- The Legend dialog enables you to add, delete, modify or re-sequence Legend elements; use the Fill tab to define the Legend elements for shapes, then click on the Line tab to define Legend elements for lines and connectors

#### **5.5.6.19.2 Style Options**

Click on the **Style Options** button ( ... ) to display the Style Options dialog, on which you can modify a Legend title, font size, background color and border color. If you choose default options for the colors, the Legend automatically assumes colors based on the diagram background color.

Access    **Legend dialog > (...)**

#### Modify appearance of a Legend

Field	Action	See also
<b>Legend Title</b>	Indicates the title of the Legend.	

<b>Heading Size</b>	Indicates the size of the font for a Legend.	
<b>Font Color</b>	Indicates the font color the Legend is to use.	
<b>Background</b>	Indicates the background color the Legend is to use.	
<b>Border Color</b>	Indicates the border color the Legend is to use.	

### 5.5.6.20 Autosize Elements

You can autosize an element or group of elements in a diagram to the default size for the element type (for a Class, 90 x 70 pixels at 100% zoom). However, if the element contains more information than the default size can show (such as a long name, long attributes or additional compartments) the autosize option resizes the element to the minimum size for revealing the information.

The size change effectively operates around the mid point of each element, so the layout and size of the diagram do not change. Automatically changing the layout of a diagram is a different process.

#### Autosize elements

Step	Action	See also
<b>1</b>	Select the elements to resize; (press <b>Ctrl+A</b> to select all).	
<b>2</b>	Either: <ul style="list-style-type: none"> <li>Right-click on any of the elements and, on the context menu, select the Autosize menu option, or</li> <li>Press <b>Alt+Z</b></li> </ul>	

#### Notes

- Not all elements resize: elements such as Events remain the same; Timing and Sequence diagrams (where position is crucial) are unchanged; and elements added from a profile or Shape Script maintain any size definitions imposed by the profile
- With an element image created with a Shape Script that contains a defSize command, **Autosize** returns the element to the defSize value and not the element default size

#### Learn more

- [Lay Out a Diagram Automatically](#)<sup>[89]</sup>
- [defSize command](#)<sup>[159]</sup>

### 5.5.6.21 Swimlanes

**Swimlanes** are vertical or horizontal bands in a diagram, which divide the diagram into logical areas or **partitions**. You can apply them to all Enterprise Architect diagram types. Activities relating to particular entities within the model (such as the User, or the back end Repository) can be placed within the same Swimlane to indicate their association. You set up the Swimlanes using the Swimlanes, Matrix and Kanban dialog.

**Access** [Diagram | Swimlanes, Matrix and Kanban > Swimlanes](#)

#### Options

Field/Option/ Button	Action	See also
<b>Active</b>	Select this checkbox to make the Swimlane facilities active on the diagram, as opposed to the Matrix or Kanban facilities.	<a href="#">Swimlanes Matrix</a> <sup>[854]</sup> <a href="#">Kanban Facilities</a> <sup>[857]</sup>
<b>Orientation</b>	Click on the drop-down arrow and select the orientation of the Swimlanes: <b>Vertical</b> or <b>Horizontal</b> .	
<b>Line Color</b>	Click on the drop-down arrow and select the color in which to display all Swimlane borders.  If the color you require is not shown, click on the <b>Other</b> button and either select from a wider range of standard colors or define a custom color.	<a href="#">Standard Colors</a> <sup>[616]</sup>
<b>Font Color</b>	Click on the drop-down arrow and select the color in which to display all Swimlane text.  If the color you require is not shown, click on the <b>Other</b> button and either select from a wider range of standard colors or define a custom color.	
<b>Line Width</b>	Click on the drop-down arrow and select the line width of all Swimlane borders - either <b>1</b> (thinnest), <b>2</b> or <b>3</b> (thickest).	
<b>Locked</b>	Select this checkbox to prevent the Swimlanes in the diagram from being made narrower or wider.	
<b>Bold Font</b>	Select this checkbox to display all Swimlane text in bold.	
<b>Hide Classifier</b>	Select this checkbox to block all Swimlane classifier names from being shown on the diagram.	

<b>Hide Names</b>	Select this checkbox to block all Swimlanes names from being displayed on the diagram.	
<b>Show Names in Title Bar</b>	(If the <b>Orientation</b> field is set to <b>Vertical</b> .) Select this checkbox to display each Swimlane name in the title bar, above the Swimlane.	
<b>Swimlanes</b>	This panel lists the currently defined Swimlanes, in the sequence (top to bottom) in which they are displayed on the diagram (left to right, or top to bottom).	
<b>New</b>	Click on this button to begin defining a new Swimlane for the diagram.  The Swimlane Details dialog displays, on which you provide the name, classifier and background color of the Swimlane.	<a href="#">Swimlane Details</a> <small>853</small>
<b>Hand (Up)</b>	Click on this button to move a selected Swimlane one position up in the <b>Swimlanes</b> list.	
<b>Hand (Down)</b>	Click on this button to move a selected Swimlane one position down in the <b>Swimlanes</b> list.	
<b>Modify</b>	Click on this button to change the name, classifier or background color of a selected Swimlane.  The Swimlane Details dialog displays.	<a href="#">Swimlane Details</a> <small>853</small>
<b>Delete</b>	Click on this button to delete a selected Swimlane from the <b>Swimlanes</b> list and from the diagram. A prompt displays for you to confirm the deletion.	
<b>OK</b>	Click on this button to save and apply your changes, and close the dialog.	
<b>Cancel</b>	Click on this button to abort your changes, and close the dialog.	

**Notes**

- If you set a background color for a swimlane, it takes on the same shading profile as the main diagram background

**5.5.6.21.1 Swimlane Details**

You use the **Swimlane Details** dialog to define a Swimlane on a diagram.

**Access** **Diagram | Swimlanes, Matrix and Kanban > Swimlanes: New** or **Modify**

Options

Field/Options/ Button	Action	See also
<b>Name</b>	Type a name for the Swimlane.	
<b>Classifier</b>	(Optional) Click on the <b>Browse ( ... )</b> button to browse for and select a classifier for the Swimlane.  To delete a previously-assigned classifier, click on the <b>Browse</b> button and select the <b>&lt;none&gt;</b> option.	<a href="#">Select &lt;Item&gt; Dialog</a> <sup>[994]</sup>
<b>Back Color</b>	Click on the drop-down arrow and select the color in which to display the Swimlane background.  If the color you require is not shown, click on the <b>Other</b> button and either select from a wider range of standard colors or define a custom color.	<a href="#">Standard Colors</a> <sup>[616]</sup>
<b>OK</b>	Click on this button to save and apply your changes, and close the dialog.	
<b>Cancel</b>	Click on this button to abort your changes, and close the dialog.	

Learn more

- [Swimlanes](#)<sup>[852]</sup>

**5.5.6.22 Swimlanes Matrix**

On any type of Enterprise Architect diagram, you can develop a **Swimlanes Matrix** to divide your diagram into cells at the intersections of columns and rows. This Matrix is based on the Zachman Framework. The top left cell of the Matrix contains the heading text, the cell at the top of each column contains the column title text and the cell at the left of each row contains the row title text.

You can define only one Matrix Heading cell, but you can create as many row and column headings as you need. You can also leave the column width and row height adjustable on the diagram, or fix the positions so that the cell borders cannot be moved.

**Access** [Diagram | Swimlanes, Matrix and Kanban > Matrix](#)

Options

Field/Option/ Button	Action	See also
<b>Active</b>	Click on this checkbox to set the diagram to use the defined Matrix facilities, as opposed to any Swimlanes or Kanban facilities that have also been defined.	<a href="#">Swimlanes</a> <sup>[852]</sup> <a href="#">Kanban</a>

		<a href="#">Facilities</a> <sup>857</sup>
<b>Type</b>	Click on the drop-down arrow and select <b>Heading</b> , <b>Column</b> or <b>Row</b> to apply your field settings to the Matrix heading cell, a column heading or a row title, respectively.	
<b>Title #</b>	<p>Type in a title name. You can provide one, two or three title names per heading to, for example, group a set of columns or rows.</p> <p>There is no effective limit on the length of the text string but it <b>does not wrap</b>, so to display the full text you might have to drag the cell borders out to create a wide cell. Alternatively, you can divide the text string between the <b>Title</b> fields to reduce the overall cell width.</p>	
<b>Color</b>	<p>Click on the drop-down arrow and select the color in which to display the text in the heading, row or column title.</p> <p>If the color you require is not shown, click on the <b>Other</b> button and either select from a wider range of standard colors or define a custom color.</p>	<a href="#">Standard Colors</a> <sup>616</sup>
<b>Font</b>	<p>Click on this button to display the Font dialog, through which you define the font and style, size and effects of the text in the Matrix header, column and row titles.</p> <p>If you also change the text color here, it overrides the setting of the <b>Color</b> field (above).</p>	<a href="#">Set Element Font</a> <sup>949</sup>
<b>Hidden</b>	Select this checkbox to hide the corresponding Matrix header, row or column <b>Title</b> text on the diagram.	
<b>Back Color</b>	<p>Click on the drop-down arrow and select the color in which to display the heading cell, row or column background.</p> <p>If the color you require is not shown, click on the <b>Other</b> button and either select from a wider range of standard colors or define a custom color.</p>	
<b>Hide Item</b>	Select this checkbox to hide the Matrix heading cell, row or column on the diagram.	
<b>Model Profiles</b>	<p>Click on the drop-down arrow and select:</p> <ul style="list-style-type: none"> <li>An existing Profile to override the current contents of the dialog</li> <li><b>Empty Profile</b> to clear the fields so that you can define a matrix to save as a new Profile (see <b>Model Profiles: Save</b>, below)</li> </ul>	
<b>Model Profiles : Save</b>	<p>Click on this button to save the current dialog contents as a new Profile.</p> <p>A prompt displays for you to assign a name to the Profile.</p>	

<b>Model Profiles : Delete</b>	Click on this button to delete the currently-displayed user-defined Matrix Profile from the drop-down list.	
<b>Lock</b>	Select this checkbox to prevent the Matrix rows and columns in the diagram from being resized.	
<b>Line Widths</b>	Click on the drop-down arrow and select the line width of all row and column borders - either <b>1</b> (thinnest), <b>2</b> or <b>3</b> (thickest).	
<b>Matrix list</b>	<p>Lists the components of the current Matrix Profile, starting with the Heading and followed by the columns and then the rows. For each item, the list shows the first title (<b>Title 1</b>) whether it is visible on the diagram, and what type it is.</p> <p>The columns are listed in the order in which they are displayed, left to right, on the diagram.</p> <p>The rows are listed in the order in which they are displayed, top to bottom, on the diagram.</p>	
<b>Hand (Up)</b>	<p>Click on this button to move a selected column or row one position up in the Matrix list. You cannot move a Column item above the Heading item, or a Row item above a Column item.</p> <p>When you save the changes, on the diagram the selected column or row is moved left or up one place.</p>	
<b>Hand (Down)</b>	<p>Click on this button to move a selected column or row one position down in the Matrix list. You cannot move the Heading item below a Column item, or a Column item below a Row item.</p> <p>When you save the changes, on the diagram the selected column or row is moved right or down one place.</p>	
<b>New</b>	Click on this button to clear the dialog fields so that you can add a new column or row item to the Matrix.	
<b>Copy</b>	<p>Click on this button to copy a selected item from the list so that you can give it a <b>new Title 1</b> and edit it as a new item.</p> <p>If you change the item, it is added to the list when you click on the <b>Save</b> button.</p>	
<b>Delete</b>	Click on this button to delete a selected item from the list. The column or row is removed from the list and diagram, which both close up.	
<b>OK</b>	Click on this button to save and apply your changes, and close the dialog.	



<b>Cancel</b>	Click on this button to abort your changes, and close the dialog.	

### Notes

- When you define columns and rows, you define the header or title cells; the properties of these cells do not reflect on the matrix cells themselves - for example, the intersection cell of a column and row has a transparent background and therefore takes the color and shading effect of the diagram background
- You can transport the Swimlanes Matrix Profiles between models (as Diagram Matrix Profiles), using the **Export Reference Data** and **Import Reference Data menu** options
- By applying a Matrix Profile to a diagram, you replace the current profile; save the current profile to avoid losing it
- To size the rows and columns, drag the row and column borders on the diagram - elements placed inside each cell are shifted when sizing; to prevent the elements shifting, press and hold ( **Ctrl** ) while sizing

### Learn more

- [Export Reference Data](#)<sup>[376]</sup>
- [Import Reference Data](#)<sup>[380]</sup>

### 5.5.6.23 Kanban Facilities




Using the **Kanban** project management methodology, you might work on tasks or the objects of tasks organized on a chart according to the value of a project management property, such as Phase, Version or Status. Your work on the task is represented by **moving** the corresponding element to a different position on the chart, which reflects the change in status of the selected project management property of that element.

In Enterprise Architect, you can apply a form of this methodology to your project administration diagrams to monitor and manage the flow of work in a particular area. You can develop the chart to divide your diagram into a series of vertical lanes that represent the status of the elements in the organization of work. Dragging the elements from one lane to another automatically applies the change in value of the selected status indicator property of the element, which can also be a Tagged Value (of type **Enumeration** or **String**).

**Access** **Diagram | Swimlanes, Matrix and Kanban > Kanban**

### Define the Kanban chart

Field/Option/ Button	Action	See also
<b>Active</b>	Click on this checkbox to set the diagram to use the defined Kanban chart, as opposed to any Swimlanes or Swimlane Matrix facilities that have also been defined.	<a href="#">Swimlanes</a> <sup>[852]</sup> <a href="#">Swimlanes Matrix</a> <sup>[854]</sup>
<b>Line Color</b>	Click on the drop-down arrow and select the color in which to display	

	<p>lane borders.</p> <p>If the color you require is not shown, click on the <b>Other</b> button and either select from a wider range of standard colors or define a custom color.</p>	<a href="#">Standard Colors</a> 
<b>Font Color</b>	<p>Click on the drop-down arrow and select the color in which to display the text in the Kanban lane headings.</p> <p>If the color you require is not shown, click on the <b>Other</b> button and either select from a wider range of standard colors or define a custom color.</p>	
<b>A</b>	<p>Click on this button to display the Font dialog, through which you define the font and style, size and effects of the text in the Kanban lane headings.</p> <p>If you also change the text color here, it overrides the setting of the <b>Font Color</b> field (above).</p>	<a href="#">Set Element Font</a> 
<b>Title Color</b>	<p>Click on the drop-down arrow and select the color with which to fill the lane heading cells.</p> <p>If the color you require is not shown, click on the <b>Other</b> button and either select from a wider range of standard colors or define a custom color.</p>	<a href="#">Standard Colors</a> 
<b>Overfilled Color</b>	<p>Click on the drop-down arrow and select the color with which to fill a Kanban column if it contains more than the maximum number of elements. You set the maximum number of elements when you define the specific columns to work with, in the <b>Swimlanes</b> panel.</p> <p>If the color you require is not shown, click on the <b>Other</b> button and either select from a wider range of standard colors or define a custom color.</p>	
<b>Line Width</b>	Click on the drop-down arrow and select the line width of the Kanban lane borders - either <b>1</b> (thinnest), <b>2</b> or <b>3</b> (thickest).	
<b>Locked</b>	Select this checkbox to prevent the lanes in the diagram from being made narrower or wider. You cannot change the depth of the header cell even if this checkbox is clear.	
<b>Bold Font</b>	Select this checkbox to display the lane headings in bold (if their font definition does not already set them to bold).	
<b>Hide Names</b>	Select this checkbox to hide the names displayed in the lane heading cells (but not the heading cells themselves).	
<b>Hand Drawn</b>	Select this checkbox to display the Kanban lanes and the elements on	<a href="#">Configure</a>

	the diagram in Hand-drawn mode.	<a href="#">Diagram Display</a> <sup>825</sup>
<b>Show Names in Title Bar</b>	<p>Select this checkbox to display the Lane names in diagram title bar, above the lanes.</p> <p>This option is not available if the <b>Hide Names</b> checkbox is selected.</p>	
<b>Enable Overfill Highlight</b>	Select this checkbox to apply the overfilled color (above) to the lane if it contains more elements than the defined maximum number (see <b>New</b> , below).	
<b>Value</b>	<p>Click on the <b>Browse</b> button ( ... ) and select, firstly, the type of property to test for in assigning elements to lanes, either:</p> <ul style="list-style-type: none"> <li>• <b>Element</b> Properties</li> <li>• <b>Tagged Value</b> (Enumeration or String types only)</li> <li>• <b>None</b> (if you want to <b>manually</b> manage elements on the basis of some value that is not captured in the element properties or Tagged Values)</li> </ul> <p>If you select <b>Element</b>, also select one of the properties <b>Phase</b>, <b>Version</b>, <b>Priority</b> (for Requirement-type elements) and <b>Status</b>.</p> <p>If you select <b>Tagged Value</b>, also choose to either <b>Create</b> a new tag (and simply type in the name) or <b>select</b> an existing one. In the latter case, the Kanban tagged value selection dialog displays.</p> <p>On the dialog, click on the appropriate radio button to either select from a list of <b>Global Tagged Values</b>, or from a list of <b>Tagged Values from a selected element</b>; click on the <b>Select Element</b> button to browse for the element anywhere in the model.</p> <p>In the <b>Tagged Value</b> field, click on the drop-down arrow and select the reference Tagged Value from those listed. Click on the <b>OK</b> button.</p> <p>The <b>Value</b> field now displays either the element property or the Tagged Value you have selected.</p>	
<b>New</b>	<p>Click on this button to define a new lane on the Kanban chart. The Kanban Lane Details dialog displays:</p> <ul style="list-style-type: none"> <li>• In the <b>Name</b> field, click on the drop-down arrow and select the property value that the lane represents and that the lane title will reflect; the drop-down list generally provides a list of appropriate values, but you can type in <b>other values</b> if necessary</li> <li>• In the <b>Max Items</b> field, leave the value set to <b>0</b> if you do not want to limit the number of elements that can be placed in a lane, or type in the maximum number allowed; <b>if</b> this number is exceeded <b>AND</b> you have selected to highlight an overfill with a different lane color (see <b>Enable Overfill Highlight</b> above) the lane will change to that overfill color</li> <li>• Click on the <b>OK</b> button to add the lane definition to the Swimlanes list and the lane to the diagram; the lane name and fill limit display in the title cell at the top of the lane</li> </ul>	

<b>Modify</b>	Click on an entry in the Swimlanes list and click on this button to display the Kanban Lane Details dialog for that lane, and to change the details as necessary (see <b>New</b> , above).	
<b>Delete</b>	Click on an entry in the Swimlanes list and click on this button to delete the lane definition and clear the lane from the diagram. A prompt displays for you to confirm the deletion.	
<b>Swimlanes</b>	This panel lists the currently defined lanes, in the sequence (top to bottom) in which they are displayed on the diagram (left to right).	
<b>Hand (Up)</b>	Click on an entry in the Swimlanes list and click on this button to move the entry one line <b>up</b> in the list, and one lane to the <b>left</b> on the diagram.	
<b>Hand (Down)</b>	Click on an entry in the Swimlanes list and click on this button to move the entry one line <b>down</b> in the list, and one lane to the <b>right</b> on the diagram.	
<b>OK</b>	Click on this button to save and apply your changes, and close the dialog.	
<b>Cancel</b>	Click on this button to abort your changes, and close the dialog.	

### Notes

- The elements in a Kanban chart **individually** represent work in progress; you cannot **create** connectors between elements on the chart
- If you apply a Kanban chart to a current diagram, all connectors on the diagram are hidden; when you de-activate the chart, the connectors are shown again
- When you first set up a Kanban chart, any element that does not have a property value represented on the chart is moved to the right-hand side of the chart
- Moving an element into a lane **automatically sets** the appropriate property of that element to the value represented by the **lane**; for a Tagged Value, if the element does not have that tag, adding the element to the lane adds the tag and sets it to the lane value
- Elements on a Kanban chart are automatically adjusted to match the width of the lane they are in, both when the element is moved into a lane and when the lane width is changed
- All elements added to a Kanban chart initially have the same height and spacing, but the height can change as displayable information is added to an element

#### 5.5.6.24 Using the Image Manager

When you add an element to a diagram, it is rendered according to the standard UML notation. However, you can replace that representation with an image. For example, you could add a custom background image to a diagram, or display an image of a Router or PC on a UML element. You import, store and select these alternative images using the Image Manager.

**Access** Right-click on an element within a diagram | **Appearance** | **Select Alternate Image (Ctrl+Shift+W)**

To locate and display an image, click on individual image filenames, or press ( ↑ ) and ( ↓ ) to scroll through the list of images. As you highlight each image filename, the Preview panel changes to reflect the image. Double-click on the required image filename to display the image in full size.

### Image Manager Options

On the Image Manager dialog, the following buttons are available:

Button	Usage	Shortcut	See also
<b>View</b>	Display the selected image in full size.	<b>Alt+V</b>	
<b>Usage</b>	Display the Element Usage dialog, which lists the diagrams in which the selected image is used.		<a href="#">Element Usage</a> <sup>[910]</sup>
<b>Rename</b>	Change the filename of the selected image.  The option first warns you that the change would impact the other elements that use this image, and prompts you to confirm that you want to go ahead with the name change.  If you continue, a prompt displays for you to enter the new filename.		<a href="#">Changing Element Appearance</a> <sup>[948]</sup>
<b>Add New</b>	Browse appropriate directories to search for and import new images.  You can import images in .BMP, .PNG, .EMF, .WMF, .TGA, .PCX or .JPG format.  Internally, the images are stored in .PNG or metafile format to conserve space.  You can also add images by: <ul style="list-style-type: none"> <li>Importing a library of alternative images from the Sparx Systems website</li> <li>Copying the Image Library from another model using the Export and Import Reference Data facilities, and</li> <li>Capturing images from other sources into the clipboard and importing them into the Image Library (these are autonamed when they are added to the Image Library)</li> </ul> If necessary, you can change the file name using the <b>Rename</b> option (above).	<b>Alt+A</b>	<a href="#">Import Image Library</a> <sup>[864]</sup> <a href="#">Export Reference Data</a> <sup>[376]</sup> <a href="#">Import Reference Data</a> <sup>[380]</sup> <a href="#">Appearance Menu Section</a> <sup>[948]</sup>
<b>Update Selected</b>	Refresh the selected image; for example, after it has been modified.	<b>Alt+U</b>	
<b>Delete</b>	(Not available for an image from a deployed MDG	<b>Alt+D</b>	

Button	Usage	Shortcut	See also
	<p>technology.)</p> <p>Delete the selected image.</p> <p>A message displays to indicate how many elements use the image. Click on the <b>Continue</b> button to delete information about the image from those elements, which then revert to their previous appearance.</p>		
<b>Cancel</b>	Close the Image Manager dialog.		
<b>OK</b>	Confirm selection of the alternative image for the element selected in the diagram.	<b>Alt+O</b>	

### Notes

- For elements with lifelines, such as those used on Sequence diagrams, the Lifeline must remain intact to enable messages to be created between the elements; therefore such elements cannot have alternative images
- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Configure Images** permission to configure alternative element images
- There are two categories of image available through the Image Manager:
  - Images that have been imported into the model and are stored in the model
  - Images that are incorporated into a deployed MDG technology, which are not copied into the current model but are held in memory; the technology images can be identified by the technology name followed by the image file name, such as *MDGFrame::ovalshape.emf*, but can also be accessed by their short form (the name minus the technology name)
- If you are creating many elements of the same type that have a particular image, you should use a custom stereotype with an associated metafile
- If an element has an image, the element name is likely to be hidden; you can display the name **underneath** the image by right-clicking on the element and selecting **Appearance | Show Name Under Image**, and hide it again by selecting **Appearance | Hide Name Under Image**
- You can transport image files between models, using the **Export Reference Data** and **Import Reference Data** options

### Learn more

- [Create Custom Diagram Background](#)<sup>[863]</sup>
- [Permission List](#)<sup>[329]</sup>
- [Adding Images in MDG Technology](#)<sup>[1556]</sup>
- [Custom Stereotype](#)<sup>[1452]</sup>

#### 5.5.6.24.1 Create Custom Diagram Background

Enterprise Architect diagrams have a single-color 'wash' background that you can set to a solid color or a fade gradient down the screen. You set the color and whether to have a fade gradient using the Diagram - Gradients and Backgrounds page of the Options dialog. You can also select a system-defined tile background for your diagrams.

Alternatively, using the Image Manager dialog, you can create a **customized** tiled or non-tiled background for your diagrams.

##### Create a customized background

Step	Action	See also
1	Create a Boundary object from the Use Case Elements page of the Toolbox; do not use the Boundary element from any other section of the Toolbox.	<a href="#">System Boundary</a> <a href="#">Toolbox</a>
2	Stretch the Boundary to a size that can contain all of the elements you intend to place on the diagram, and drag it to the edges of the diagram workspace.	
3	Right-click on the Boundary element and select the <b>Z-Order   Send to Bottom</b> context menu option.  This ensures that the Boundary is not displayed in front of any other element in the diagram.	
4	Either: <ul style="list-style-type: none"> <li>Press ( <b>Ctrl+Shift+W</b> ), or</li> <li>Right-click on the Boundary to display the context menu, and select the <b>Appearance   Alternate Image</b> menu option</li> </ul>	
5	On the Image Manager dialog, select an appropriate image as the diagram background and ensure that the image size is large enough to span the required size of the diagram background.	<a href="#">Image Manager</a>
6	When you have selected the required image, click on the <b>OK</b> button.  Alternatively, you can copy an image from another source onto the Windows clipboard, right-click on the Boundary element in the Enterprise Architect diagram, and select the <b>Appearance   Apply Image From Clipboard</b> context menu option.	

##### Learn more

- [Diagram Gradients and Backgrounds](#)

#### 5.5.6.24.2 Import Image Library

Using the Image Library enables you to create attractive diagrams with custom images. A bundled clip art collection of UML-based images is available as an Imported Image Library, from the Sparx Systems website. Image libraries enable you to import a collection of images into the Image Manager in one process.

##### How to

To import the Image Library, follow the steps below (before starting, you must have a suitable Image Library file):

Step	Action	See also
1	Download the Image Library from the Sparx Systems website.	<a href="http://www.sparxsystems.com/resources/image_library.html">www.sparxsystems.com/resources/image_library.html</a>
2	Select the <b>Project   Model Import/Export   Import Reference Data</b> menu option. The Import Reference Data dialog displays.	
3	Locate the XML Image Library file to import using the <b>Select File</b> button. The file name is <i>ImageLibrary.xml</i> in the directory in which you saved the file.	
4	Select the data set containing the Image Library. Then click on the <b>Import</b> button.	

To use the images contained within the Image Library

Step	Action	See also
1	Create a diagram to associate with the images contained in the Image Library.	
2	Select the element to change from the default appearance to one of the images contained within the library.	
3	Either: <ul style="list-style-type: none"> <li>Press ( <b>Ctrl+Shift+W</b> ), or</li> <li>Right-click on the selected element and select the <b>Appearance   Select Alternate Image</b> context menu option</li> </ul>	



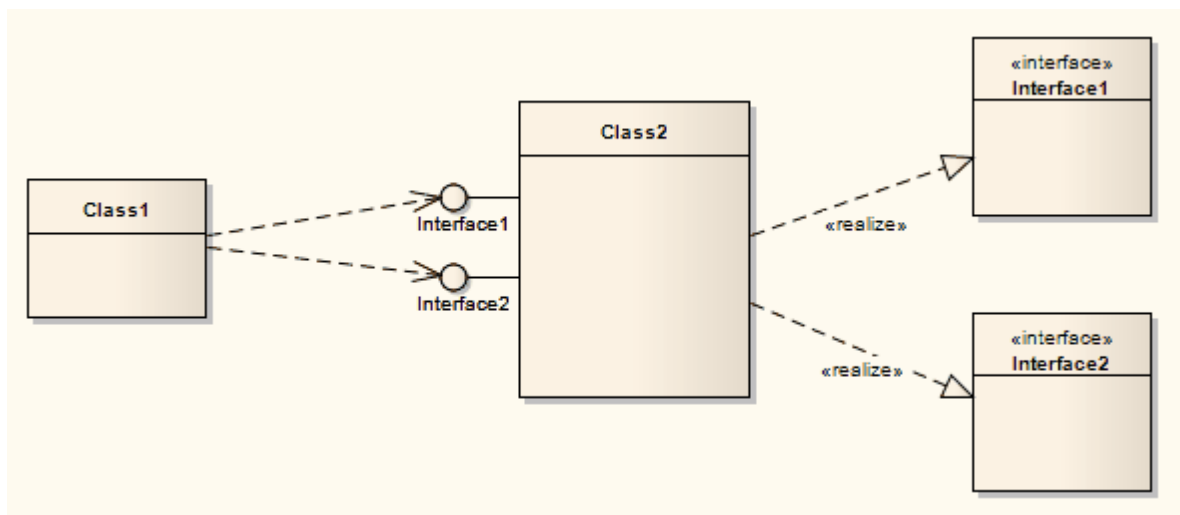
Step	Action	See also
4	On the Image Manager dialog, in the <b>Name</b> field highlight the appropriate image name and click on the <b>OK</b> button.	

### Notes

- Images contained within the Image Library are copyright of Sparx Systems, are only available for use in conjunction with Enterprise Architect, and are supplied on the understanding that they are not used under any other circumstance

### 5.5.6.25 Show Realized Interfaces of Class

You can display each interface directly realized by a Class as a 'lollipop' style interface node, which protrudes from the left-hand side of the Class. Connectors can be directly attached to the node, indicating usage of the interface part of the Class or component. See the example below:



### Topics

Topic	Detail	See also
<b>Explanation</b>	<p>In this example, <i>Class2</i> realizes <i>Interface1</i> and <i>Interface2</i> as represented by the interface nodes protruding from the Class.</p> <p><i>Class1</i> is dependent on these two interfaces, which is shown by the Dependency connectors linking to the nodes.</p> <p>To show nodes for the interfaces a Class realizes, as in the above diagram, right-click on the Class and select the <b>Features &amp; Properties   Show Realized Interfaces</b> context menu option.</p> <p>This setting only applies to the selected Class, and can be changed at any time.</p>	

### 5.5.6.26 Manage Object Labels

When you create certain elements or connectors, some of the properties of the created object are displayed on the diagram as labels offset from the object. For example, the following display as labels:

- Embedded element name
- Run state, attributes and operations (if they exist) of an element that has an alternative image
- Connector name, stereotype, constraints and/or source and target role names

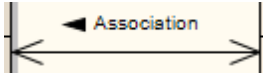
You can format and hide these labels using the **Labels** context menu.

**Access**    **Right-click on a label**

#### Options - For element labels

Option	Usage	Shortcut	See also
<b>Set Label Color</b>	Displays the Color dialog, from which you select a standard or custom color in which to display the label text.		
<b>Hide Label</b>	Hides the label on the diagram; to unhide the label, right-click on the element and select the <b>Appearance   Show labels</b> context menu option.		<a href="#">Changing Element Appearance</a> [948]
<b>Bold</b>	Displays the label text in bold.		
<b>Label Rotation</b>	(For embedded elements.) Orients the label in the horizontal (default) or vertical planes, with options to rotate the text in the vertical plane either clockwise or anti-clockwise.		
<b>Text Alignment</b>	Aligns the text to the left, center or right of the label text area. This is more apparent where there are several items of text in the label, of different lengths.		
<b>Default Position</b>	Moves the label to the position it was initially created in, relative to the parent element.		
<b>Default Color</b>	Sets the label color to the default color in which it was first created.		

#### Options - For connector labels

Option	Usage	Shortcut	See also
<b>Edit Label</b>	Displays the text within a small editing box, with the cursor within the text. Delete or add text as appropriate.	<b>F2</b>	
<b>Set Label Color</b>	Displays the Color dialog, from which you select a standard or custom color in which to display the label text.		
<b>Hide Label</b>	Hides the label on the diagram.  To unhide the label, right-click on the connector and select the <b>Visibility   Set Label Visibility</b> <sup>[1122]</sup> context menu option. Select the checkbox against each label to redisplay.		<a href="#">Hide/Show Labels</a> <sup>[1122]</sup>
<b>Bold</b>	Displays the label text in bold.		
<b>Label Rotation</b>	Orients the label in the horizontal (default) or vertical planes, with options to rotate the text in the vertical plane either clockwise or anti-clockwise.		
<b>Text Alignment</b>	Aligns the text to the left, center or right of the label text area. This is more apparent where there are several items of text in the label, of different lengths.		
<b>Direction</b>	Sets or clears a small arrow at the end of the label. You can choose to set the arrow pointing to the label source or destination.    This is part of the label, so if there is no label text (or if the label consists only of spaces) there is no direction indicator.		
<b>Default Position</b>	Moves the label to the position it was initially created in, relative to the parent connector.		
<b>Default Color</b>	Sets the label color to the default color in which it was first created.		



### Notes

- As labels can be concentrated on and around the element or connector, make sure that you click on a section of the required label that is clear of any other label or structure
- The Label Rotation option is not currently available for stereotyped elements and connectors

Learn more

- [Using the Image Manager](#)<sup>[860]</sup>
- [Run-time State](#)<sup>[1380]</sup>
- [Connector Properties](#)<sup>[1126]</sup>

**5.5.6.27 Pan and Zoom a Diagram**Topics

Topic	Detail	See also
<b>Pan</b>	<p>Pan across the Diagram View in the following ways:</p> <ul style="list-style-type: none"> <li>• Use ( ← ), ( → ), ( ↑ ), ( ↓ ), ( <b>Page Up</b> ), ( <b>Page Down</b> ), ( <b>Home</b> ) and ( <b>End</b> ) when the Diagram View is selected</li> <li>• Use the scrollbars</li> <li>• Use the middle mouse button</li> <li>• Use the Pan &amp; Zoom window</li> </ul>	<a href="#">Pan &amp; Zoom Window</a> <sup>[698]</sup>
<b>Zoom</b>	<p>You can zoom into and out from a diagram using the zoom buttons on the diagram toolbar, or by using the <b>Diagram   Zoom</b> submenu.</p>  <p>Change the zoom level by 10% by clicking on either the <b>Zoom In (+)</b> or <b>Zoom Out (-)</b> buttons.</p> <p>Alternatively, select the <b>Zoom In</b> or <b>Zoom Out</b> options from the <b>Diagram   Zoom</b> submenu.</p> <p>There are three ways to return the diagram to 100%:</p> <ul style="list-style-type: none"> <li>• Click on the  button</li> <li>• Select <b>Zoom to 100%</b> from the <b>Diagram   Zoom</b> submenu</li> <li>• ( <b>Ctrl</b> ) + middle-click the mouse</li> </ul>	

Notes

- You can zoom in and out of the main window dynamically by holding ( **Ctrl** ) and rolling the mouse wheel
- Changes in diagram magnification through the zoom options can be saved as permanent changes to the diagram
- At high levels of zoom, element features cease to display. This is because of the difficulty the Windows font mapper has in choosing a font for extreme conditions, and the result can look odd

### 5.5.6.28 Move Elements In Diagram Sections

As you build up a diagram, you might find that you have to move part of the diagram up, down or to one side. You can do this in one of two ways:


#### Topics

Topic	Detail	See also
<b>Moving Elements</b>	<ul style="list-style-type: none"> <li>Hold the left mouse button down and drag over a group of elements to move (creating an outline around the elements), then click on an element in the outline and move the group as required</li> <li>Press ( <b>Alt</b> ) and click on the diagram, then drag the cursor to move everything beyond the cursor in the direction of the movement.</li> </ul> <p>The first method enables you to reposition groups of elements within the larger diagram. The second method enables you to create space within the diagram without pushing some elements into others, as might happen if you cannot see the whole diagram on one screen.</p> <p>When you press ( <b>Alt</b> ) and click on the diagram, as you move the cursor a line displays on the diagram just behind the cursor. If you are moving the cursor left, everything to the left of the line moves with the cursor. If you move the cursor up, everything above the line moves up.</p> <p>However, if you move the cursor diagonally, two lines display to create a quadrant, and everything within the quadrant moves. For example, if you move the cursor left and down, everything below and left of the cursor moves.</p>	
<b>Fine Movement</b>	To adjust (or 'nudge') the position of a single element or a selected group of elements, press ( <b>Shift</b> ) + ( <b>→</b> ), ( <b>←</b> ), ( <b>↑</b> ) or ( <b>↓</b> ).	

### 5.5.6.29 View Last and Next Diagram

Enterprise Architect enables you to step backwards and forwards through the currently-open diagrams, including the Start Page.

#### Topics

Topic	Detail	See also
<b>Usage</b>	 <p>To view the previous or next diagram use the <b>Previous</b> or <b>Next</b> buttons on the Diagram toolbar.</p> <p>Use the <b>Home</b> button to display the default project diagram (if one has been specified).</p>	<a href="#">Default Project Diagram</a> <sup>[844]</sup>

### 5.5.6.30 Set Up Diagram Page

You can change the size of the diagram area (or scrollable/printable area) using the Diagram Properties dialog.

#### How to

To set the page size

Step	Action	See also
1	Load a diagram.	
2	Double-click on the background to open the Diagram Properties dialog.	
3	Click on the Diagram tab and, in the Appearance panel ensure that the <b>Show Page Border</b> checkbox is selected.	
4	On the Page Setup panel, click on the <b>Advanced</b> button. The Print Advanced dialog displays.	
5	Click on the <b>Page Setup</b> button. The Page Setup dialog displays. As you adjust the settings on this dialog, the page icon at the top illustrates the effects of your changes.	
6	In the <b>Size</b> field, click on the drop-down arrow and select an appropriate page size.	
7	In the Orientation panel click on the radio button for the orientation of the page to print.	
8	In the <b>Margins</b> panel, type the required left, right, top and bottom page margins for the diagram, in inches.	
9	Click on the <b>OK</b> button on the Page Setup dialog, the Print Advanced dialog, and the Diagram Properties dialog, in turn.  The area within the page boundary lines on your diagram is expanded or reduced accordingly. When you print or print preview, the output is cropped to these boundary lines and the diagram divided between the necessary number of pages.	

#### Topics

Topic	Detail	See also
<b>Setting the Default Paper Size for New Diagrams</b>	<p>You can set the default paper size for new diagrams on the Diagram page of the Options dialog (select the <b>Tools   Options   Diagram</b> menu option).</p> <p>Once the paper size is set there, all new diagrams have that as the default size.</p>	<a href="#">Diagram Options</a> <sup>608</sup>

**Notes**

- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Update Diagrams** permission to change diagram page setup

**Learn more**

- [Permissions List](#) <sup>329</sup>

**5.5.6.31 Scale Image to Page Size**

When you print a diagram, the default setting is to scale the image to fit the size of the printer paper you have defined in the page set-up. The image is not scaled up to fill the page, but it is scaled down if it exceeds the current page boundary. The image retains its current proportions; that is, it is scaled down equally in the X and Y dimensions. For a large diagram, this can mean that the components of the diagram are small and hard to read.

Alternatively, you can print a multi-page image; that is:


- allow the diagram image to print on as many printer pages as it naturally occupies, (no scaling), or
- scale the diagram image to exactly fit a specified number of pages

In all three cases you also define the paper size and orientation.

**How to**

To turn off or customize image scaling options

Step	Action	See also
<b>1</b>	Select the diagram to scale.	
<b>2</b>	<p>Either:</p> <ul style="list-style-type: none"> <li>• Double-click on the diagram background, or</li> <li>• Right-click on the background and select the <b>Properties</b> context menu option</li> </ul> <p>The &lt;type&gt; Diagram: &lt;name&gt; dialog displays.</p>	

Step	Action	See also
3	Click on the Diagram tab and, in the Page Setup panel click on the <b>Advanced</b> button.  The Print Advanced dialog displays.	<a href="#">Set Up Diagram Page</a> 
4	Select from the following options as required: <ul style="list-style-type: none"> <li>• <b>None</b>: select to print on as many pages as the diagram image covers</li> <li>• <b>Scale to 1 page</b>: select to scale the diagram image to fit on the currently selected page</li> <li>• <b>Custom</b>: select to specify the width and height of the diagram images across a specified number of pages</li> <li>• <b>Page Setup</b>: click to select the page size and alignment</li> </ul>	

#### Notes

- Before printing, make sure you have selected the required page layout using the **Page Setup** button

#### Learn more

- [File Menu](#) 

### 5.5.6.32 Lock Diagram

You can lock a diagram against inadvertent changes, such as moving or sizing elements.

#### How to

To lock a diagram

Step	Action	See also
1	Open the diagram to lock.	
2	Right-click on the background to open the diagram context menu.	
3	Click on the <b>Lock Diagram</b> option to prevent further changes.	
4	Click on the <b>OK</b> button.  If a user selects an item on a locked diagram, the object border or outline displays in red.	



### Notes

- This does not apply in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions if security is enabled; in that case, see the *Lock Model Elements* topic


### Learn more

- [Lock Model Elements](#) 

## 5.5.6.33 Undo Last Action

When editing diagrams, Enterprise Architect supports multiple undo levels for moving, re-sizing and deleting elements, and for deleting connectors.

### Topics

Topic	Detail	See also
<b>Usage</b>	<p>There are three ways to undo the last action:</p> <ul style="list-style-type: none"><li>• Press ( <b>Ctrl+Z</b> )</li><li>• Select the <b>Edit   Undo</b> menu option</li><li>• Click on the <b>Undo</b> button in the Default Tools toolbar - </li></ul>	


### Notes

- Currently you cannot undo element additions or connector moves

## 5.5.6.34 Redo Last Action

When editing diagrams, Enterprise Architect supports multiple undo levels for moving, re-sizing and deleting elements, and for deleting connectors. If an Undo action is in error, you can redo the action to reverse the Undo.

### Topics

Topic	Detail	See also
<b>Usage</b>	<p>There are three ways to redo the last action:</p> <ul style="list-style-type: none"><li>• Press ( <b>Ctrl+Y</b> )</li><li>• Select the <b>Edit   Redo</b> menu option</li><li>• Click on the <b>Redo</b> button in the Default Tools toolbar - </li></ul>	


### 5.5.7 Layout Diagrams



As you develop a model diagram and add more elements and connectors to it, you can apply one of a number of formats to automatically lay out specific **areas** or **sets of elements** in the diagram in a structured arrangement. If the diagram is very simple, you could even use this facility to lay out the **entire** diagram.

Additionally, you can provide filters and parameters for two or three types of layout and then 'chain' the layouts so that they are executed in sequence to result in a more refined layout. You can select a system-defined chain, or one that you have created yourself.

**Access**    **Diagram | Diagram Layout or**  
**Right-click on main toolbar | Diagram Layout**

#### Lay out elements on a diagram

Step	Action	See also
1	<p>Select the elements to lay out on the currently-active diagram - hold <b>Shift</b> or <b>Control</b> while you click on each required element, or hold the mouse button down while you sweep over the area containing the required elements.</p> <p>If no elements are selected on the active diagram, then all elements on the diagram are laid out (except where otherwise documented).</p>	
2	<p>Click on the drop-down arrow on the top left field of the Layout Tools window, and select the required layout type:</p> <ul style="list-style-type: none"> <li>• Circle/Ellipse</li> <li>• Box</li> <li>• Per Page</li> <li>• Digraph</li> <li>• Spring</li> <li>• Neaten</li> <li>• Converge/Diverge</li> <li>• Fan Relations</li> <li>• Auto Route</li> </ul> <p>The layout type determines the fields presented in the window, therefore the appropriate fields are described separately for each layout type.</p> <p>If you want to apply a sequence or <b>chain</b> of diagram layouts to a diagram, select either:</p> <ul style="list-style-type: none"> <li>• The appropriate system-defined chain name, or</li> <li>• The name of a chain you have created yourself</li> </ul> <p>You can create and modify a 'Layout Chain' by clicking on the  <b>Edit Layout Chains</b> button in the dialog toolbar.</p>	<p><a href="#">Circle/Ellipse</a> [875]</p> <p><a href="#">Box</a> [879]</p> <p><a href="#">Per Page</a> [881]</p> <p><a href="#">Digraph</a> [883]</p> <p><a href="#">Spring</a> [883]</p> <p><a href="#">Neaten</a> [885]</p> <p><a href="#">Converge/Diverge</a> [886]</p> <p><a href="#">Fan Relations</a> [888]</p> <p><a href="#">Auto Route</a> [890]</p> <p><a href="#">Chain diagram layouts</a> [894]</p>

Step	Action	See also
3	When you have completed the fields, click on the  button.  The selected elements are reorganized into the pattern format you have selected and according to the options you have chosen.	
4	If you do not want to work with the new layout, click on the <b>Undo</b> button in the toolbar (  ).	

### Notes

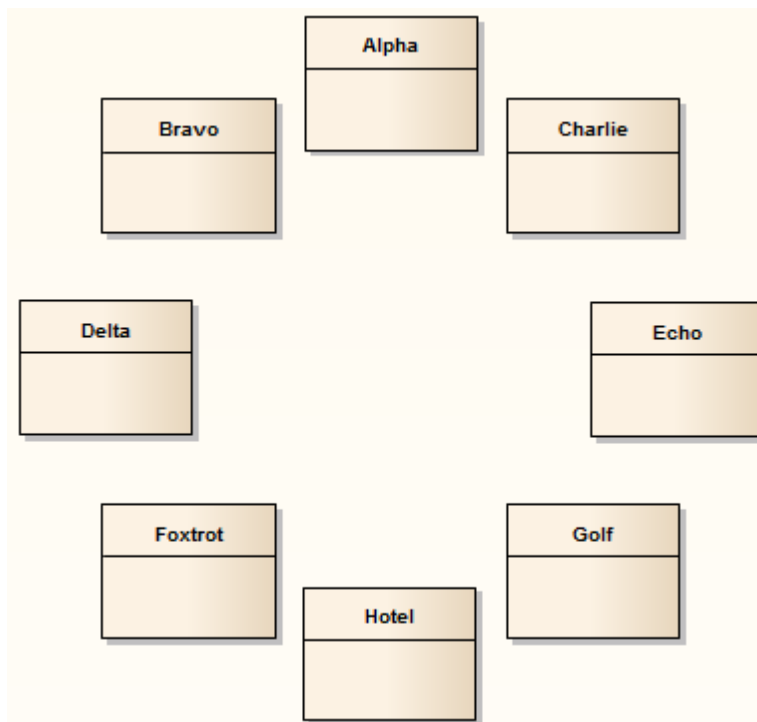
- You can also automatically lay out an entire diagram, without specifying any format or filters, using the alternative *Layout Diagram* facility

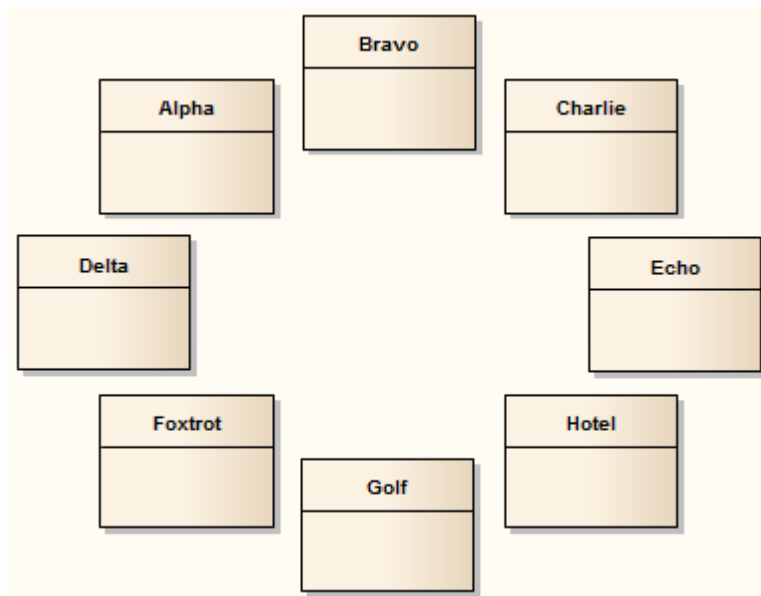
### Learn more

- [Lay Out a Diagram Automatically](#)<sup>[89]</sup>

#### 5.5.7.1 Circular/Elliptical Layout

The **Circle** and **Ellipse** layouts arrange the selected elements in a circle or elliptical pattern, using the largest horizontal and vertical element edge in the set of elements when calculating the radius of the layout arc.





#### Invoke the Circular or Ellipse layout

Step	Action	See also
1	Follow the general <i>Layout Diagrams</i> procedure, and at step 2 select either <b>Circle</b> or <b>Ellipse</b> as required.	<a href="#">Layout Diagrams</a> <sup>874</sup>
2	<p>Click on the drop-down arrow in the <b>Sort By:</b> field and select the required sort parameter:</p> <ul style="list-style-type: none"> <li>• <b>None</b> - elements are passed to the specified layout in the order in which they appear on the original diagram (left to right, top to bottom)</li> <li>• <b>Area (Ascending)</b> - elements are passed to the specified layout in order of the screen space they occupy, smallest to largest</li> <li>• <b>Area (Descending)</b> - elements are passed to the specified layout in order of the screen space they occupy, largest to smallest</li> <li>• <b>Name (Ascending)</b> - elements are passed to the specified layout in alphanumeric order, based on the element name</li> <li>• <b>Name (Descending)</b> - elements are passed to the specified layout in reverse alphanumeric order, based on the element name</li> <li>• <b>Element Type</b> - elements are grouped by type (for example, Class, Use Case) and in alphanumeric order within the group by name</li> </ul>	
3	<p>Under the <b>Placement</b> option, select either:</p> <ul style="list-style-type: none"> <li>• <b>Top to Bottom</b> - the elements are positioned in the required order, zig-zagged across the perimeter of the circle or ellipse, as in Diagram A</li> <li>• <b>Circular</b> - the elements are placed in the required order, clockwise around the perimeter of the circle or ellipse, as in Diagram B</li> </ul>	

Step	Action	See also
4	Select the <b>Center focused element</b> checkbox to put the last-selected element (the one with the hashed border) in the <b>center</b> of the circle or ellipse, as in Diagram C.	

Diagram A - Top To Bottom Layout

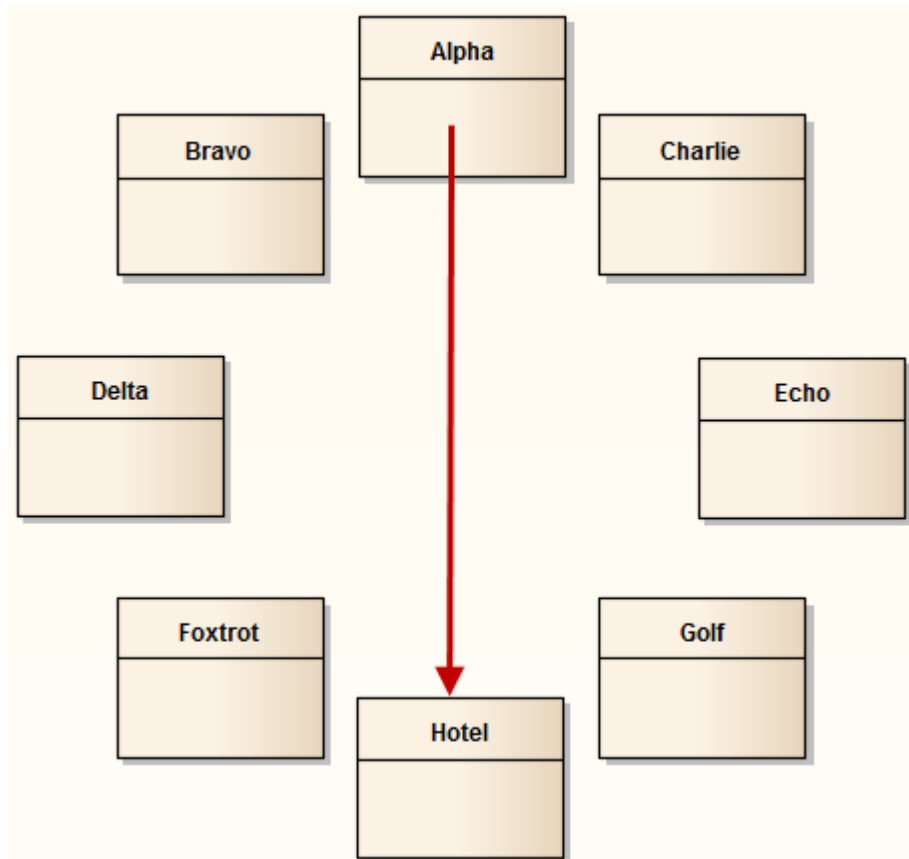


Diagram B - Circular Layout

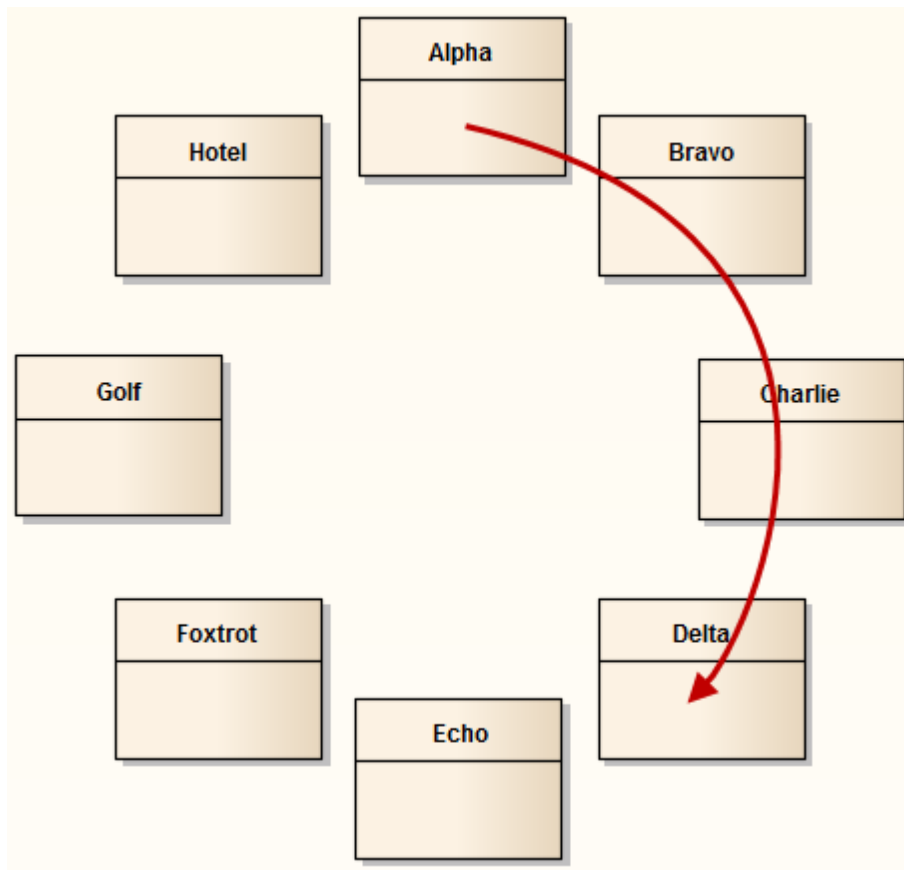
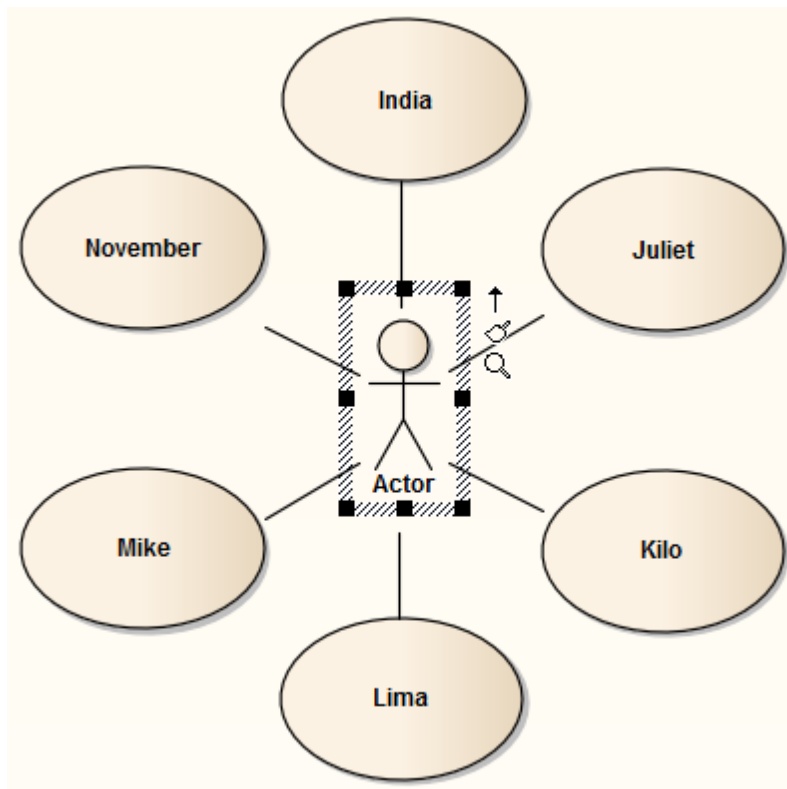


Diagram C - Center Focused Element



### 5.5.7.2 Box Layout

The **Box** layout arranges the set of selected elements into a square grid.



The dimensions of the box are determined by the number of columns that either:

- You set manually or
- The system automatically sets to a value that results in the same number of columns and rows (for example, 5, 6, 7, 8 or 9 elements would all be arranged in a three-column rectangle).

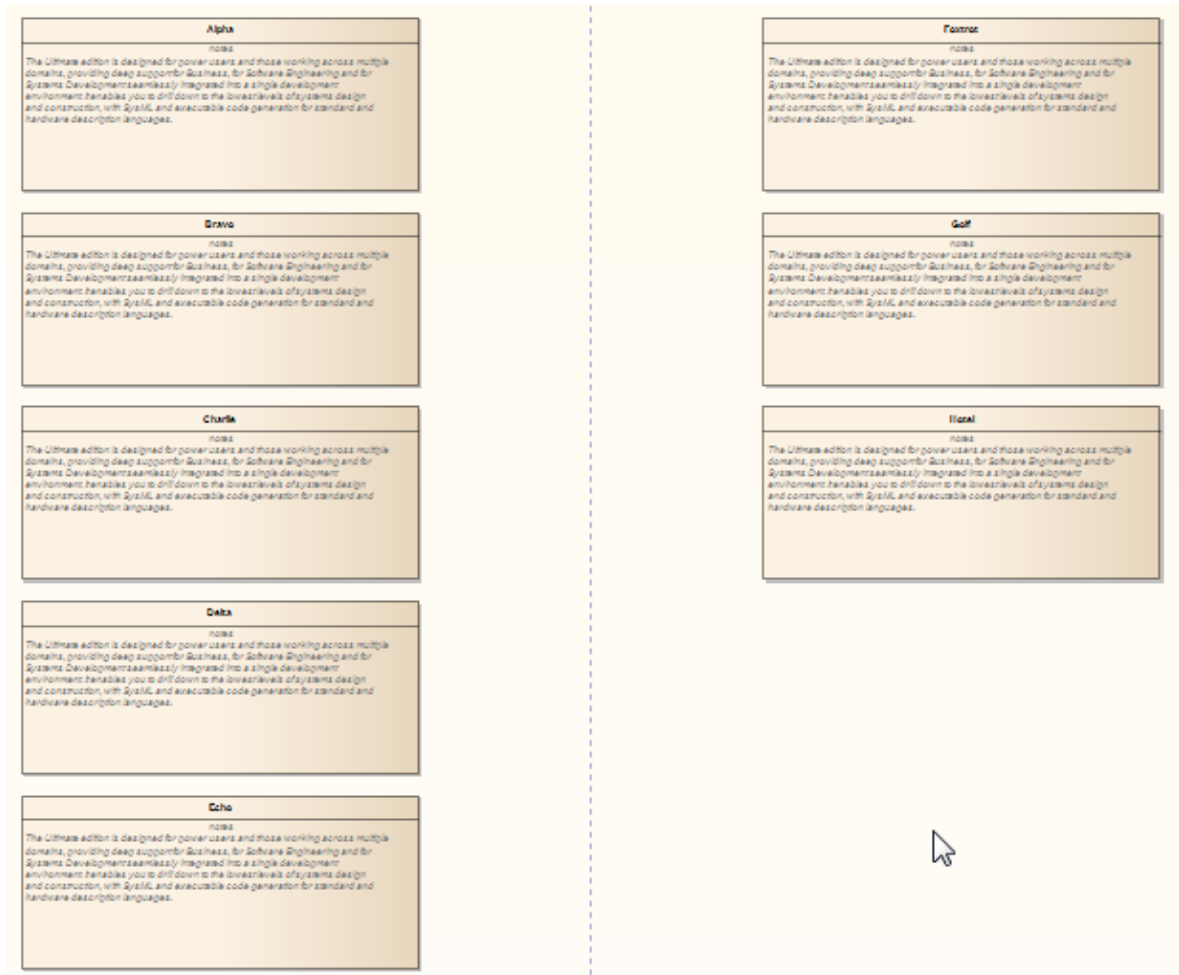
**Invoke the Box layout**

Step	Action	See also
1	Follow the general <i>Layout Diagrams</i> procedure, and at step 2 select <b>Box</b> .	<a href="#">Layout Diagrams</a> <sup>[874]</sup>
2	Click on the drop-down arrow in the <b>Sort By:</b> field and select the required sort parameter: <ul style="list-style-type: none"> <li>• <b>None</b> - elements are passed to the specified layout in the order in which they appear on the original diagram (left to right, top to bottom)</li> <li>• <b>Area (Ascending)</b> - elements are passed to the specified layout in order of the screen space they occupy, smallest to largest</li> <li>• <b>Area (Descending)</b> - elements are passed to the specified layout in order of the screen space they occupy, largest to smallest</li> <li>• <b>Name (Ascending)</b> - elements are passed to the specified layout in alphanumeric order, based on the element name</li> <li>• <b>Name (Descending)</b> - elements are passed to the specified layout in reverse alphanumeric order, based on the element name</li> <li>• <b>Element Type</b> - elements are grouped by type (for example, Class, Use Case) and in alphanumeric order within the group by name</li> </ul>	
3	In the <b>Padding (px)</b> field, type the vertical and horizontal distance between elements, in pixels.	
4	Select the appropriate element distribution option: <ul style="list-style-type: none"> <li>• <b>Automatically distribute:</b> Automatically calculate the dimensions of the box (the square root of the number of selected elements; for example, 16 elements create a 4x4 box)</li> <li>• <b>Specify distribution:</b> Manually define the width of the box, in columns</li> </ul>	
5	If you selected <b>Specify Distribution</b> , in the <b>Columns</b> field type the required number of columns.	



### 5.5.7.3 Per Page Layout

The **Per Page** layout divides each diagram page into a number of cells, which house the selected elements. The number of cells per page is determined by the page distribution parameter.



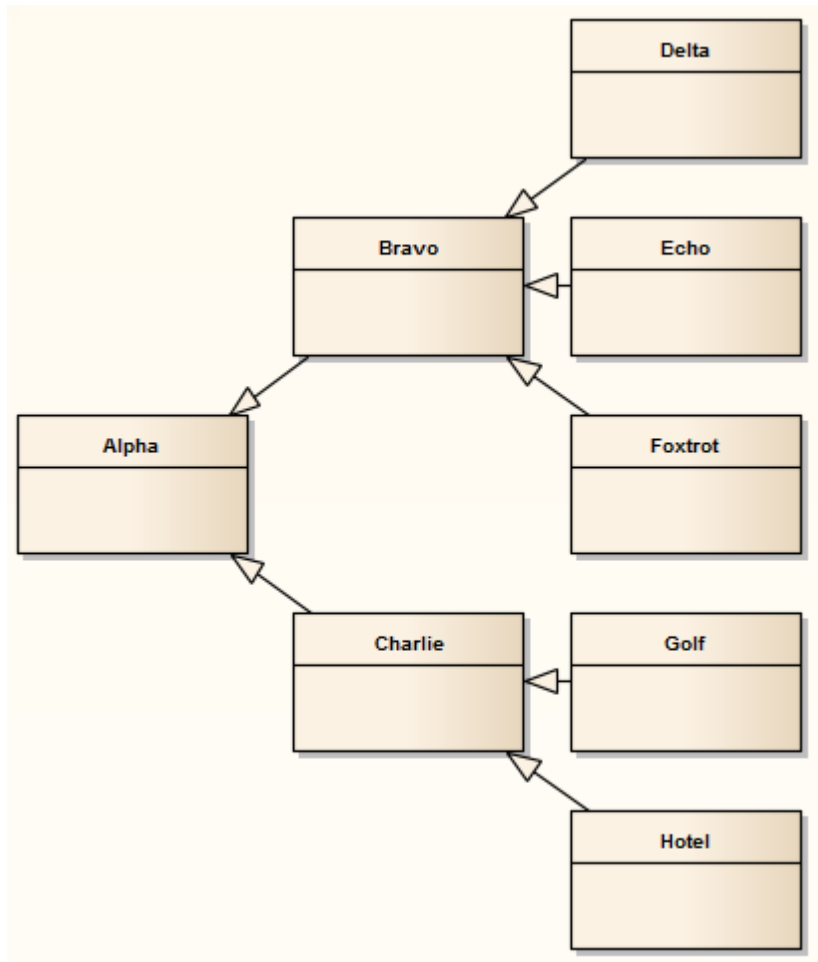
#### Invoke the Per Page layout

Step	Action	See also
1	Follow the general <i>Layout Diagrams</i> procedure, and at step 2 select <b>Per Page</b> .	<a href="#">Layout Diagrams</a> [874]
2	<p>Click on the drop-down arrow in the <b>Sort By:</b> field and select the required sort parameter:</p> <ul style="list-style-type: none"> <li>• <b>None</b> - elements are passed to the specified layout in the order in which they appear on the original diagram (left to right, top to bottom)</li> <li>• <b>Area (Ascending)</b> - elements are passed to the specified layout in order of the screen space they occupy, smallest to largest</li> <li>• <b>Area (Descending)</b> - elements are passed to the specified layout in order of the screen space they occupy, largest to smallest</li> </ul>	

Step	Action	See also
	<ul style="list-style-type: none"> <li>• <b>Name (Ascending)</b> - elements are passed to the specified layout in alphanumeric order, based on the element name</li> <li>• <b>Name (Descending)</b> - elements are passed to the specified layout in reverse alphanumeric order, based on the element name</li> <li>• <b>Element Type</b> - elements are grouped by type (for example, Class, Use Case) and in alphanumeric order within the group by name</li> </ul>	
3	In the <b>Padding (px)</b> field, type the vertical and horizontal distance between cells, in pixels.	
4	Select the appropriate page distribution option: <ul style="list-style-type: none"> <li>• <b>Automatically distribute</b>: automatically calculate the optimum number of cells, taking into consideration the largest horizontal and vertical element edges</li> <li>• <b>Specify distribution</b>: manually enter the per page grid dimensions</li> </ul>	
5	If you selected <b>Specify Distribution</b> , in the <b>Rows</b> and <b>Columns</b> fields type the required number of rows and columns.	
6	Select the <b>Center Elements</b> checkbox to place each element in the center of its cell; otherwise the element placement defaults to the top left corner of the cell.	
7	In the <b>Start Page</b> field, type the number from which to start page numbering; pages begin at the top left and continue horizontally to the right.	

#### 5.5.7.4 Digraph Layout

The **Digraph** layout arranges the selected elements into a directed graph ('digraph' for short). The Digraph attempts to highlight the hierarchy of the elements while keeping the direction of all connectors pointing to the same edge of the diagram.



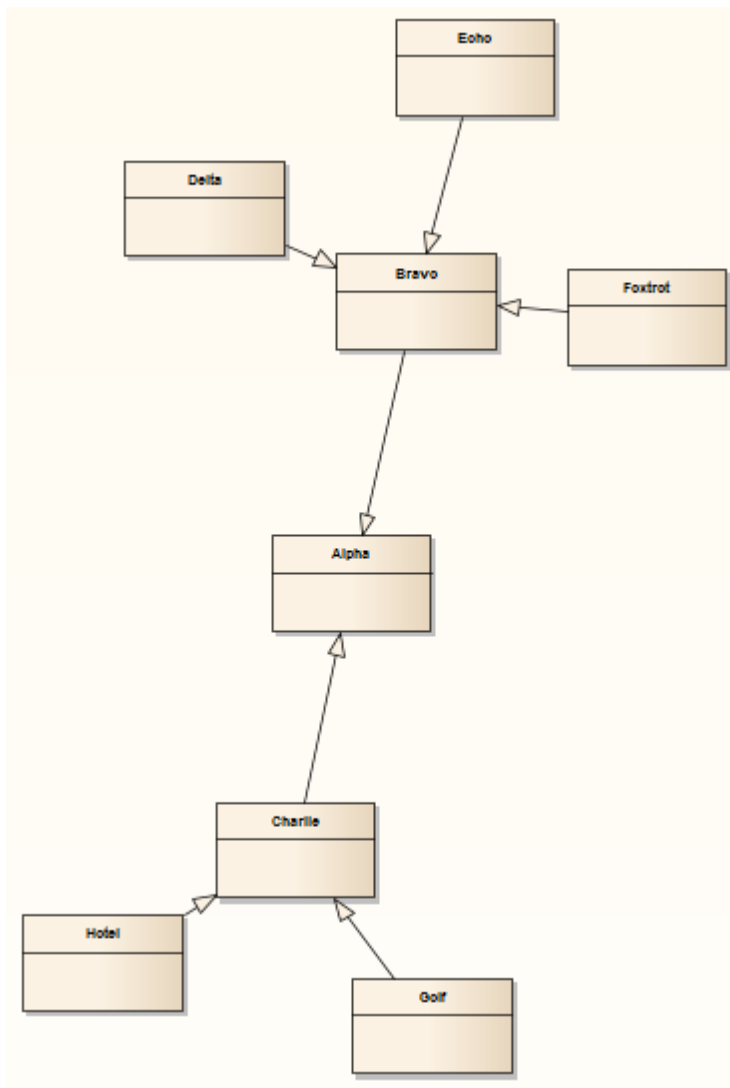
The Digraph layout provides the same behaviour as the Automatic Diagram layout.

##### Learn more

- [Layout Diagrams](#)<sup>[874]</sup>
- [Layout a Diagram Automatically](#)<sup>[891]</sup>

#### 5.5.7.5 Spring Layout

The **Spring** layout uses a force-directed approach to arrange the selected elements organically. It employs a physical analogy to lay out elements; each element is treated as a particle with a like electrical charge that repels other elements. Connectors act as springs (hence the name Spring layout) that draw connected elements back together. The layout is good for highlighting clusters of related objects and identifying symmetry in the graph.



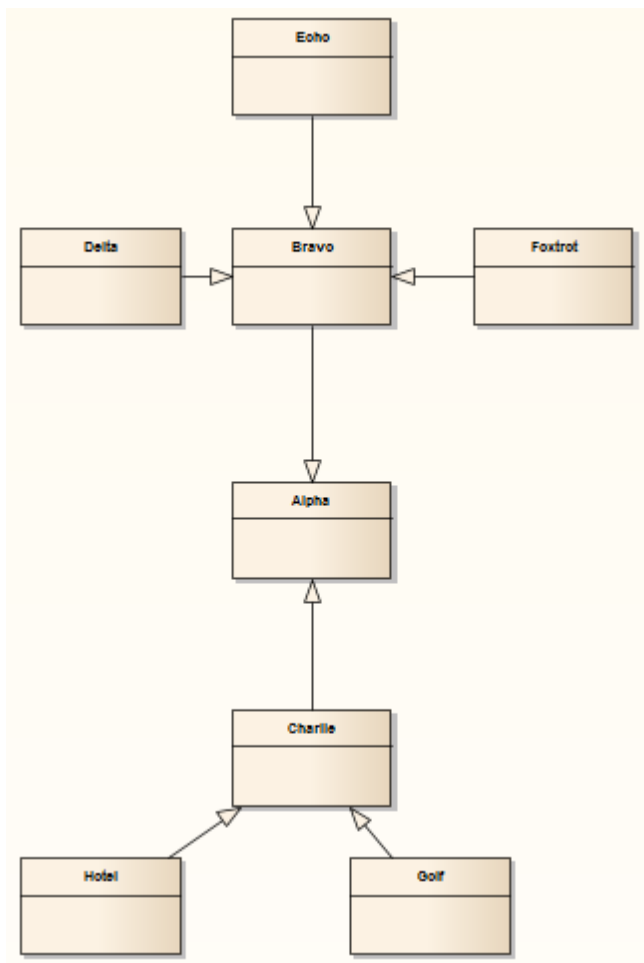
### Invoke the Spring layout

Step	Action	See also
1	Follow the general <i>Layout Diagrams</i> procedure, and at step 2 select <b>Spring</b> .	<a href="#">Layout Diagrams</a> <sup>[874]</sup>
2	In the <b>Iterations</b> field, type the number of iterations, or rounds, to perform to reach the final layout.  The layout is developed over several iterations; depending on the complexity of the graph, increasing the number of iterations produces a better result but takes longer to calculate.	
3	If the diagram contains elements that significantly vary in size, and that might overlap in the final layout, select the <b>Scale to prevent overlap</b> checkbox to scale up	

Step	Action	See also
	the positions of the selected elements (preserving size) until no elements overlap.	

### 5.5.7.6 Neaten Layout

The **Neaten** layout attempts to pull the selected elements into a grid based on their horizontal and vertical proximity to each other. That is, the layout pulls elements that are roughly aligned into a neater arrangement. Elements that share the same row or column are aligned based on the **Column** and **Row Snap** parameters. The arrangement shown for the **Spring** layout, when the Neaten layout is applied, might look like this:



#### Invoke the Neaten layout

Step	Action	See also
1	Follow the general <i>Layout Diagrams</i> procedure, and at step 2 select <b>Neaten</b> .	<a href="#">Layout Diagrams</a> <sup>874</sup>

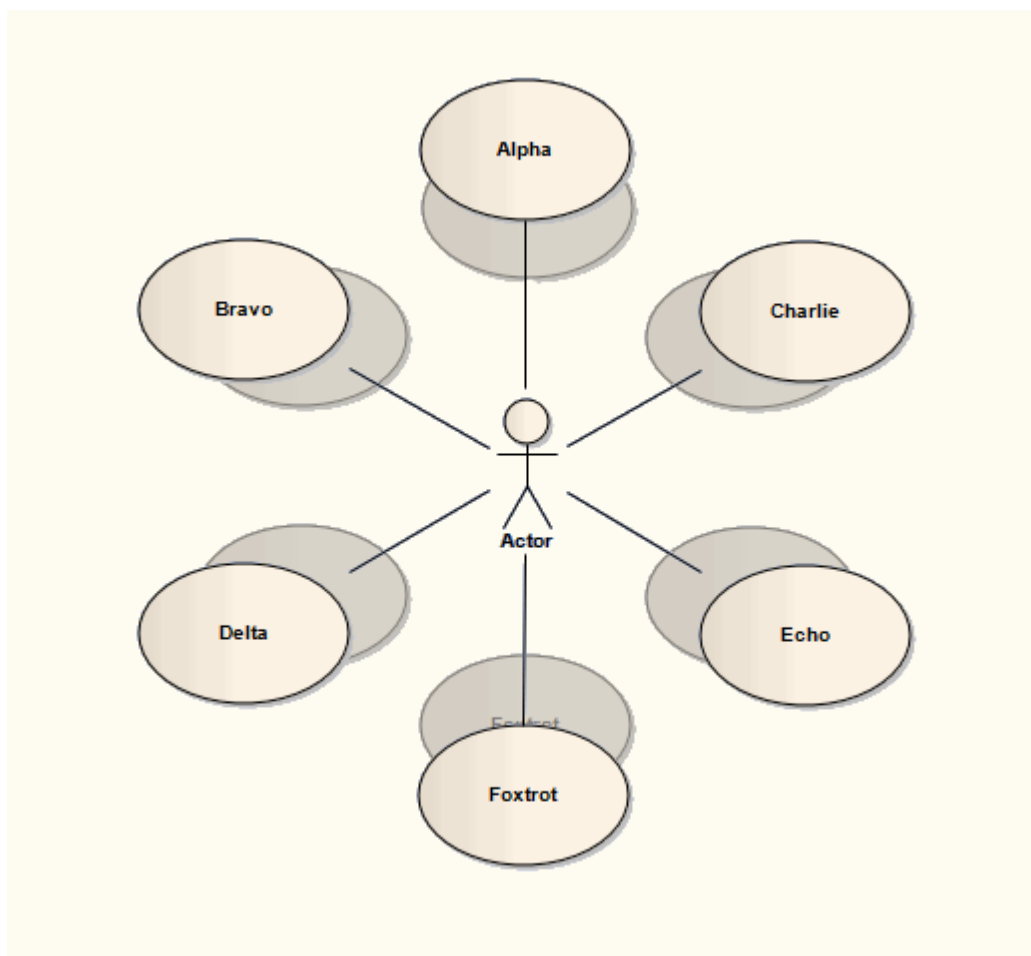
Step	Action	See also
2	<p>In the <b>Threshold (px)</b> field type the separation, in pixels, up to which elements are considered to be in the same row or column.</p> <p>A higher threshold value produces a tighter result, pulling elements that are far distant into the same row or column. A lower threshold value provides for a larger number of rows and columns</p>	
3	<p>In the <b>Column Snap</b> field, click on the drop-down arrow and select the appropriate alignment for elements in the same column:</p> <ul style="list-style-type: none"> <li>• <b>Left</b> - elements are aligned with the left edge of the left-most element in the column</li> <li>• <b>Center</b> - elements are aligned with the vertical center of the center-most element in the column</li> <li>• <b>Right</b> - elements are aligned with the right edge of the right-most element in the column</li> </ul>	
4	<p>In the <b>Row Snap</b> field, click on the drop-down arrow and select the appropriate alignment for elements in the same row:</p> <ul style="list-style-type: none"> <li>• <b>Top</b> - elements are aligned with the top edge of the highest element in the row</li> <li>• <b>Center</b> - elements are aligned with the horizontal center of the center-most element in the row</li> <li>• <b>Bottom</b> - elements are aligned with the bottom edge of the lowest element in the column</li> </ul>	

#### Learn more

- [Spring Layout](#)<sup>883</sup>

### 5.5.7.7 Converge/Diverge Layout

The **Converge** layout attracts the set of selected elements towards the center of their bounding rectangle. Conversely, the **Diverge** layout repels the set of selected elements away from the center of their bounding rectangle. The Converge/Diverge layout also tries to maintain connector angles if an element in the set contains a connector with waypoints.



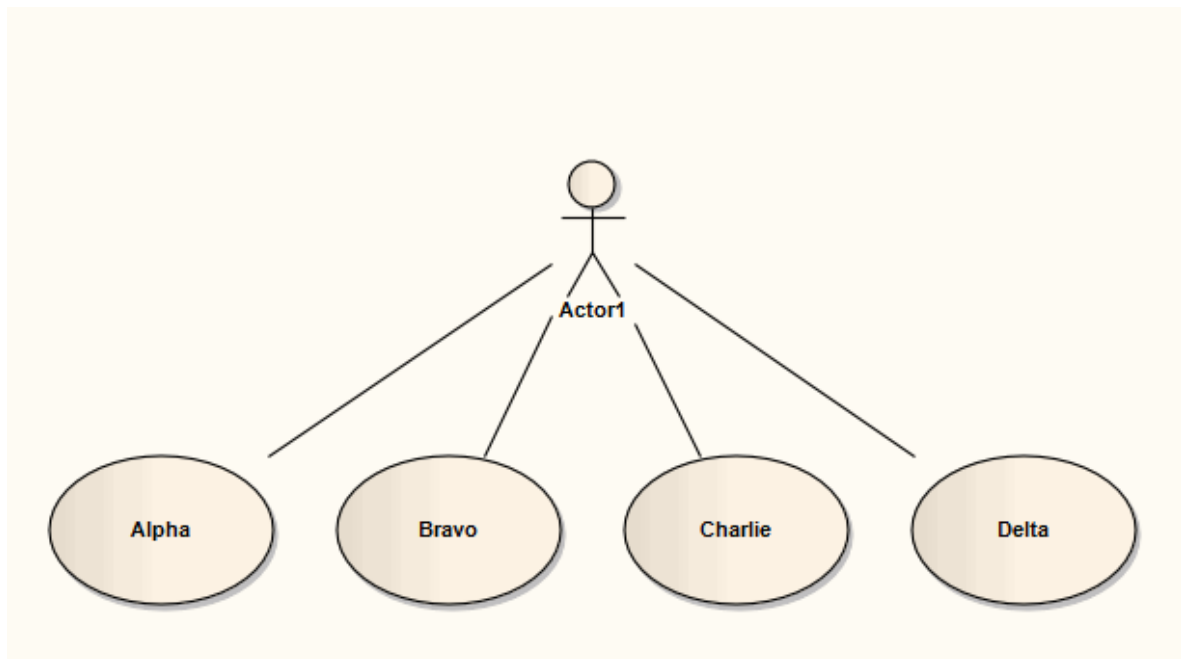
#### Invoke the Converge/Diverge layout

Step	Action	See also
1	Follow the general <i>Layout Diagrams</i> procedure, and at step 2 select <b>Converge/Diverge</b> .	<a href="#">Layout Diagrams</a> [874]
2	Select the radio button for the required direction of the layout: <ul style="list-style-type: none"> <li>• <b>Converge</b> - attracts the set of selected elements to the center point</li> <li>• <b>Diverge</b> - repels the set of selected elements from the center point</li> </ul>	
3	Set the <b>Amount (%)</b> slider to the required percentage. A smaller percentage provides finer control of the change in layout.  The slider determines how far the elements are moved towards or away from the center point; the movement is the element's current distance from the center point multiplied by the percentage value set on the slider.  In the <b>Converge</b> layout, the element moves <b>towards</b> the center point; in the <b>Diverge</b> layout the element moves further <b>away</b> from the center point.	

Step	Action	See also

### 5.5.7.8 Fan Relations Layout

The **Fan Relations** layout arranges the immediate relations of an element around a specified edge. This layout requires a single element to be selected on the diagram, to be used as the context for the layout.



#### Invoke the Fan Relations layout

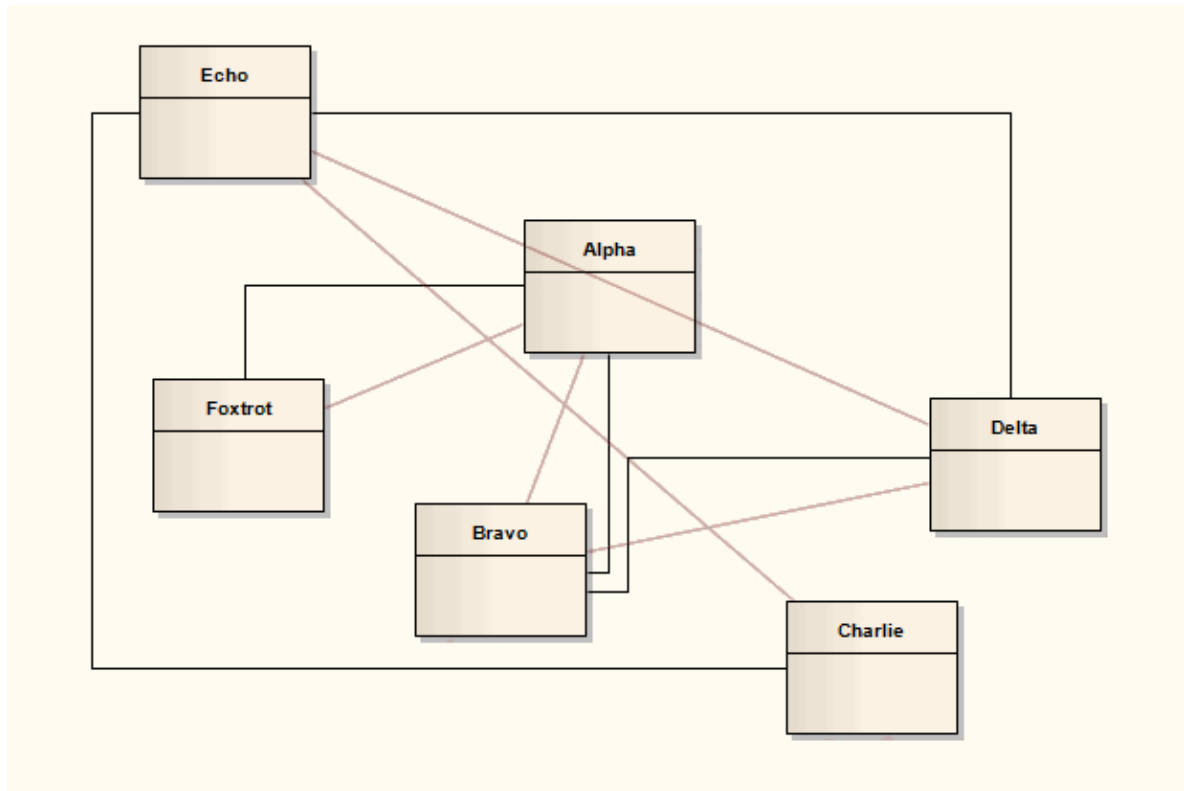
Step	Action	See also
1	Open the Layout Tools window; either: <ul style="list-style-type: none"> <li>Select the <b>Diagram   Diagram Layout</b> menu option, or</li> <li>Right-click on the main toolbar and select the <b>Layout Tools</b> context menu option</li> </ul>	<a href="#">Layout Diagrams</a> <sup>[874]</sup>
2	Select the single element around which to lay out related elements on the currently-active diagram.	
3	Click on the drop-down arrow on the top left button of the Layout Tools window, and select <b>Fan Relations</b> .	
4	Click on the drop-down arrow in the <b>Sort By</b> field and select the required sort parameter.	



Step	Action	See also
	<p>The options are:</p> <ul style="list-style-type: none"> <li>• <b>None</b> - elements are passed to the specified layout in the order in which they appear on the original diagram (left to right, top to bottom)</li> <li>• <b>Area (Ascending)</b> - elements are passed to the specified layout in order of the screen space they occupy, smallest to largest</li> <li>• <b>Area (Descending)</b> - elements are passed to the specified layout in order of the screen space they occupy, largest to smallest</li> <li>• <b>Name (Ascending)</b> - elements are passed to the specified layout in alphanumeric order, based on the element name</li> <li>• <b>Name (Descending)</b> - elements are passed to the specified layout in reverse alphanumeric order, based on the element name</li> <li>• <b>Element Type</b> - elements are grouped by type (for example, Class, Use Case) and in alphanumeric order within the group by name</li> </ul>	
5	In the <b>Padding (px)</b> field, type the separation required between the selected element and its related elements, in pixels.	
6	<p>Select the connector direction to use in determining the related elements to lay out; either:</p> <ul style="list-style-type: none"> <li>• <b>Incoming Nodes</b> - to lay out related elements that have the selected element as the target</li> <li>• <b>Outgoing Nodes</b> - to lay out related elements that have the selected element as the source</li> </ul>	
7	<p>In the <b>Fan Edge</b> field, click on the drop-down arrow and specify the edge of the selected element from which to lay out the related elements:</p> <ul style="list-style-type: none"> <li>• <b>Left</b> - to arrange related elements to the left of the selected element</li> <li>• <b>Right</b> - to arrange related elements to the right of the selected element</li> <li>• <b>Top</b> - to arrange related elements from the top of the selected element</li> <li>• <b>Bottom</b> - to arrange related elements from the bottom of the selected element</li> </ul>	

### 5.5.7.9 Auto Route Layout

The **Auto Route** layout orthogonally routes connectors between the selected elements. The layout attempts to find the shortest path between the two connected elements while minimizing crossings. In this illustration, the original connectors are shown in red.



#### Invoke the Autoroute layout

Step	Action	See also
1	Follow the general <i>Layout Diagrams</i> procedure, and at step 2 select <b>Auto Route</b> .	<a href="#">Layout Diagrams</a> [874]
2	In the <b>Cell Size (px)</b> field, type the effective connector separation in pixels.  When calculating connector routes, the algorithm divides the diagram into cells of a size determined by this field value; a smaller cell size results in connectors being placed closer together.	
3	In the <b>Element Margin</b> field, type the separation between connector segments and element borders, in pixels.	

### 5.5.7.10 Lay Out a Diagram Automatically

As you develop a diagram, it can become difficult to position the elements and connectors so that the layout and organization remain clear. Rather than continually moving structures around manually, you can simply select a **Layout Diagram** option to **automatically** reposition all the structures on the diagram in a logical, tree-based structure. If your diagram is complex, you can then do some manual 'tweaking' to draw out particular elements and relationships.

Generally the default layout parameters provide adequate layouts for a wide range of diagrams, but for a fine degree of control you can also set layout parameters using the Diagram Layout Options dialog.

**Access**    <open diagram> | **Diagram** | **Layout Diagram**  
               <open diagram> | **diagram toolbar: Auto Layout**  
               **Double-click background** > **Diagram: Set Layout Style**

#### Apply Layout Configurations

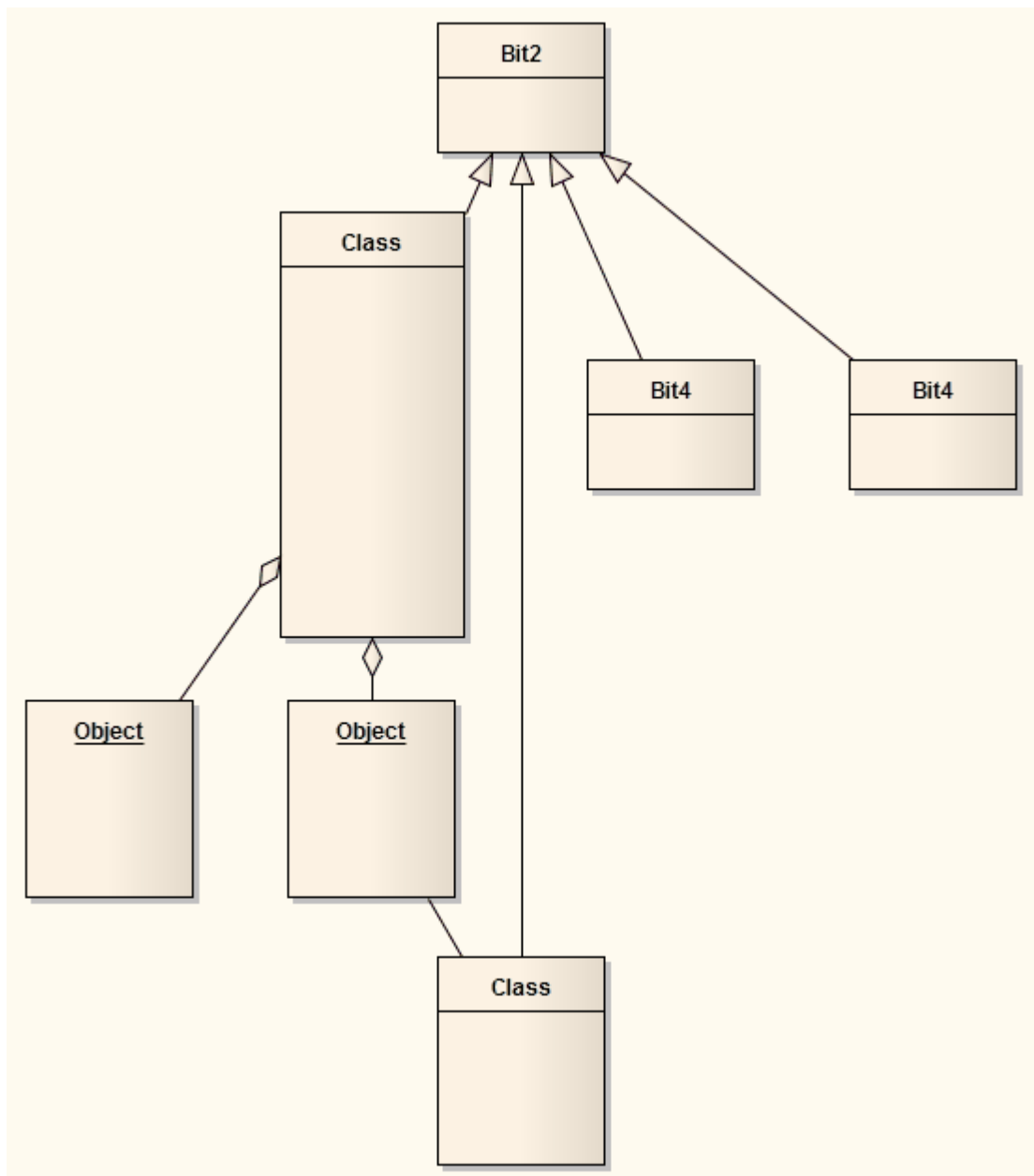
Options/Buttons	Action	See also
<b>Cycle Remove Options</b>	<p>These settings remove cycles in the element organization (where element <b>X</b> is the <b>source</b> of a path but also becomes the <b>target</b> of a branch of the path), by reversing the connectors that impose the cycling and then reorganizing the diagram and reinstating the reversed relationships; this identifies the primary source element in the diagram. Select</p> <ul style="list-style-type: none"> <li>• <b>Greedy</b> - to use the <b>Greedy Cycle Removal</b> algorithm, which minimizes the number of connectors reversed</li> <li>• <b>Depth First Search</b> - to use the <b>Depth First Search Cycle Removal</b> algorithm, which establishes the longest linear sequence possible before establishing parallel sequences and branches; this algorithm is less effective in large and/or complex diagrams, but produces a more natural layout than the Greedy algorithm</li> </ul>	
<b>Crossing Reduction Options</b>	<p>These options determine the length of time the routine searches for ways of reorganizing the layout to avoid crossed relationships:</p> <ul style="list-style-type: none"> <li>• <b>Iterations</b> - Type the number of iterations to be used during cycle removal (more than 8 does not usually provide any improvement)</li> <li>• <b>Aggressive</b> - Select this checkbox to use an aggressive (detailed and time-consuming) crossing reduction step</li> </ul>	
<b>Layering Options</b>	<p>These settings determine how elements are organized in layers during layout. Select:</p> <ul style="list-style-type: none"> <li>• <b>Longest Path Sink</b> - to use the <b>Longest Path Sink Layering</b> algorithm, where the final target elements (<b>sinks</b>, which have no relationships <b>issuing</b> from them) are arranged in a layer at the top of the diagram, and the relationship paths built downwards from there in as many layers as there are nodes in the longest path</li> <li>• <b>Longest Path Source</b> - to use the <b>Longest Path Source Layering</b> algorithm, where the original <b>source</b> elements (those with no relationships <b>entering</b> them) are arranged in a layer at the bottom of the diagram and the relationship paths built up from there in as many layers as there are nodes in the longest path</li> <li>• <b>Optimal Link Length</b> - to use the <b>Optimal Link Length Layering</b></li> </ul>	

Options/Buttons	Action	See also
	algorithm, which organizes the elements into the number of layers that minimizes the total source-to-sink relationship chain; in this layout you can have both source elements and sink elements at various levels of the diagram	
<b>Layout Options</b>	<p>These settings define how far apart the element layers and columns are in the layout.</p> <ul style="list-style-type: none"> <li>• <b>Layer Spacing</b> - Type the default number of logical units <b>between layers</b> of elements (vertical spacing)</li> <li>• <b>Column Spacing</b> - Type the default number of logical units between elements <b>within</b> a layer (horizontal spacing)</li> </ul>	
<b>Initialize Options</b>	<p>The autolayout routine inserts line waypoints and connectors into relationship paths to help plot the direction of the relationships; the routine then assigns an index number to every node, such that nodes in the same layer are numbered left to right. These settings determine how those index numbers are assigned. Select:</p> <ul style="list-style-type: none"> <li>• <b>Naive</b> - to use the <b>Naive Initialize Indices</b> algorithm, which assigns index numbers to nodes as they are encountered in a sweep and tends to place all waypoints to the right of real nodes (and therefore long relationships between a small number of elements to the right of chains of short relationships between several elements)</li> <li>• <b>Depth First Search Outward</b> - to use the <b>Depth First Out Initialize Indices</b> algorithm, which assigns index numbers to nodes as they are encountered in a depth first search from source nodes outwards (and would therefore place longer relationship chains to the left of shorter chains, with the primary <b>source</b> node at the <b>start</b> of the diagram flow)</li> <li>• <b>Depth First Search Inward</b> - to use the <b>Depth First In Initialize Indices</b> algorithm, which also assigns index numbers to nodes as they are encountered in a depth first search, but from sink nodes inwards (and would therefore place longer relationship chains to the left of shorter chains, with the ultimate <b>target</b> node at the <b>end</b> of the diagram flow)</li> </ul>	
<b>Direction</b>	Select the direction in which all directed connectors should point, to set the position of the primary source element and the overall flow of the diagram. That is, <b>Up</b> , <b>Down</b> , <b>Left</b> , or <b>Right</b>	
<b>Set as Project Default</b>	<p>Select this checkbox to apply the diagram layout settings to <b>all diagrams in the project</b>.</p> <p>If you later define different settings and select this checkbox, the new settings override all those saved previously.</p>	
<b>OK</b>	Click on this button to save your settings.	

### Example

This illustration shows an automatically laid out diagram with the following options set:

- Depth First Search
- Optimal Link Length
- Depth First Search Outward
- Direction - Up



#### Notes

- This facility is available for all types of diagram other than Timing and Interaction Overview diagrams
- If you decide that the autolayout is not appropriate, you can **reverse** it before saving the diagram; click **Ctrl+Z**

### Learn more

- [Layout Diagrams](#) <sup>[874]</sup> (manually)

#### 5.5.7.11 Chain Diagram Layouts

Each of the diagram layout formats has a specific effect on the layout and appearance of a diagram. These effects can be combined by chaining certain layouts in a sequence, to further improve the final appearance of the diagram. Enterprise Architect provides a selection of standard 'chains' for your use, and you can quickly and easily set up chains of your own.

The system does not restrict the way in which you combine the layouts, but in practice some layouts complement each other better than others, or are more logically executed in a certain order. You therefore need to consider the effect of each layout on the diagram - or part of the diagram - as you develop the chain. You also provide parameters for each layout, so you would set the parameters for each format to complement those for the other formats in the chain, using the Layout Chains dialog.

The available formats could be grouped by impact on the diagram:

- **Circle, Ellipse, Box, Per Page, Digraph** and **Spring** each adjust element positions completely, so would destroy the effects of any other layout if used after it in a chain; one of these would therefore be the first layout to apply in the chain
- **Fan Relations** operates on a specific element, so you might not want to further adjust the diagram layout; if you did, you would only use these next layouts
- **Converge/Diverge** could be used after some of the above layouts, to broaden or narrow the final layout
- **Neaten** is a logical successor to the previous types, adjusting positions to provide more uniform separation
- **Auto-route** is the best final layout, to adjust connectors to the clearest routes without changing element positions






In creating a chain of layouts, you might consider these suggestions in selecting which layouts to use and in which sequence.

**Access** **Diagram | Diagram Layout:**   
**Right-click on main toolbar | Diagram Layout**

### Set up a chain

On the Layout Chains dialog:

Step	Action	See also
1	<p>Either:</p> <ul style="list-style-type: none"> <li>• Click on the <b>New Chain</b> button and type in a new chain name (and click on the <b>OK</b> button) or</li> <li>• Click on the drop-down arrow and select the name of an existing chain to edit</li> </ul> <p>If you want to delete the existing chain you have selected, click on the <b>Delete Chain</b> button.</p>	

Step	Action	See also
	If you want to rename the existing chain, click on the  button and overwrite the existing name with the new name (and click on the <b>OK</b> button).	
2	In the <b>Existing Layouts</b> field, click on the first layout to execute in the chain, then click on the  button to add it to the chain. Repeat this step to add further layouts to the chain.	
3	If you do not want to include a layout that is currently in the chain, click on it and click on the  button. If you want to move a layout in the chain to a different point in the sequence, click on it and click on the  or  button as appropriate.	
4	If you want to change the name of a <b>layout</b> , click once on the name in the chain list and overwrite it.  To set the parameters of each layout, double-click on the layout name in the chain list. The appropriate parameter dialog for the layout displays; complete the fields as appropriate and click on the <b>OK</b> button.  These are the parameters that will be applied whenever you execute this layout chain, regardless of the settings in any other use of the layout.	<a href="#">Circular/Elliptical Layout</a> <sup>[87]</sup> <a href="#">Box Layout</a> <sup>[87]</sup> <a href="#">Per Page Layout</a> <sup>[88]</sup> <a href="#">Digraph Layout</a> <sup>[88]</sup> <a href="#">Spring Layout</a> <sup>[88]</sup> <a href="#">Neaten Layout</a> <sup>[88]</sup> <a href="#">Converge/Diverge Layout</a> <sup>[88]</sup> <a href="#">Fan Relations Layout</a> <sup>[88]</sup> <a href="#">Auto Route Layout</a> <sup>[89]</sup>
5	Click on the <b>Save Chains</b> button to save your changes, then click on the <b>OK</b> button. (If you have not saved your changes, an error message displays; click on the <b>Yes</b> button.)	

#### Learn more

- [Layout Diagrams](#)<sup>[87]</sup>

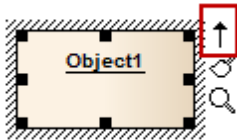
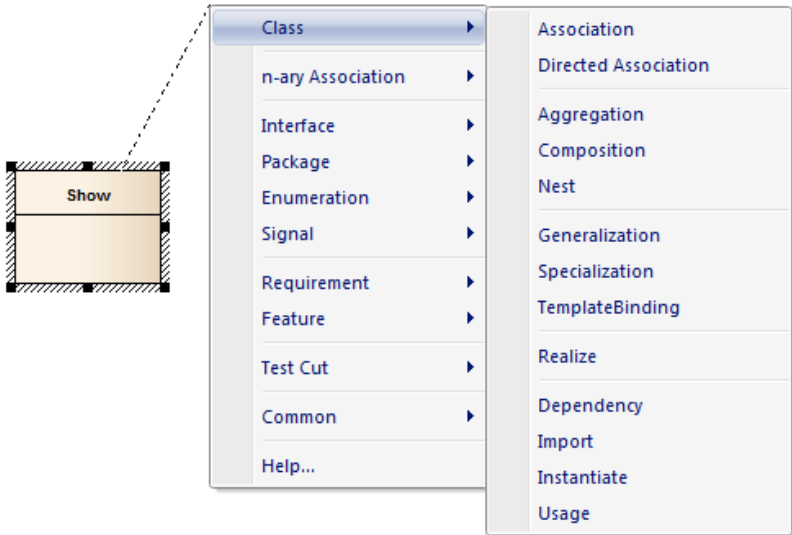
### 5.5.8 The Quick Linker

If you want to create new elements and connectors quickly on a diagram, use the **Quick Linker** on your initial element to guide you in creating a related element and connector of the appropriate types, without having to weigh up the full range of options from the Diagram Toolbox. The Quick Linker consists of paired menus that list the commonest objects appropriate to the context of the selected element. You simply select the type of element you need from the first menu, which displays the second menu offering a choice of appropriate types of connector. When you select the connector, you create a new element connected to your initial element, complete with any subordinate objects such as Object Nodes and Action Pins.

If the connectors and elements suggested by the Quick Linker do not include the types you need, you can create them from the Diagram Toolbox pages. Also, a Technology Developer can edit the lists of elements and connectors, and create new combinations.

There are two methods of accessing the Quick Linker menus.

#### Access Method

Action	Detail	See also
<b>Click on Quick Linker Arrow</b>	<p>When you click on an element in a diagram, the Quick Linker arrow is displayed at the upper right corner of the element:</p>  <p>Click on this arrow and drag it in the direction in which you want to create the new connector and element. When you release the mouse button, the element menu displays. When you select an element type, the connector menu displays.</p>  <p>When you select the connector type, the element and connector are</p>	



Action	Detail	See also
	added to the diagram in the location in which you displayed the menus.	
<b>Press ( Ctrl+Shift+ &lt;arrow key&gt; )</b>	<p>After selecting the element, press ( <b>Ctrl+Shift</b> ) and the keyboard arrow key for the direction in which you want to create the new connector and element (<b>Up, Down, Left</b> or <b>Right</b>).</p> <p>The element and connector menus display as above. When you select the type of connector, the new element and connector are created off the edge of the element appropriate to the arrow key you pressed. For example, if you press the <b>Left</b> arrow key, the new element is created to the left of the original element.</p>	

### Notes

- When using the **Ctrl+Shift+<arrow key>** method on a **Sequence** diagram, you cannot use the **Up** or **Down** keys; the **Left** and **Right** arrow keys create a connector to a Lifeline element
- When using the **Ctrl+Shift+<arrow key>** method on a **Timing** diagram, you cannot use the **Left** or **Right** keys

### Learn more

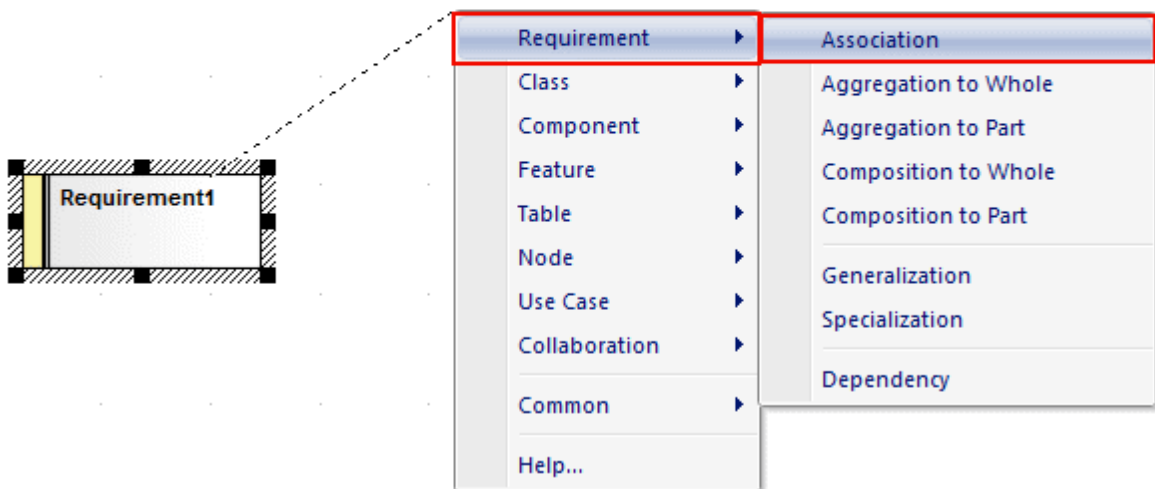
- [Create New Elements](#) <sup>897</sup>
- [Create Connectors](#) <sup>898</sup>

#### 5.5.8.1 Create New Elements



*Click, drag and release*

*Choose element and connector type*



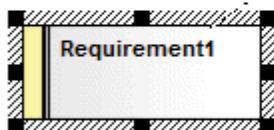
### Topics

Topic	Detail	See also
<b>Usage</b>	<ul style="list-style-type: none"> <li>Press and hold ( <b>Shift</b> ) while selecting the type of connector to select an existing classifier as the target</li> <li>For rapid modeling, you can suppress the Properties dialog when creating new elements; see the option <b>Tools   Options   Objects: Edit Object on New</b></li> </ul> <p>The available Quick Linker options depend on the type of element selected; for example, the Quick Linker options for a Class differ from those of an Actor.</p> <p>These are the most appropriate, commonly used elements and connectors for the source element; you can create other target elements and connectors by selecting them from the appropriate Toolbox page.</p>	

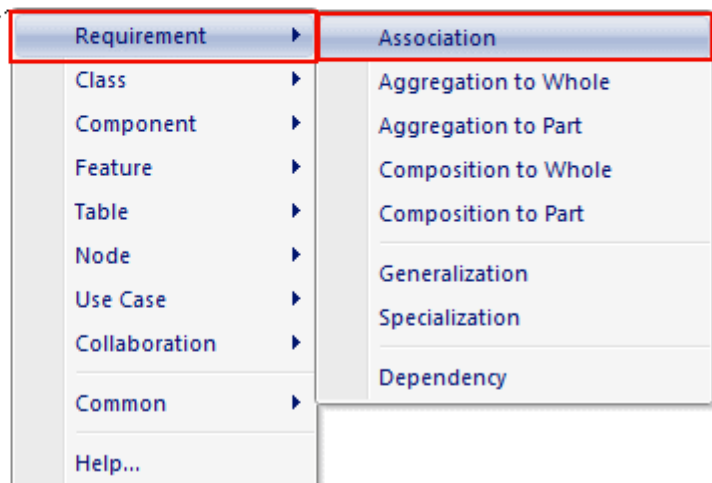
#### 5.5.8.2 Create Connectors



*Click, drag and release*



*Choose element and connector type*



### Topics

Topic	Detail	See also
<b>Usage</b>	<p>The list of connectors provides the most appropriate, commonly-used connectors for the source and target element types.</p> <p>If you want to use a different connector, select the appropriate Toolbox page, click on the required connector and then on the source element, and drag across to the target element.</p> <p>The connector does not actually establish until you release the mouse button over the target element; however, a dotted line shows where the connector would be at any point, and the solid outline of the nearest element or extension changes to a hatched outline as you move the cursor onto it - this helps you identify where the connector should connect to, if there are many closely-arranged elements, Parts, Ports and other extensions.</p> <p>You can also bend the connector, pressing ( <b>Shift</b> ) as you drag the cursor in a new direction.</p> <p>You can create a Notelink to attach a Note or Constraint to a connector, simply by dragging the Quick Linker arrow on the Note to the connector.</p>	

## 5.6 Elements



Models are constructed from elements, each of which has its own meaning, rules and notation. Elements can be used at different stages of the design process for different purposes.

The properties and features of an element can be defined and displayed through a set of windows and dialogs.

### Learn more

- [Element Property Displays](#) <sup>[956]</sup>
- [Element Tasks](#) <sup>[900]</sup>
- [Element Context Menu](#) <sup>[939]</sup>
- [Visual Representation](#) <sup>[953]</sup>
- [In-place Editing Options](#) <sup>[1032]</sup>

### 5.6.1 Element Tasks

This topic describes the following common tasks that you can perform on elements in Enterprise Architect:

Topic	Link
Create Elements	<a href="#">Create Elements</a> <sup>[902]</sup>
Add Elements Directly to Packages	<a href="#">Add Elements Directly to Packages</a> <sup>[903]</sup>
Use Auto Naming and Auto Counters	<a href="#">Set Auto Naming and Auto Counters</a> <sup>[904]</sup>
Set Element Parent	<a href="#">Set Element Parent</a> <sup>[908]</sup>
Show Element Use	<a href="#">Show Element Use</a> <sup>[910]</sup>
Set Up Cross References	<a href="#">Set Up Cross References</a> <sup>[916]</sup>
Move Elements Between Packages	<a href="#">Move Elements Between Packages</a> <sup>[913]</sup>
Move Elements Within Diagrams	<a href="#">Move Elements Within Diagrams</a> <sup>[911]</sup>
Copy structured or composite elements between diagrams	<a href="#">Copy Elements Between Diagrams</a> <sup>[912]</sup>

Topic	Link
Copy Elements Between Packages	<a href="#">Copy Elements Between Packages</a> <sup>[918]</sup>
Change Element Type	<a href="#">Change Element Type</a> <sup>[918]</sup>
Align Elements	<a href="#">Align Elements</a> <sup>[918]</sup>
Resize Elements	<a href="#">Resize Elements</a> <sup>[919]</sup>
Delete Elements	<a href="#">Delete Elements from Diagram and Model</a> <sup>[921]</sup>
Customize Visibility of Elements	<a href="#">Customize Visibility of Elements</a> <sup>[922]</sup>
Create Notes and Text	<a href="#">Create Notes and Text</a> <sup>[923]</sup>
Link Note to Internal Documentation	<a href="#">Link Note to Internal Documentation</a> <sup>[924]</sup>
Set an Element's Default Appearance	<a href="#">Set an Element's Default Appearance</a> <sup>[927]</sup>
Use Element Templates	<a href="#">Use Element Templates</a> <sup>[929]</sup>
Highlight Context Element	<a href="#">Highlight Context Element</a> <sup>[930]</sup>
Make Linked Element a Local Copy	<a href="#">Make Linked Element a Local Copy</a> <sup>[931]</sup>
Copy Features (Attributes and Operations) Between Elements	<a href="#">Copy Features (Attributes and Operations) Between Elements</a> <sup>[932]</sup>
Move Features Between Elements	<a href="#">Move Features Between Elements</a> <sup>[933]</sup>

### Notes

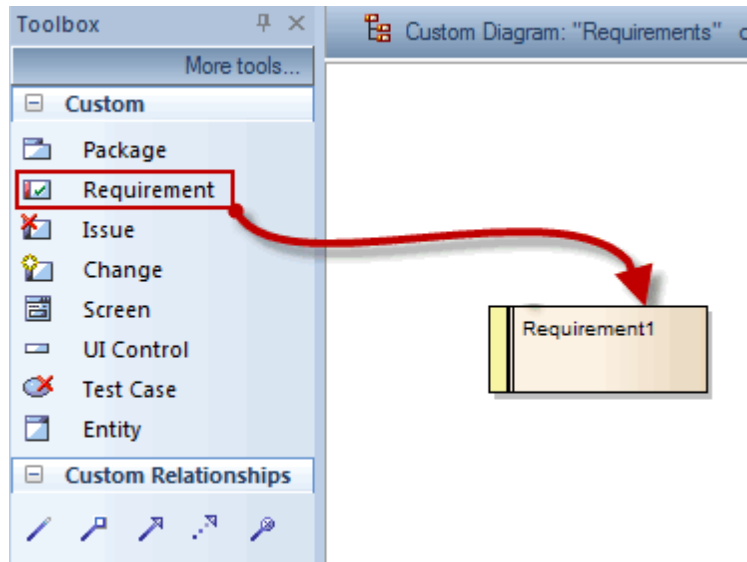
- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have Update Element permission to update element properties or delete an element

### Learn more

- [Permissions List](#)<sup>[329]</sup>

### 5.6.1.1 Create Elements

#### Example



#### Notes

- If you have defined a template element in the Templates Package, Enterprise Architect applies the characteristics of that template to each new element you create of that type
- If you are creating several elements of one type, after creating the first just press **(Shift+F3)** or **(Ctrl) + click** to create the next element of that type
- Once you have created elements, you can re-use them by dragging them from the Project Browser and dropping them onto your diagrams
- Sometimes it is useful to add elements to a Package, without a diagrammatic representation; this can be accomplished via the Project Browser and is explained in the topic - *Add Elements Directly to a Package*

#### Learn more


- [Create Elements In Place Using the Quick Linker](#)<sup>[897]</sup>
- [Create Elements Using the Toolbox](#)<sup>[792]</sup>
- [Create Elements Using the Diagram Context Menu](#)<sup>[778]</sup>
- [Create Elements From Text File](#)<sup>[1773]</sup>
- [Create a Group of Elements Using Design Patterns](#)<sup>[1464]</sup>
- [Using UML Profiles](#)<sup>[1472]</sup>
- [Pasting from the Tree](#)<sup>[833]</sup>
- [Add Elements Directly to a Package](#)<sup>[903]</sup>
- [Set Element Templates Package](#)<sup>[929]</sup>
- [Behavioral Diagram Elements](#)<sup>[1265]</sup>
- [Structural Diagram Elements](#)<sup>[1357]</sup>

### 5.6.1.2 Add Elements Directly To Packages

You can quickly add new elements to a Package without the necessity of adding a diagram element at the same time. This is particularly useful in defining a group of Requirements, Changes, Issues, base Classes or other element types that might not require diagrammatic representation in the model.

**Access** **Project Browser | Right-click on Package | Add Element** (Ctrl+M)

#### Add an element to a Package


Field/Option/ Button	Action	See also
<b>Active Toolset</b>	Shows the UML Diagram Type, profile, Add-In or MDG Technology within which you are creating the element.	
<b>Select Toolset</b>	<p>If the Active Toolset is not appropriate for the type of element you want to create, click on this button and select the required diagram type or technology.</p> <p>The <b>&lt;default&gt;</b> group in the list provides a basic set of elements drawn from across the UML Behavioral and Structural groups, and the Enterprise Architect Extended groups.</p> <p>The selected technology is displayed in the <b>Active Toolset</b> field.</p>	
<b>Name</b>	Type the name of the element.	
<b>Auto</b>	<p>As an alternative to typing in the name, click on this button to insert predefined auto-counter text.</p> <p>If you already have some text in the <b>Name</b> field, it is over-written by the auto-counter text.</p>	<a href="#">Set Auto Naming and Auto Counters</a> 
<b>Type</b>	<p>Click on the drop-down arrow and select the element type from the list.</p> <p>The list of elements is derived from the diagram or technology shown in the <b>Active Toolset</b> field.</p>	
<b>Stereotype</b>	If required, either type the stereotype name or click on the drop-down arrow and select the stereotype.	
<b>Open Properties Dialog on Creation</b>	Select the checkbox if the Properties dialog is to open immediately after the element is created.	
<b>Close Dialog on OK</b>	<i>Deselect</i> the checkbox to add multiple elements in one session.	
<b>Add to current</b>	Available if you have a diagram open.	

Field/Option/ Button	Action	See also
<b>diagram</b>	The checkbox is defaulted to selected to add the new element to the diagram; if you do not want the element in that diagram, deselect the checkbox.	
<b>Create</b>	Click on the <b>OK</b> button to create the element.	

### Notes

- The New Element dialog also displays when you create an element in the **Specification Manager**, if the Specification Type is **<Any>**

### Learn more

- [Adding Elements](#)  (Specification Manager)

## 5.6.1.3 Set Auto Naming and Auto Counters

When you create a new element in your model, you have different options for **naming** the element depending on how you are creating it. If you create the element:

- On a diagram, the system applies a default naming convention of the element **type** plus the increment of the **number** of elements of that type in the same **Package** - for example, *Actor4*; you can change this to any other name by editing the **Name** field on the element's Properties dialog
- Within the Project Browser, the system prompts you to type the element name into the New Element dialog

You can also set up your **own, automatic**, naming and numbering convention for elements of the same type, so that:

- On a diagram, it is **your** naming/numbering convention that is automatically applied to the new element (which you can still change if you need to, as before)
- Within the Project Browser, the New Element dialog **Auto** button applies the naming convention to your new element

If you have **existing** elements in a Package, and you want to apply the new naming convention to them, you can make this change very easily to **all** elements of the same type in the Package at once (see *Learn more*, below).

You define a naming convention for the element type as a model setting. At the same time, you can also optionally define a naming/numbering convention for the **Alias** field for elements of that type.

**Access** [Settings | Auto Names and Counters](#)

**Set up auto naming for an element type**



Field/Option	Action	See also
<b>Type</b>	Click on the drop-down arrow and select the element type to which to apply auto naming/auto numbering.	
<b>Name</b>		
<b>Prefix</b>	Type a prefix for the new name (optional).	
<b>Counter</b>	Type the counter value; use as many 0's as required to pad the number.	
<b>Suffix</b>	Type a suffix for the new name (optional).	
<b>Apply on creation</b>	<p>Select the checkbox to apply auto naming to each new element of the specified type, on creation.</p> <p>If you do not select the checkbox, the autonaming details for the specified element type remain defined in the dialog, but are not applied to new elements of that type. The default element naming convention takes effect (<b>&lt;element type&gt;&lt;n&gt;</b>).</p>	
<b>Alias</b>		
<b>Prefix</b>	Type a prefix for the new Alias (optional).	
<b>Counter</b>	Type the counter value; use as many 0's as required to pad the number.	
<b>Suffix</b>	Type a suffix for the new Alias (optional).	
<b>Apply on creation</b>	<p>Select the checkbox to apply Alias auto naming to each new element of the specified type, on creation.</p> <p>If you do not select the checkbox, the Alias autonaming details for the specified element type remain defined in the dialog, but are not applied to new elements of that type. The default element naming convention takes effect (<b>&lt;element type&gt;&lt;n&gt;</b>).</p>	
<b>Save</b>	<p>Click on the <b>Save</b> button.</p> <p>New elements of this type now have an automatically-generated name and/or Alias with an incrementing counter value.</p> <ul style="list-style-type: none"> <li>• If the user creates an element on a diagram, such as from the Diagram Toolbox or Quicklinker, the element is automatically assigned the defined name and/or Alias</li> <li>• If the user creates an element outside a diagram (such as in the</li> </ul>	<a href="#">Add Elements</a>

Field/Option	Action	See also
	<p>Project Browser) using the New Element dialog, they can click on the <b>Auto</b> button next to the <b>Name</b> and/or <b>Alias</b> field on the appropriate dialog to apply the auto-generated text</p> <ul style="list-style-type: none"> <li>If the user already has some text in the <b>Name</b> or <b>Alias</b> field, it is over-written by the auto-counter text</li> </ul> <p>If an Alias is active, to view the Alias in a diagram requires that the option <b>Use Alias if Available</b> is selected in Diagram Properties.</p>	<p><a href="#">Directly To Packages</a><sup>[903]</sup></p> <p><a href="#">Configure Diagram Display</a><sup>[825]</sup></p>

#### Notes

- Automatic numbering - whether system default or user-defined - does not increment across Package levels; if you have an element called *Actor1* in Package A, and you create an Actor element in a **child** Package of Package A, that Actor will **also** be *Actor1*

#### Learn more

- [Apply Auto Naming to Existing Elements](#)<sup>[906]</sup>

### 5.6.1.4 Apply Auto Naming to Existing Elements

If you have set up **auto-naming** conventions for the names and/or Aliases of specific types of element, you can apply those conventions to all of the **existing** elements of those types in a selected Package, in **one** operation.


You apply auto-naming through the Apply Auto Naming to Elements dialog. This groups the elements that have auto-naming conventions by type, and shows the effects of applying the conventions to the element names and/or Aliases. You can change all the element names in the Package, or select only certain elements to update.

**Access**    **Project Browser | Right-click on Package | Advanced | Apply Auto Naming to Elements**

#### Apply Auto-naming to elements

Field/Option	Action	See also
<b>Current Name</b>	<p>Initially displays the name of each element <b>type</b> that has auto-naming conventions, and the number of elements of that type in the Package.</p> <p>Click on the '+' symbol to expand the type group and list the elements of that type by their current name, with the most recently created element first.</p> <p>Select the checkbox against each element to be renamed, or against the element type name to select all elements in the group. Deselect the checkbox to leave the element name and Alias at their current values.</p>	
<b>Name Preview</b>	Initially shows the elements' current names.	

Field/Option	Action	See also
	When you select the element checkboxes, the names change to show the effect of applying the auto-naming conventions to each element name.	
<b>Alias Preview</b>	Initially shows the elements' current Aliases.  When you select the element checkboxes, the Aliases change to show the effect of applying the auto-naming conventions to each element Alias.	
<b>All</b>	Click on this button to select all elements listed on the dialog.	
<b>Clear</b>	Click on this button to clear the selection of all selected elements.	
<b>Update Name</b>		
<b>Prefix Auto Name to Current Name</b>	Set the change to the auto-name text followed by the current name.	
<b>Overwrite current Name with Auto Name</b>	Replace the current name with the auto-name text.	
<b>Move current Name to Notes Apply Auto Name</b>	Show the current name in the <b>Notes</b> field in the element Properties dialog (in front of any existing notes) and overwrite the element <b>Name</b> field with the auto-name.	
<b>No change</b>	Leave the current names unchanged.	
<b>Update Alias</b>		
<b>Prefix Auto Alias to Current Alias</b>	Set the change to the auto-Alias text followed by the current Alias.	
<b>Overwrite current Alias with Auto Alias</b>	Replace the current Alias with the auto-Alias text.	
<b>Move current Alias to Notes Apply Auto Alias</b>	Show the current Alias in the <b>Notes</b> field in the element Properties dialog (after any replaced element name text, but in front of any existing notes) and overwrite the <b>Alias</b> field with the auto-Alias.	

Field/Option	Action	See also
<b>No change</b>	Leave the current Aliases unchanged.	
<b>Include sub packages</b>	Select this checkbox to include elements in sub-Packages in this run of applying auto-naming (the checkbox reverts to deselected when the dialog is closed).	
<b>Configure Auto Naming</b>	<p>Click on this button to immediately define automatic naming conventions for an element type if:</p> <ul style="list-style-type: none"> <li>You want to change the conventions for an element type, or</li> <li>The element type that you want to apply auto-naming to has no conventions (and is therefore not listed on the dialog)</li> </ul> <p>The Auto Name Counters dialog displays, on which you define your automatic element and/or Alias naming convention. When you click on the <b>Save</b> button on that dialog, any elements of that type in the Package are listed on the Apply Auto Naming to Elements dialog.</p>	<a href="#">Set Auto Naming and Auto Counters</a> 
<b>OK</b>	Click on this button to close the dialog and put the naming changes into effect.	
<b>Cancel</b>	Click on this button to close the dialog and cancel any naming changes you have defined.	

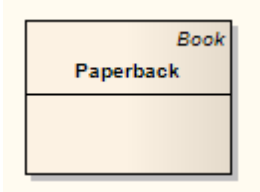
#### Notes

- If you select this facility and there have **never been** any auto-naming conventions defined in the model, the **Current Name** column displays the message *Configure Auto Naming*; click on the **Configure Auto Naming** button to define any conventions for any element type you need
- The contents of Packages that are locked under version control will not be auto-named
- Because you are applying auto-naming to **existing** elements, the auto-naming set up **Apply on Creation** setting has no impact on this operation
- The Apply Auto Naming to Elements dialog is resizable; if you want to expand or reduce the dialog, drag its edges to the size you want

#### 5.6.1.5 Set Element Parent

You can manually set an element's parent or parents, or an interface it realizes (implements), using the Set Parents and Interfaces dialog.

##### Set an element parent

Step	Action	See also
1	Select a generalizable element in a diagram.	
2	<p>Select the <b>Element   Advanced   Set Parents and Interfaces</b> menu option. Alternatively:</p> <ul style="list-style-type: none"> <li>• Press (<b>Ctrl+I</b>) or</li> <li>• Right-click and select the <b>Advanced   Parent</b> context menu option.</li> </ul> <p>The Set Parents and Interfaces dialog displays.</p>	
3	You can elect to enter a parent or interface name by either manually typing it in, or clicking on the <b>Choose</b> button to locate the element within the current model.	
4	Set the <b>Type</b> of relationship ( <b>Implements</b> or <b>Generalizes</b> ) from the drop-down list.	
5	<p>Click on the <b>Add</b> button to add the relationship to the <b>Type Details</b> list, and to the selected child element.</p>  <p>The diagram shows a rectangular box representing a 'Book' element. Inside this box, there is a smaller rectangular box labeled 'Paperback', indicating a generalization relationship where 'Paperback' is a specific type of 'Book'.</p>	
6	Repeat this process to add further parent elements, or click on the <b>Delete Selected</b> button to remove a selected relationship.	

If the parent name is shown in the element on the diagram, you can select it and perform further operations, as follows:

Step	Action	See also
1	Click on the element.	
2	Click on the parent name to highlight it.	
3	Right-click to display the context menu, which provides options enabling you to:	

Step	Action	See also
	<ul style="list-style-type: none"> <li>Redisplay the Set Parents and Interfaces dialog to, for example, delete the parent or add further parents</li> <li>Display the parent element Properties dialog</li> <li>Find the parent element in the Project Browser</li> </ul> <p>Find the parent element in the diagram.</p> <p>If the parent element does not currently exist in the model, and is listed on the element in the diagram, you can highlight it and click on it to redisplay the Set Parents and Interfaces dialog to, for example, delete it or add further parents.</p>	

#### Notes

- You can specify parent elements that exist in the model, or that you intend to create later; if the parent element does not currently exist in the model, select the **Accept classifier even if not in model** checkbox to ensure that the reference to a non-existent element is accepted
- If Parents are not in the same diagram as their corresponding related element, the parentage is shown in the top right corner of the child element

#### 5.6.1.6 Show Element Use

You can display the use of an element using the Element Usage dialog. This lists all occurrences of the element throughout the model, and enables you to easily navigate to any occurrence.

#### How to

To show element usage

Step	Action	See also
1	Select an element in a diagram.	
2	<p>Select the <b>Element   Find in Diagrams</b> menu option (or, for a Package element, the Find in all Diagrams option).</p> <p>Alternatively, press ( <b>Ctrl+U</b> ).</p> <p>If the element exists in other diagrams, the Element Usage dialog displays, listing all occurrences of the current element in diagrams in the model.</p>	
3	<p>If you want to display the usage information in a more readable layout, you can resize the dialog and its columns</p> <p>Either:</p> <ul style="list-style-type: none"> <li>Double-click on a line item to open the relevant diagram and display the selected element (the Element Usage dialog remains open), or</li> <li>Click on the <b>Open</b> button to display the selected diagram and close the Element Usage dialog</li> </ul>	

Step	Action	See also

### Notes

- You can also access this feature from the Project Browser; select an element in the tree and select the **Element | Find in Diagrams** menu option

If there is only one instance of the element in any diagram, that diagram displays instead

### 5.6.1.7 Move Elements Within Diagrams

As you add new elements to a diagram, you might want to move the **existing** elements to create more space, make a structure more apparent, or create a child-parent relationship between two elements. There are several ways in which you can move the elements within the diagram.

#### Move Elements

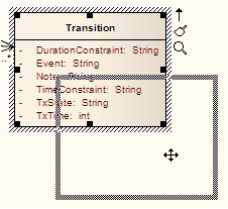
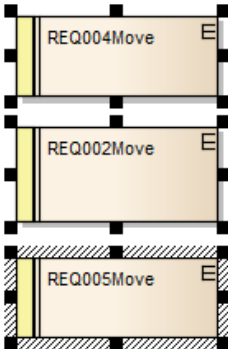

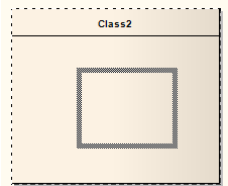
Image	Action	Detail	See also
	<b>Change Position</b>	<p>Select an element or group of elements in the diagram, then:</p> <ul style="list-style-type: none"> <li>Use the mouse to drag the element or group of elements to the required position (the cursor switches to the four-arrow icon as shown)</li> <li>Hold down ( <b>Shift</b> ) and use the arrow keys to move the element or group of elements by small increments to the required position</li> </ul>	
	<b>Align Elements</b>	<p>Select a group of elements in the diagram and align them on the last-selected element using:</p> <ul style="list-style-type: none"> <li>The <b>Align...</b> options in the right-click context menu or</li> <li>The <b>Alignment</b> buttons on the Diagram toolbar</li> </ul> 	<p><a href="#">Operations on Multiple Elements</a> <sup>[95]</sup></p> <p><a href="#">Diagram Toolbar</a> <sup>[138]</sup></p>
	<b>Confirm Possible Parent Elements</b>	<p>If you want to make one element the child of another, you can drag the child onto the parent; the child is then always encapsulated by the parent and moved within the parent.</p> <p>Not all parent/child element combinations are valid. For example:</p> <ul style="list-style-type: none"> <li>Drag a Signal element over a Class; because a Class element <b>can</b> be a</li> </ul>	

Image	Action	Detail	See also
		<p>parent to a Signal, the Signal element is encapsulated</p> <ul style="list-style-type: none"> <li>• Drag a Class element over a Signal element; because a Signal <b>cannot</b> be a parent to a Class, the Class element is not encapsulated</li> </ul> <p>To confirm that the proposed parent element is able to accept the selected child element, observe the parent element border as you drag the child element over it. If the parent can accept the selected element as a child, the element border changes to a dashed line. If the border does not change, the selected element cannot be a child to the parent element.</p>	

#### Notes (on Adding child element to parent)

- The **Support for Composite Objects** checkbox must be selected on the Objects page of the Options dialog (select **Tools | Options | Objects**); if this option is not selected, the dashed border does not show and the child element cannot be embedded on the parent in the diagram
- Both elements must already exist **on the diagram**; the parent element border does not change if you drag a potential child element over it from the **Toolbox** or **Project Browser**
- The child element must be dragged right over the parent, so that its borders are completely within the parent element borders
- The Z-order of the child element is automatically adjusted so that the child is not hidden by the parent
- When you embed a child element on its parent, the child element becomes part of the parent element hierarchy in the Project Browser; if you drag the child element **out of** the parent, the child element becomes independent and is no longer embedded in the parent element hierarchy
- If you have selected the **Structured Compartment** checkbox on the Feature and Compartment Visibility dialog, the child element is confined to the Structured Compartment and cannot be dragged out until the checkbox is deselected again

#### Learn more

- [Object Display Options](#)<sup>[63]</sup>
- [Feature Visibility](#)<sup>[84]</sup>

### 5.6.1.8 Copy Elements Between Diagrams

On a diagram, if you have created a structure of elements such as a State Machine, you might want to copy that structure onto another diagram as a new set of elements without losing the organization of the structure. In doing this, you can change any of the element names within the structure and still preserve the relative nesting of the elements.

**Access** **Source diagram > Right-click selected elements | Copy (Ctrl+C)** then  
**Target diagram > Right-click diagram background | Paste Element(s) as New (Ctrl+Shift+V)**



**Complete the Paste Element(s) as New Dialog**

Field/Option	Action	See also
<b>Original Element Name</b>	Each element name defaults to selected, to be inserted in the target diagram.  If you want to omit an element from the copied structure, clear the checkbox against it.	
<b>New Element Name</b>	If you want to give a copied element a new name, delete the current name and type in the new name.	
<b>Include Connectors</b>	This checkbox defaults to selected, to make copies of the original connectors where both source and target elements in the relationship have been copied. This maintains the relationships between and organization of the copied elements.  If you do not want to copy the connectors, clear the checkbox.	
<b>Paste</b>	Click on this button to add the copied elements and connectors to the target diagram as new elements.	

**Notes**

- Package elements cannot be copied as structured elements
- Elements - including **special** embedded elements such as SysML Properties - do not need to have a parent when pasting as new, although **normal** embedded elements such as Ports must have a parent
- Elements in State regions cannot be pasted as new outside the parent element
- Parenting is still applied when copying and pasting as new using **Ctrl+click-and-drag** between (floating) diagrams

**Learn more**

- [Copy Elements Between Packages](#) <sup>[915]</sup>
- [Paste from Project Browser](#) <sup>[833]</sup>

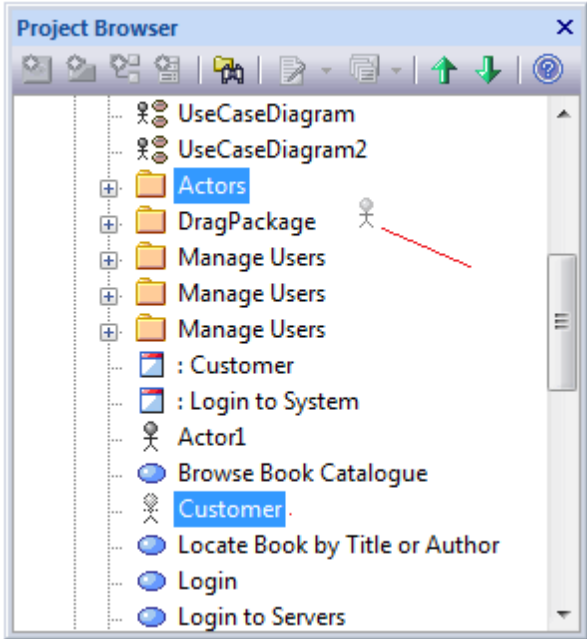
**5.6.1.9 Move Elements Between Packages**

If an element or Package is not in the most appropriate part of the model structure, you can move it to a different location in the Project Browser, either by:

- Dragging and dropping the element to a target destination, or
- Cutting and pasting the element or Package

If you move a Package, ALL the child Packages and their contents are moved to the new location also.

**Drag an element between Packages**

Step	Action	See also
1	Click on the element in the Project Browser. See <i>Customer</i> in the diagram below.	
2	Drag the element so that the cursor is over the target Package icon. The element symbol (and, in some operating systems, the element name) displays at the moving cursor position.  	
3	Release the mouse button. The element is moved into the target Package.	

#### Cut and Paste the Element

Step	Action	See also
1	In the Project Browser, right-click on the element to be moved and select the <b>Copy / Paste   Cut</b> context menu option. The selected element is highlighted.	
2	Right-click on the Package or element under which to move the cut element, and select the <b>Copy / Paste   Paste</b> context menu option. The cut element is listed as a child of the selected element or Package, in the Project Browser.	

Step	Action	See also

### Notes

- You can also drag the element under a host element in the new Package; for example, drag an element under a Class
- Moving an element has no effect on any relationships that the element might have
- Moving an element in the Project Browser has no effect on the use of that element in any diagram
- Moving a diagram generally does not affect the location of elements in Packages - if you move a diagram out of one Package into another, all the elements in the diagram remain in the original Package; however, certain elements (such as Decision, Initial and Final elements) are used only within one diagram, have no meaning outside that diagram, and are never re-used in any other diagram, therefore, if you move a diagram containing these elements they **are** moved to the new parent Package with the diagram
- In a multi-user environment, if one person moves or updates the Project Browser structure, other users must reload their project to see the latest changes in the Project Browser; although this is true of any addition or modification to the tree, it is most important when big changes are made, such as dragging a Package to a different location

### Learn more

- [Refresh View of Shared Project](#)<sup>[30]</sup>

#### 5.6.1.10 Copy Elements Between Packages

You can quickly and easily duplicate one or more elements, including their child elements and diagrams, across packages. You can insert a copy of an element under one or more other packages, in the same project file or any other project file.

### Topics

Topic	Detail	See also
<b>Copying Elements</b>	<p>In the Project Browser, select each required element, right-click on one of them and select the <b>Copy Element(s) to Clipboard</b> context menu option.</p> <p>Alternatively, click on a selected element and press ( <b>Ctrl+C</b> ).</p> <p>The Copy Element(s) to Clipboard dialog briefly displays until the copy operation completes.</p>	
<b>Pasting Elements</b>	<p>In the Project Browser, right-click on the package into which to paste the copied elements, and select the <b>Paste Element(s) from Clipboard</b> context menu option.</p> <p>Alternatively, click on the package and press ( <b>Ctrl+V</b> ).</p> <p>The Paste Element(s) from Clipboard dialog briefly displays until the paste operation completes.</p>	

Topic	Detail	See also
	<p>The target package is expanded and the pasted elements are exposed in the Project Browser. If you are pasting the elements within the same model as the copied source, the source parent package is also collapsed.</p> <p>If the target package already contains:</p> <ul style="list-style-type: none"> <li>• An element of the same type with the same name as a pasted element, the pasted element name has the suffix - <i>Copy</i></li> <li>• An element with the same name as the pasted element including the - <i>Copy</i> suffix, the suffix becomes - <i>Copy1</i> (or - <i>Copy 2</i>, - <i>Copy3</i> and so on, as copies of the element accumulate in the target package)</li> </ul> <p>You can keep the same element names as the source, or you can rename each element either by clicking twice on it and editing the name in the Project Browser, or by double-clicking on it and editing the name in the Properties dialog.</p>	

#### Notes

- A copy of an element does not have the external cross references of the source element; that is:
  - if one element is copied it has no connectors
  - if more than one element is copied, only the connectors between the copied elements are retained
  - however, if those elements come from a Sequence or Communication diagram and the diagram itself is not copied, the message connectors between the copied elements are not retained
- You cannot paste an element into a package that is locked by another user or that is checked in; the **Paste...** option is grayed out in the context menu

#### Learn more

- [Locking Packages](#) <sup>337</sup>
- [Checking In/Checking Out Packages](#) <sup>422</sup>

#### 5.6.1.10.1 Set Up Cross References

It is possible to set up a cross reference (or Custom Reference) from one element in Enterprise Architect to another. You can also view existing cross references on an element, using the Context References tab on the element's Properties dialog, or the Traceability window.

#### Set up a cross reference

Step	Action	See also
1	In the Project Browser, locate the target element or diagram (that is, the object of the cross reference).	

Step	Action	See also
2	Open a diagram that contains the elements that are to have the currently selected element as a reference.	
3	Right-click on the element in the Project Browser. The context menu displays.	
4	Select the <b>Add custom reference</b> menu option.	
5	In the <b>Set up references</b> dialog, select the checkbox against each element to that is to have the target element as a reference.	
6	Optionally, in the <b>Comment</b> field, type some text to describe the purpose of the reference.	
7	Click on the <b>OK</b> button.	

#### Use the cross reference

Step	Action	See also
1	Select an element in a diagram.	
2	Select the <b>Element   Advanced   Custom References</b> menu option. Alternatively, either press ( <b>Ctrl+J</b> ), or right-click on the element and select the <b>Find   Custom References</b> context menu option.	
3	The Custom References dialog displays, showing a list of elements that have been set as cross references for the selected element.	
4	You can open the Properties dialog for an element by highlighting it and clicking on the <b>Open</b> button.	
5	If you have a diagram cross reference, you can open that diagram.	
6	If you have a string of diagram links, click on the <b>Home</b> button to return to the original diagram.	

**Notes**

- You can delete a cross reference by selecting it on the Custom References dialog and clicking on the **Delete** button; cross references are also automatically deleted if the source or target element in the reference is deleted

**Learn more**

- [Context References Tab](#)<sup>[987]</sup>
- [The Traceability Window](#)<sup>[725]</sup>

**5.6.1.11 Change Element Type**

**Access** **Element | Advanced | Change Type**

Select the new element type from the list and click on the **OK** button.

The target element is transformed into the required type.

**5.6.1.12 Align Elements**

When you have created a number of elements on a diagram, you might want to organize them in groups so that they form a column or row with a uniform edge. You can do this quickly and easily by selecting the group of elements and aligning them all against the left, right, top or bottom edge or center of one element in the group. You can also space three or more elements evenly in a row or column, at an average of their previous separations.

**Access** **On diagram | select elements | right-click 'master' element |**  
**Align Left**  
**Align Right**  
**Align Top**  
**Align Bottom**  
**Align Centers (Align Vertically or Align Horizontally)**  
**Space Evenly (Across or Down the page)**

**Align multiple elements**

Step	Action	See also
1	Select a group of elements by drawing a selection box around them all (or select them one by one by holding down ( <b>Ctrl</b> ) and clicking on each element).	
2	Right-click on the element in the group to align others to. The context menu displays.	
3	Select the alignment function you require. All selected elements are aligned to the one beneath the cursor.	

**Notes**

- You can also use the first four buttons of the Diagram toolbar to align elements; these are made available when more than one element is selected in a diagram

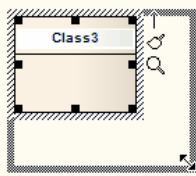
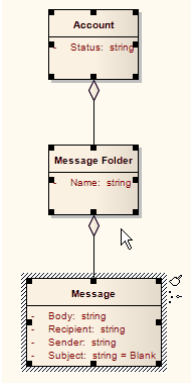
**Learn more**

- [Operations on Multiple Elements](#) <sup>951</sup>
- [Diagram Toolbar](#) <sup>138</sup>
- [Position Elements](#) <sup>953</sup>

**5.6.1.13 Resize Elements**

Any one of the following options enables you to resize an element.

**Topics**

Images	Topic	Detail	See also
	<b>Resize handles</b>	<p>Select an element or group of elements in the diagram view, then:</p> <ul style="list-style-type: none"> <li>Use the resize handles that appear at each corner and side to resize the element(s) by dragging with the mouse (the cursor switches to the double-ended arrow as shown in the image to the left)</li> <li>Press and hold ( <b>Ctrl</b> ) and use the arrow keys to resize by increments as required</li> <li>Autosize selected element(s) using the option in the <b>Element   Appearance</b> submenu, or by pressing ( <b>Alt+Z</b> ) (with multiple elements selected, <b>Autosize</b> also appears in the right-click context menu)</li> <li>Set multiple elements to the same height, width or both, using these options in the right-click context menu</li> </ul>	
	<b>Resize a Set of Objects to a specific size</b>	<p>If you right-click a selected set of objects, you can resize them to the same dimensions (height, width or both) using the context menu.</p> <p>When you select multiple elements using ( <b>Ctrl</b> ) + <b>click</b>, then resize the dimensions, the dimensions of the selected hatched object are used to set the dimensions of the other selected objects.</p> <p>For example, in the diagram to the left, the <i>Message</i> Class height and width are used to set the height and width of the <i>Account</i> and <i>Message Folder</i> Classes.</p> <p>The aim is to make the <i>Account</i> and <i>Message</i></p>	

Images	Topic	Detail	See also
		<i>Folder</i> elements the same height and width as the <i>Message</i> element.	

#### Resize a set of objects to a specific size

Step	Action	See also
1	Set one element to the required size (for example, <i>Message</i> as above).	
2	Select all other elements (for example, <i>Account</i> and <i>Message Folder</i> as above).	
3	Right-click on the pre-sized element (for example, <i>Message</i> ).	
4	Select your resizing option (such as same height or width) from the context menu.	

#### Resize single or multiple elements by increments

Action	Keys
Make all selected elements wider.	<b>Ctrl+→</b>
Make all selected elements narrower.	<b>Ctrl+←</b>
Make all selected elements shorter.	<b>Ctrl+↑</b>
Make all selected elements taller.	<b>Ctrl+↓</b>

#### Learn more

- [Highlight Context Elements](#) 



### 5.6.1.14 Delete Elements from Diagram and Model

#### Delete an element from a diagram

Step	Action	See also
1	In the active diagram, click on the element to delete.	
2	Either: <ul style="list-style-type: none"> <li>Press <b>Delete</b> or</li> <li>Right-click to display the context menu and select the <b>Delete &lt;element name&gt;</b> option</li> </ul> <p>This does not delete the element from the model, only from the current diagram.</p>	

#### Delete an element from the model

Step	Action	See also
1	In the Project Browser, right-click on the element to delete and select the <b>Delete &lt;element name&gt;</b> context menu option. A confirmation prompt displays.	
2	Click on the <b>Yes</b> button.	

Alternatively, click on the element in the Project Browser and press **Ctrl+Delete**. The element is completely removed from the model.

#### Delete multiple elements from a diagram

Step	Action	See also
1	In the active diagram, press <b>Ctrl</b> and click on each element to delete.	
2	Either: <ul style="list-style-type: none"> <li>Press <b>Delete</b>, or</li> </ul>	

Step	Action	See also
	<ul style="list-style-type: none"> <li>Right-click to display the context menu and select the <b>Delete selected elements</b> option</li> </ul>	

#### Delete multiple elements from a diagram and model

Step	Action	See also
1	Open the diagram containing the elements to remove from the model.	
2	Press <b>Ctrl+A</b> to select all of the elements in the diagram, or press <b>Ctrl</b> and click to select specific elements.	
3	Press <b>Ctrl+Delete</b> to completely remove the elements from the model.	

#### Delete multiple elements from the Project Browser and model

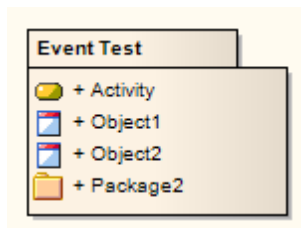
Step	Action	See also
1	In the Project Browser, press and hold either <b>Shift</b> or <b>Ctrl</b> and click on the required items.	
2	<p>To completely remove the elements from the model, either:</p> <ul style="list-style-type: none"> <li>Press <b>Ctrl+Delete</b>, or</li> <li>Right-click on the selected items and select the <b>Delete selected item(s)</b> context menu option</li> </ul> <p>If you delete an element in this way, you delete all its properties and connectors as well.</p>	

#### **5.6.1.15 Customize Visibility of Elements**

Some elements are hidden from view in Packages and in report documents by default. These include Events, Decisions, Sequence elements and Associations. You have the option of turning these elements back on.

For example, some Events and Decisions contained in a Package do not appear in the Package view, as

shown.



Images	Step	Action	See also
	1	Select the <b>Tools   Options   Objects</b> menu option. The Objects page of the Options dialog displays.	
	2	Click on the <b>Advanced</b> button. The Advanced Settings dialog displays.	
	3	Select the checkbox for each type of element to show in packages and in report documents.	
	4	Click on the <b>Close</b> button on each dialog.	
	5	<b>Reload</b> the current diagram if required. The package from the example shows the Event and Decision elements it now contains (see left image).	<a href="#">Refreshing the view of a shared project</a> <sup>[308]</sup>

### 5.6.1.16 Create Notes and Text

You can create both notes and text on a diagram; the two are slightly different.



#### How to

To create a Note

Step	Action	See also
1	Drag the <b>Note</b> icon from the Common page of the Toolbox onto the diagram. <ul style="list-style-type: none"> <li>If you have the <b>Edit Object On New</b> checkbox deselected on the Objects page of the Options dialog, the Note element displays on your diagram; type your note text directly within the Note element</li> <li>If you have the checkbox selected, the Notes <i>window</i> displays; type your text in that window</li> </ul>	<a href="#">Object Display Options</a> <sup>[631]</sup> <a href="#">Notes</a> <sup>[1142]</sup> <a href="#">Notes Toolbar</a> <sup>[1143]</sup>

Step	Action	See also
	<p>If you want to display the Notes information in a more readable layout, you can resize the window</p> <p>You can format the text if necessary, using the Notes toolbar at the top of the window; when you have completed the text, click on the <b>OK</b> button to save it</p>	

To create a Text element

Step	Action	See also
1	<p>Drag the <b>Text Element</b> icon from the Common page of the Toolbox onto the diagram.</p> <p>The Notes window displays.</p>	<a href="#">Notes</a> 
2	<p>Type your text in the window.</p> <p>If you want to display the Notes information in more readable layout, you can resize the window.</p> <p>You can format the text if necessary, using the Notes toolbar at the top of the window; when you have completed the text, click on the <b>OK</b> button to save it.</p>	<a href="#">Notes Toolbar</a> 

### Notes

- You can also create a note by clicking on the **New Note** icon (the text page) on the UML Elements toolbar and clicking on the diagram

### Learn more

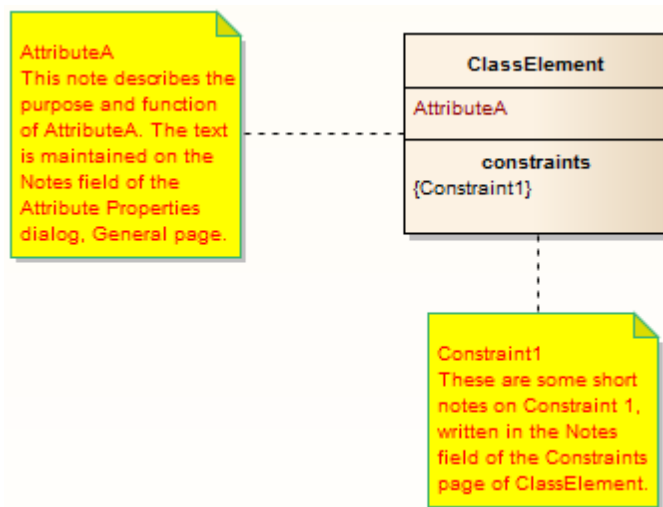
- [UML Elements Toolbar](#) 

#### 5.6.1.17 Link Note to Internal Documentation

It is possible to connect a *Note* element to another element's internal documentation. This enables you to externalize model documentation to the diagram level, and as Enterprise Architect keeps the Note and the internal structure in synch, you do not have to update the Note; in fact, you are blocked from editing the Note text, because it is maintained automatically.


In a similar way, it is also possible to connect a *Note* element to the notes property of the diagram which it is displayed on.

In the example below, two notes are connected into an element's internal structures. One is connected to an attribute, and displays the attribute name and notes. The other is connected to a constraint, showing the constraint name and documentation.



### How to

To connect a Note element to another element's documentation

Step	Action	See also
1	Click on the element and then click on the feature so that it is highlighted for inline editing.	<a href="#">Inline Features Submenu</a> 
2	Select the <b>Element   Inline Features   Create Linked Note</b> menu option. This creates a Note element linked to the selected feature, reflecting the content of that feature.	

Alternatively:

Step	Action	See also
1	Insert the target element into a diagram.	
2	Drag the <b>Note</b> icon from the Common page of the Toolbox onto the diagram, next to the target element. The Notes dialog displays.	
3	Do not type any text, just click on the <b>OK</b> button.	

Step	Action	See also
4	Click on the <b>Note Link</b> icon from the Common page of the Toolbox.	
5	Click on the Note, and drag across to the target element to create the connector.	
6	Right-click on the Note Link to display the context menu.	
7	Select the <b>Link this Note to an Element Feature</b> context menu option. The Link note to element feature dialog displays.	
8	In the <b>Feature Type</b> field, click on the drop-down arrow and select the type of feature to link to.	
9	In the <b>Feature</b> list, click on the specific feature to link to.	
10	Click on the <b>OK</b> button. The note now automatically derives its contents from the target element.	

To connect a Note element to the diagram notes property

Step	Action	See also
1	Open the diagram.	
2	Select the <b>Diagram   Properties</b> menu option, enter an appropriate description in the <b>Notes</b> field, then click on the <b>OK</b> button.	<a href="#">General Tab</a> [824]
3	Drag the <b>Note</b> icon from the Common page of the Toolbox onto the diagram. The Notes dialog displays.	
4	Do not type any text, just click on the <b>OK</b> button.	
5	Right click the Note element and select the <b>Advanced   Link to Diagram Note</b> option. The note now automatically derives its contents from the diagram's notes property.	

**Notes**

- You can unlink a Note linked to an element feature, leaving the Note text, which can be edited manually and directly as normal; to unlink the Note from an element feature, either:
  - Right-click on the Note and select the **Advanced | Unlink from Element Feature** context menu option, or
  - Right-click on the Notelink connector, select the **Link this Note to an Element Feature** context menu option and, on the Link note to element feature dialog, set the **Feature Type** field to **None**

**5.6.1.18 Set an Element's Default Appearance**


To override the **global** element appearance settings and define a default appearance for a **specific element** on all diagrams on which it is found, you use the Default Appearance dialog. You can alter:

- The background, border and font colors
- Font styles, including font face and size
- The border thickness for an element

An alternative option for changing the element's appearance is to define a stereotype that has different fill, border and font settings, or either a Shape Script or an image file associated with it. You can then assign the stereotype to the element to change the element's appearance to that defined by the stereotype.

**Access** Click on the element, **Element | Appearance | Default Appearance** (Ctrl+Shift+E)

**Options**

Field/Option/ Button	Action	See also
<b>Background Color</b>	<p>Click on the drop-down arrow and select the background - or fill - color of the element.</p> <p>This setting overrides all other fill color definitions in the model, <b>except</b> for an element-specific definition from the Format Toolbar, or a color defined in a Shape Script and applied through a stereotype.</p> <p>If you want to use a color that is not available, click on the <b>Other</b> button and define the required color.</p>	<a href="#">Create Custom Colors</a> 
<b>Border Color</b>	<p>Click on the drop-down arrow and select the border color for the element.</p> <p>If you want to use a color that is not available, click on the <b>Other</b> button and define the required color.</p>	
<b>Border Width</b>	<p>Type in the width of the border surrounding the element, or use the increment/decrement arrows to the right of the field.</p> <p>The minimum width setting is <b>1</b>, the maximum is <b>5</b>.</p>	
<b>Font Style</b>	<p>Display the current font typeface for the element text. To change this, click on the (...) button.</p>	

Field/Option/ Button	Action	See also
	<p>The Font dialog displays, on which you can select a different typeface, style (such as bold/italic) and font size.</p> <p>You can also select from a limited number of font colors and effects on this dialog.</p>	<a href="#">Font Dialog</a> <sup>[94]</sup>
<b>Font Color</b>	<p>Click on the drop-down arrow and select the font color to apply to the text of the element.</p> <p>If you want to use a color that is not available, click on the <b>Other</b> button and define the required color.</p>	Notes, below <a href="#">Create Custom Colors</a> <sup>[619]</sup>
<b>Reset to defaults</b>	(Optional) Click on this button to reset all appearance settings to the system default values.	
<b>Preview</b>	Displays a representation of the colors, border width and font styles, which updates as you choose the new settings.	
<b>OK</b>	Click on this button to close the dialog and <b>apply</b> the appearance settings you have defined.	
<b>Cancel</b>	Click on this button to close the dialog and <b>abort</b> all the changes you have made within the dialog.	

### Notes

- You can adjust several elements at the same time; select all of the required elements, right-click on one of them and select the **Appearance | Default Appearance** context menu option, or use the Format toolbar
- If the **Show Project Custom Colors in Color Combo Boxes** checkbox is selected on the Options dialog, Standard Colors page, the last 16 color squares in the color selection palettes contain the project colors set on that dialog
- The **Font Color** field affects only the element name text; it does not override the feature text colors as defined on the Options dialog, Standard Colors page
- When it has been called directly on one or more selected elements (**Appearance | Set Font**), the Font dialog overrides the Default Appearance dialog
- When called from the Default Appearance dialog, the **Color** field on the Font dialog is overridden by the **Font Color** field on the Default Appearance dialog
- The **Font** and **Font Style** fields on the Font dialog affect all text except for the Note compartment text
- The **Size** field on the Font dialog affects all text

### Learn more

- [Standard Colors](#) <sup>[616]</sup>



- [Diagram Appearance Options](#) <sup>[622]</sup>
- [Format Toolbar](#) <sup>[785]</sup>
- [Apply Stereotypes](#) <sup>[1453]</sup>
- [Custom Stereotypes](#) <sup>[1461]</sup>

### 5.6.1.19 Set Element Templates Package

In building up a model, you might want to represent or emphasize certain characteristics of elements in the appearance of those elements, or select particular display options as standard. For example, you could make new Interface elements a different default color to new Class elements, ensure all new Activity Partitions are vertical rather than horizontal, or set a specific group of display options for new diagrams; you could also define a set of characteristics to use for each development stage of a project.

To do this, you create a diagram with all the characteristics you require, and store the diagram in an Element Templates package; the system then checks this package:

- Whenever you start to create an element in a diagram or
- Whenever you create a new diagram

If it finds a template for that diagram type, the system applies the settings in that template to the new element or to the display options of the diagram. For example, you could save a diagram under the name *ClassTemplate*, to apply a set of display characteristics to all new Class elements and Class diagrams.

You can create the Templates package anywhere in your model; however, it is better to create it in a location that is not likely to be accidentally changed or lost in any project development work.

#### Set up the element Templates package

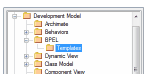
Image	Step	Action	See also
	1	Create a new package. You can give this package any name; <i>Templates</i> is an unambiguous option.	
	2	Within the Templates package create new diagrams, one for each type of diagram to template.  Give them easily recognized names; for example <i>ClassTemplate</i> for the template for Class diagrams.	
	3	Add new elements to the template diagrams from the Toolbox, and configure the size, appearance, notes, version and other properties.	
	4	Select the <b>Settings   Project Template Package</b> menu option to set the templates as the default element templates.  The Browse Project window displays.	
	5	Locate and click on the Templates package, and click on the <b>OK</b> button to set the package as the default element template.  Now each new element or diagram you add to your project is created with the settings from the appropriate Template diagram.	

Image	Step	Action	See also

### Notes

- The **Fill color** for a type of element defined in the element template can be overridden by the element fill color defined from the Format Toolbar, a Shape Script applied to the element, or the Default Appearance dialog
- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Reference Data - Update** permission to set up or access the element Templates package
- If you decide not to use the default element template, set the default element template to **<none>** in the Browse Project window; the **<none>** package is at the bottom of the hierarchy shown in the Browse Project window
- There are two other ways in which you can modify the standard appearance of elements in diagrams:
  - Define the default appearance of elements (and other structures) grouped in a diagram using UML Profiles, which provide a means of extending the UML to build UML models in particular domains; Profiles are based on additional stereotypes and Tagged Values that are applied to elements, attributes, methods, connectors and connector ends
  - Modify the appearance of elements or connectors of a specific type using **stereotypes, which take precedence over templates**; if you drop a **stereotyped** Class onto a diagram, the stereotype immediately defines the Class appearance so the template is not accessed

Stereotypes are much more flexible for defining the appearance of an element under different scenarios

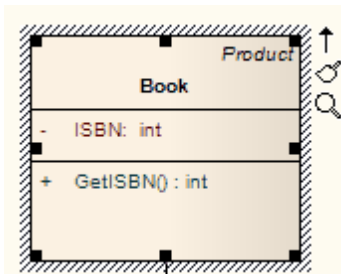
### Learn more

- [Set an Element's Default Appearance](#)<sup>[927]</sup>
- [Format Toolbar](#)<sup>[785]</sup>
- [Drawing Methods](#)<sup>[1591]</sup>
- [Using UML Profiles](#)<sup>[1472]</sup>
- [UML Stereotypes](#)<sup>[1452]</sup>

## 5.6.1.20 Highlight Context Element

**Access** **Tools | Options | Diagram | Behavior**

You can show a hatched border around a selected element by selecting the **Always Highlight Context Element** checkbox on the Diagram Behavior page. If you have selected this checkbox, the selected element displays similarly to the following example:



If you have not selected the **Always Highlight Context Element** checkbox, the selected element does not have a hatched border around it.

### Topics

Images	Topic	Detail	See also
	<b>Multiple Selections</b>	<p>Whether you have selected the <b>Always Highlight Context Element</b> checkbox or not, if you select multiple elements one of the elements you select always has a hatched border.</p> <p>If you align the elements, this element is the one used to align the other elements against.</p> <p>For example, if the elements in the diagram are aligned, the top element aligns to the bottom element (the element showing a hatched border).</p>	
	<b>Change the Element to Align Against</b>	To change which element has a hatched border in a selected group (and thus the element that is aligned with) click on the element that the other elements are to align with.	

### 5.6.1.21 Make Linked Element a Local Copy

#### How to

To convert a linked element to a local copy

Step	Action	See also
1	Open the diagram with the linked element.	
2	Select the linked element and right-click on it to display its context menu.	
3	<p>Select the <b>Convert Linked Element to Local Copy</b> menu option.</p> <p>The element changes to a local copy and is placed in the appropriate package.</p>	


Step	Action	See also

### 5.6.1.22 Copy Features Between Elements

Using drag and drop, you can copy attributes and/or operations from an element in the Project Browser or Element Browser on to another element in a diagram.


#### How to

To copy an element feature

Images	Step	Action	See also
	1	Open a diagram that contains the target element.  In the example to the left, the <i>AccountItem</i> Class is the target and the <i>Customer</i> element is the donor - this is depicted for the Project Browser but is almost identical for the Element Browser.	
	2	Click on the attribute or operation and drag it to the target element.	
	3	Release the mouse button.	

The image below shows *AccountItem* after the attribute *Account* has been dropped from the browser on to it.

To copy multiple element features from the Project Browser (not the Element Browser)

Images	Step	Action	See also
	1	Open a diagram that contains the target element (as in the example above, the <i>AccountItem</i> Class is the target and <i>Customer</i> is the donor).	
	2	Hold down ( <b>Ctrl</b> ) (separate features) or ( <b>Shift</b> ) (select a range) and click on the attributes and/or operations to copy, then drag the selected features to the target element.	
	3	Release the mouse button.	

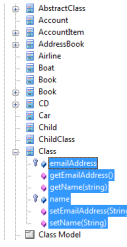
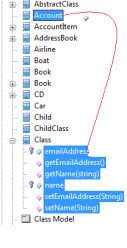
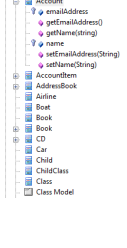
#### Learn more

- [Attributes](#) <sup>999</sup>
- [Operations](#) <sup>1014</sup>
- [The Project Browser](#) <sup>646</sup>
- [The Element Browser](#) <sup>989</sup>
- [Move Features Between Elements](#) <sup>933</sup>

### 5.6.1.23 Move Features Between Elements

Using drag and drop, you can move attributes and/or operations from an element in the Project Browser on to another element within the Project Browser.

#### Move element features

Images	Step	Action	See also
	1	In the Project Browser, locate the attributes and/or operations to move from the target element and select them while holding down: <ul style="list-style-type: none"> <li>• <b>(Ctrl)</b> (single item select) or</li> <li>• <b>(Shift)</b> (multiple item select)</li> </ul>	
	2	Holding down the mouse button, drag the attributes and/or operations to the target element.  A single feature symbol (and, under some operating systems, the feature name) displays during the move; however all of the selected features are moved.	
	3	Release the mouse button.  The image to the left shows the final stage of the attribute and operations move between the <i>Class</i> element and the <i>Account</i> element.	

#### Learn more

- [Attributes](#) <sup>999</sup>
- [Operations](#) <sup>1014</sup>
- [Copy Features Between Elements](#) <sup>932</sup>

### 5.6.1.24 Insert Related Elements

Elements can be used more than once in a model, either individually or as a group. If you want to reproduce an existing set of elements that have one element as the focus, without copying and linking the elements individually in the new diagram, you can use the **Insert Related Elements** facility. With this facility, you drag the focus element from the Project Browser onto your diagram, right-click on it and select to add related elements. The Insert Related Elements dialog displays, on which you can specify:

- Which types of relationship to filter on, and what direction these relationships have
- Which types of element to filter on
- Which package the related elements exist in

- How many levels of relationship to bring in; that is, just the elements immediately related to the focus element, or also elements related through those elements
- Whether to add all elements that satisfy the criteria you have specified, or just those you select
- Whether or not to automatically re-organize the diagram when the related elements have been added

**Access** **Right-click on element | Insert Related Elements**

**Define selection criteria to insert related elements**

Field/Button	Action	See also
<b>Connector types</b>	<p>Displays the <b>types</b> of <b>relationship</b> in which the selected element and, if more than one level of relationship is selected, its related elements are currently involved.</p> <p>To limit the inserted elements to those connected by specific relationship types, select the checkbox against each of those types of connector.</p> <p>To deselect a selected checkbox, click on it again.</p>	
<b>Element types</b>	<p>Display the <b>types</b> of <b>element</b> to which the selected element and, if more than one level of relationship is selected, its related elements are connected.</p> <p>To limit the inserted elements to those of specific types, select the checkbox against each of those types.</p> <p>To deselect a selected checkbox, click on it again.</p>	
<b>Link Direction</b>	<p>Limit the inserted elements to those in incoming or outgoing relationships with the selected element; click on the drop-down arrow and select the appropriate direction, or <b>&lt;All&gt;</b> for either direction.</p>	
<b>Limit to Namespace</b>	<p>Restrict the inserted elements to those from a specific package; select the namespace from which the elements are to come. Select <b>&lt;None&gt;</b> for elements in any namespace.</p>	
<b>Find relationships to: levels</b>	<p>Click on the drop-down arrow and select the level down to which to insert connected elements, between levels <b>1</b> (immediately related elements) and <b>5</b> (elements 5th in a chain from the focus element).</p> <p>You can select levels <b>4</b> or <b>5</b> to see how far the element/relationship hierarchy extends, but as this can produce a complicated and tangled diagram, it is better to use level <b>1</b> or <b>2</b> on selected elements in turn.</p>	
<b>Refresh</b>	<p>Click on this button to locate and list all elements that satisfy the criteria you have specified.</p> <p>Each list item shows the depth at which the related element exists in relation to the focus element, the name of the element, its parent package, and whether it is currently in the diagram.</p>	<a href="#">Customizing the Search</a>

Field/Button	Action	See also
	<p>If there are many entries, you can re-organize them for clarity using the <b>Drag a column header...</b> panel.</p> <p>If you want to add an element to your diagram, select the checkbox against that element. Alternatively, to select all elements, click on the <b>All</b> button.</p> <p>To deselect all selected checkboxes, click on the small <b>Clear</b> button.</p> <p>To empty the list panel altogether, click on the large <b>Clear</b> button.</p>	<a href="#">View</a> <sup>[709]</sup>
<b>Layout Diagram when complete</b>	<p>Select the checkbox if you want the system to layout the diagram after the elements have been inserted; the layout applied is the Digraph<sup>[883]</sup> layout.</p> <p>If no elements have been added, this option has no effect; elements have to be added for the system to adjust the layout.</p>	<a href="#">Digraph Layout</a> <sup>[883]</sup>
<b>OK</b>	Click on this button to add the selected elements to the diagram and close the dialog.	

#### Notes

- If the selected element has a relationship to an edge-mounted structural element such as a **Port** or **Exposed Interface**, adding the Port or Interface to the diagram using the Insert Related Elements dialog will also add the associated parent elements onto the diagram, if they are not already shown.

#### 5.6.1.25 Manage Structural Elements

You can add a number of structural - or embedded - elements to a modeling element. For example, you can add Ports to a Class, Action Pins to an Action, or Activity Parameter to an Activity. These structural elements cannot exist by themselves, they must be attached to a parent element.

**Access** **Diagram | right-click on element | Structural Elements**

#### Create and manage Structural Elements

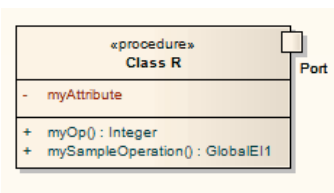
Image	Detail	See also
	<p>On the Structural Elements dialog, click on the <b>New</b> button to create a new embedded element. The Insert New Embedded Element dialog displays.</p> <p>The <b>Type</b> field defaults to the first in the list of types of structural element you can create for the selected element. If this is not the type you need, click on the drop-down arrow and select another type.</p> <p>Enter details such as type, name and stereotype, and click on the <b>OK</b> button.</p> <p>The embedded element now shows on the primary</p>	

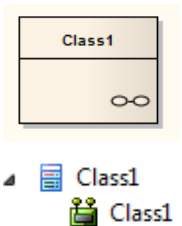
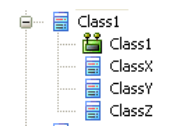
Image	Detail	See also
	<p>element as shown on the left.</p> <p>You can add as many embedded elements as you need; modify or delete embedded elements using the Structural Elements dialog.</p> <p>To incorporate inherited or owner properties, select the <b>Show Owned/Inherited</b> checkbox.</p> <p>The name of the embedded element is a label, which you can edit using the <b>Labels</b> context menu.</p>	<p><a href="#">Manage Object Labels</a> [866]</p>

### 5.6.1.26 Composite Elements

You might want to make one element represent a structure composed of other elements; for example, a general Use Case representing a collection of more specific Use Cases. To achieve this, you can make the 'general' element a **composite** element, as a pointer to a child diagram containing the 'detail' elements. The elements that you can turn into composites include Classes, Objects, Activities and Use Cases.

**Access** [Right-click element in diagram | New Diagram | Composite Structure Diagram](#)  
[Right-click element in Project Browser | Add | Composite Structure Diagram](#)

#### Make an element composite

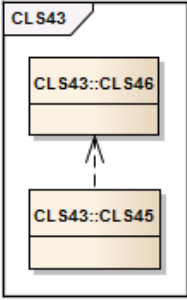
Images	Step	Action	See also
	1	<p>When you select to make an element composite, it displays in the diagram and in the Project Browser as illustrated on the left.</p> <p>Note:</p> <ul style="list-style-type: none"> <li>The small icon in the bottom right hand corner of the element shape, indicating that this is now a Composite element</li> <li>The icon against the child diagram name in the Project Browser, indicating that this diagram was created as a child of the composite element</li> </ul>	
	2	<p>Double-click on the Composite element to access the child diagram that it points to. You can populate the child diagram with elements and connectors to represent the internal structure of the (parent) composite element.</p> <p>At this point, embedded elements cannot be added to the child composite diagram. A modified procedure enables you to include them.</p> <p>The Composite element and its child diagram are now represented in the Project Browser as shown. <i>ClassX</i>, <i>ClassY</i> and <i>ClassZ</i> are elements added to the child diagram.</p>	<p><a href="#">Show Embedded Elements In Composite</a> [938]</p>

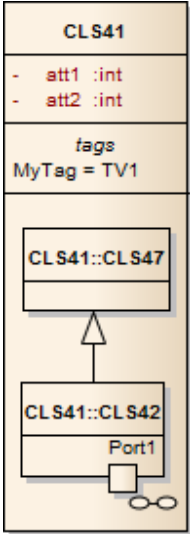


Images	Step	Action	See also
	3	<p>If you already have, or subsequently create, another diagram that better represents the content of your composite element, you can select <b>that</b> diagram as the child of the composite.</p> <p>Right-click on the composite element and select the <b>New Diagram   Select Composite Diagram</b> context menu option. The Select a Diagram browser displays, through which you can browse the entire project to locate the alternative diagram. Click on the diagram and on the <b>OK</b> key to apply it as the child of the composite element.</p> <p>Neither the original nor the alternative child diagram move location or change icons in the Project Browser (unless you specifically make these changes as a separate operation).</p>	

### Alternative Notation

You can display the contents (the child diagram) of a Composite element within the element, in two ways.

Display Format	Detail	See also
<p><b>Display Composite Element as Frame</b></p> 	<p>Composite elements can show their contents <b>in place of</b> their usual notation.</p> <ol style="list-style-type: none"> <li>1. Right-click on the element to open the context menu.</li> <li>2. Select the <b>New Diagram   Show Composite Diagram in Frame</b> option.</li> </ol> <p>The element is replaced by a frame containing the child diagram. The frame label shows the composite element name. The element is resized to contain the diagram; whilst it can be enlarged, it cannot be reduced to less than the diagram size.</p> <p>Some technologies, such as SysML, qualify the frame label to show the diagram type, element type, element name and child diagram name. If you design custom diagram profiles, you can also qualify the frame label as required.</p>	<p><a href="#">Create Custom Diagram Profiles</a> <sup>[1570]</sup></p>
<p><b>Display Composite Element Content In Compartment</b></p>	<p>Composite elements can also show their contents <b>in addition to</b> their usual notation, in an additional compartment of the element.</p> <ol style="list-style-type: none"> <li>1. Right-click on the element to open the context menu.</li> <li>2. Select the <b>New Diagram   Show Composite Diagram in Compartment</b> option.</li> </ol> <p>A compartment is added to the element, containing the child diagram. The element is resized to contain the diagram; whilst it can be enlarged, it cannot be reduced to less than the diagram size.</p> <p>You can also display a Composite Diagram inside its parent element,</p>	<p><a href="#">Show Composite</a></p>

Display Format	Detail	See also
	<p>using a Shape Script. The diagram is displayed in a <b>custom</b> compartment in the <b>center</b> of the element and not necessarily with the standard compartments (such as attributes, tags).</p>	<p><a href="#">Diagram</a> <sup>[1608]</sup></p>

### Notes

- If the **Composite Structure Diagram** option is not listed in the context menu, the option is not available for the type of element you have selected
- If you select either of the alternative notation options to display the child diagram **on** the composite element, and then you select a **different** child diagram, the display settings are cleared; reselect the appropriate option to display the new diagram on the composite element
- Automation support is available for Composite elements; *Element* has an *Elements* collection and a *Diagrams* collection. Additionally you can set the Composite diagram with *SetCompositeDiagram*.

### Learn more

- [Element Class](#) <sup>[2881]</sup>

#### 5.6.1.26.1 Show Embedded Elements In Composite

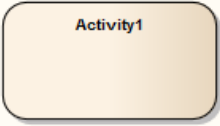
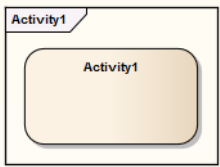
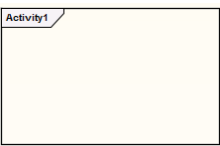
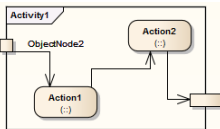
A Composite element shows the external relationships of the element with other components of the model and, via a child diagram, its own internal structure. You cannot add embedded elements to the initial representation of this *internal* structure; this procedure explains how to add embedded elements to the child diagram.

Embedded elements include:

- ActivityParameter
- RequiredInterface
- ActionPin
- Port
- EntryPoint
- ExpansionNode
- ObjectNode
- ProvidedInterface
- ExitPoint

- Part

#### Include embedded elements in a child composite diagram

Images	Step	Action	See also
	1	Create the element to set as a Composite element. For example, a Class, Activity, or SysML Block.	
	2	Right-click on the element in the diagram and select the <b>New Diagram   Composite Structure Diagram</b> context menu option. The element redisplay with a small icon in the bottom right hand corner, indicating that this is now a Composite element.	<a href="#">Composite Elements</a> [936]
	3	Double-click on the Composite element to access the child diagram that it points to.	
	4	From the Project Browser, drag the Composite element itself onto the child Composite diagram, and paste it in as a Simple Link.	
	5	Right-click on the element in the diagram, and select the <b>New Diagram   Show Composite Diagram in Frame</b> context menu option. The element redisplay with a labeled frame around it.	
	6	Press <b>(Ctrl+S)</b> to save the diagram. Right-click on the diagram tab and select the <b>Reload &lt;diagram name&gt;</b> context menu option. The diagram redisplay with just the frame showing. This frame represents the original composite element.	
	7	Populate the frame with the appropriate internal elements, including the embedded elements that attach to the frame itself.	

## 5.6.2 Operations on Elements

You can perform a wide range of operations on element objects on a diagram, such as displaying an element's structural components, displaying or hiding its features on the diagram, and changing the element's appearance. You access these operations through the **element context menu**. If you are working under a technology such as SysML or BPMN, the menu can also provide options that are specific to that technology.

**Element Context Menu Options on Diagrams**

Option	Action	Shortcut	See also
<b>&lt;Technology Options&gt;</b>	For some technologies the context menu begins with technology-specific options. Refer to the topics on elements specific to the technology you are working in, such as BPEL, BPMN and SysML.		
<b>Properties</b>	Open the Properties dialog for the selected element.  For <b>State Lifeline</b> and <b>Value Lifeline</b> elements, display the Configure Timeline dialog. For a <b>Required Interface</b> element, display the Exposed Interface dialog.	<b>Alt+Enter</b>	<a href="#">Properties dialog</a> [956] <a href="#">Configure Timeline - States</a> [1237] <a href="#">Expose Interface</a> [1375]
<b>Lock Element</b>	Lock the element so it can't be edited; to unlock the element, select <b>Lock Element</b> again.  This does not apply in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions <b>when security is enabled</b> ; in that situation, use the <b>Lock</b> option for model elements.		<a href="#">Lock model elements</a> [336]
<b>Other Properties</b>	For State Lifeline, Value Lifeline and Required Interface elements, display the Properties dialog for the selected element.	<b>Ctrl+Shift+G</b>	<a href="#">Properties dialog</a> [956]
<b>Feature and Compartment Visibility</b>	Open the Feature and Compartment Visibility dialog to define which features and feature properties to display on the element, and which element compartments.	<b>Ctrl+Shift+Y</b>	<a href="#">Feature Visibility</a> [845]
<b>Rule Composer</b>	For a Rule Task element, invoke the Rule Composer in Business Rule Modeling.		<a href="#">Compose Business Rules</a> [1833]
<b>Create (or Edit) Linked Document</b>	(Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions) Create a document linked to the element.	<b>Ctrl+Alt+D</b>	<a href="#">Create Linked Document on an Element</a> [1047]
<b>Delete Linked Document</b>	Delete an existing linked document for the element.		<a href="#">Replace or Delete Documents</a> [1097]
<b>Create Document to Artifact</b>	(On an Artifact element linked to a Master Document or Model Document element). Generate a virtual document as the linked document of the Artifact element.		<a href="#">Generate Report to an Artifact Element</a> [2680]

Option	Action	Shortcut	See also
<b>New Element</b>	<p>Attach:</p> <ul style="list-style-type: none"> <li>One of a number of types of structural element, such as a Port, Part or Activity Parameter, to the selected element, or</li> <li>A text element to the selected element, documenting a Note, Constraint, Rationale or Problem.</li> </ul>		<a href="#">Create Notes and Text</a> <sup>[923]</sup> <a href="#">Note</a> <sup>[1324]</sup>
<b>New Diagram</b>	<p>Display a submenu, listing - in order:</p> <ul style="list-style-type: none"> <li>Individual types of child diagram (often technology-specific) that you might <b>usually</b> create for this type of element</li> <li>For classifier elements, options to create Activity, Interaction, State Machine or Rule Flow elements that encompass the appropriate type of child diagram</li> <li>Options for creating and displaying the child diagram for a <b>Composite</b> element</li> <li>An option to display the New Diagram dialog to add any <b>other</b> type of diagram as the child diagram of this element</li> </ul>		<a href="#">Classifiers and Instances</a> <sup>[1009]</sup> <a href="#">Composite Elements</a> <sup>[936]</sup> <a href="#">Show Composite Diagram</a> <sup>[1608]</sup> <a href="#">Add New Diagrams</a> <sup>[822]</sup>
<b>Features and Properties</b>	<p>Select to add Attributes and/or Operations to the element, through the Attribute Properties dialog or Operation Properties dialog.</p> <p>For elements that can have interfaces, you can also select to show realized and/or dependent interfaces.</p>		<a href="#">Attributes</a> <sup>[999]</sup> <a href="#">Operations</a> <sup>[1014]</sup> <a href="#">Show Realized interfaces of Class</a> <sup>[865]</sup>
<b>Structural Elements</b>	Display the Structural Elements dialog, for <b>managing</b> the structural elements (such as Provided Interfaces, Ports and Activity Parameters) attached to the selected element.	<b>Ctrl+Shift+B</b>	<a href="#">Manage Structural Elements</a> <sup>[935]</sup>
<b>Find</b>	Display the <b>Find</b> submenu, which provides options for locating the selected element in other diagrams and in the Project Browser, and for setting up cross references (Custom References) to other elements.		<a href="#">Finding Elements</a> <sup>[944]</sup>
<b>Insert Related Elements</b>	Displays the Insert Related Elements dialog, to add elements related to the selected element to the current diagram.		<a href="#">Insert Related Elements</a> <sup>[933]</sup>
<b>Advanced</b>	Open the Advanced sub-menu, which provides options for performing more complex or specialized operations on the element.		<a href="#">Advanced Operations on Elements</a> <sup>[944]</sup>

Option	Action	Shortcut	See also
<b>Transform</b>	Transform the selected element from one domain to another.	<b>Ctrl+H</b>	<a href="#">Transform Elements</a> <sup>[2017]</sup>
<b>Generate DDL</b>	Generate DDL for a Table, Procedure or View Class.		<a href="#">Generate DDL</a> <sup>[2380]</sup>
<b>Generate Code</b>	Generate source code for the selected element (forward engineer).	<b>F11</b>	<a href="#">Generate a Single Class</a> <sup>[2113]</sup>
<b>Synchronize with Code</b>	Reverse engineer source code for the selected element.  You can also synchronize Class elements from the source code via the Project Browser, and reverse engineer or forward engineer model packages and code to synchronize them and incorporate changes made in one but not in the other.	<b>F7</b>	<a href="#">Importing Source Code</a> <sup>[2136]</sup>  <a href="#">Element Options in the Project Browser</a> <sup>[660]</sup>  <a href="#">Operation Menu - Project Browser</a> <sup>[667]</sup>  <a href="#">Update Package Contents</a> <sup>[2117]</sup>
<b>View Source Code</b>	Open the source editor if a source code file exists for that selected element.	<b>F12</b>	<a href="#">Editing Source Code</a> <sup>[2146]</sup>
<b>Execution Analyzer</b>	(If the selected Class element has generated code.) Display the Execution Analyzer submenu, to prepare the Class for Execution Analysis by: <ul style="list-style-type: none"> <li>• Creating workbench variables from the selected Class (<b>Ctrl+Shift+J</b>)</li> <li>• Marking up a process for the Class or</li> <li>• Marking up the Class for recording</li> </ul>		<a href="#">Create Workbench Variables</a> <sup>[2568]</sup>  <a href="#">Recording Activity For a Class</a> <sup>[2542]</sup>
<b>Selectable</b>	Toggle whether the element is selectable or not <ul style="list-style-type: none"> <li>• If an element is selectable, you can move it around the diagram and perform right-click context menu operations</li> <li>• If an element is unselectable, you cannot move it around the diagram and the only right-click operation available is to make the element selectable</li> </ul> <p>This option has no effect on double-click operations on the element, such as displaying child diagrams or Properties dialogs.</p> <p>An element on a locked diagram is also unselectable - if you click on it, the element outline displays in red.</p>		<a href="#">Diagram Filters</a> <sup>[718]</sup>

Option	Action	Shortcut	See also
	In a diagram, you can filter the display to show selectable or non-selectable elements only.		
<b>Dockable</b>	<p>Align and join two elements either vertically or horizontally, on the current diagram only.</p> <p>Both elements must have the <b>Dockable</b> option selected, and must have the joining edges parallel.</p> <p>As the distance between the elements narrows, the moving element snaps to the edge of the other element.</p> <p>For Activity Partitions, the option is selected by default.</p> <p>Deselecting the <b>Dockable</b> option does not separate the elements; if necessary, you can simply move the elements apart again.</p>		<a href="#">Partition</a> <sup>1325</sup>
<b>Appearance</b>	Display a short menu of options to change and reproduce the appearance of the element.		<a href="#">Changing Element Appearance</a> <sup>948</sup>
<b>Z-Order</b>	<p>Display a short submenu providing options to:</p> <ul style="list-style-type: none"> <li>• Move the element further back in the diagram</li> <li>• Move the element further forward in the diagram</li> <li>• Put the element at the back of the diagram</li> <li>• Put the element at the front of the diagram</li> </ul>		<a href="#">Z Order Elements</a> <sup>844</sup>
<b>UML Help</b>	Display the Help topic describing the type of element you have selected.		
<b>Delete &lt;element name&gt;</b>	Delete the element from the diagram. The element still exists within the model and can be accessed from the Project Browser.	<b>Ctrl+D</b>	

### Notes

- Context menus vary between element types; for example, the **Generate Code**, **Synchronize with Code** and **View Source Code** options won't display for a Use Case element
- If you select two or more elements at the same time, you can perform an operation on all of the selected elements at once; the operations available for multiple selection are provided on a separate context menu

### Learn more

- [Execution Analyzer](#) <sup>[2527]</sup>
- [Operations on Multiple Elements](#) <sup>[957]</sup>

### 5.6.2.1 Finding Elements

When you have selected an element on a diagram, you might want to find out where it is held in the model structure, where else it is used in the model, and how to locate it easily in the future. For these operations, you can select the **Find** submenu.

**Access**    **Right-click on element | Find**

#### Find Options

Option	Action	Shortcut	See also
<b>Locate in State Table</b> <b>Locate in State Chart</b>	For a State element, switch to the alternative diagram display format and highlight the element.		
<b>In Project Browser</b>	Highlight the currently selected element in the Project Browser.	<b>Alt+G</b>	
<b>Locate Classifier In Project Browser</b>	For an <b>instance</b> or <b>Object</b> , highlight the <b>classifier</b> for that element in the Project Browser.	<b>Ctrl+Alt+G</b>	
<b>Locate Operation in Project Browser</b>	For an <b>Action</b> element, highlight the <b>call operation</b> for that element in the Project Browser.	<b>Ctrl+Alt+G</b>	<a href="#">Class Operations in Diagrams</a> <sup>[1275]</sup>
<b>Find in all Diagrams</b>	List the other diagrams in which the element is also used, on the Element Usage dialog.	<b>Ctrl+U</b>	<a href="#">Show Element Use</a> <sup>[910]</sup>
<b>Custom References</b>	Identify and, if necessary, open the Properties dialog for, any elements that are the target of a cross reference from the selected element.	<b>Ctrl+J</b>	<a href="#">Set Up Cross References</a> <sup>[917]</sup>
<b>Add to Favorites</b>	Add the selected element to the <i>Favorites</i> folder in the Resources window.		<a href="#">Favorites</a> <sup>[1176]</sup>

### 5.6.2.2 Advanced Operations on Elements












You can perform a wide range of more specialized functions on an element, specific to the element type and relationship to other modeling objects.






**Access**    **Right-click on element | Advanced**



**Advanced Options**

Option	Action	Shortcut	See also
<b>Parent</b>	(For a classifier element that can be generalized) Set the element parent.	<b>Ctrl+I</b>	<a href="#">Set Element Parent</a> <sup>[908]</sup>
<b>Instance Classifier</b>	(For an instance or Object element) Set the instance classifier for the element, on the Select <Item> dialog.	<b>Ctrl+L</b>	<a href="#">Select &lt;Item&gt; Dialog</a> <sup>[994]</sup>
<b>Classifier Properties</b>	(For an instance or Object element) Open the Properties dialog for the <i>classifier</i> of the selected element.	<b>Ctrl+Alt+Enter</b>	<a href="#">Properties Dialog</a> <sup>[956]</sup>
<b>Change to State (Value) Lifeline</b>	(For a State Lifeline or Value Lifeline element) Switch one type of Lifeline element to the other.		<a href="#">Value Lifeline</a> <sup>[1355]</sup> <a href="#">State Lifeline</a> <sup>[1335]</sup>
<b>Set Property Type</b>	(For a Part element) Browse for and select the element that defines the Part type, using the Select <Item> dialog.		<a href="#">Part</a> <sup>[1383]</sup>
<b>Multiplicity</b>	Define the multiplicity for the element, using the format defined on the Cardinality Values tab of the UML Types dialog ( <b>Settings   UML Types</b> ).  This is the number of instances of the element that can exist in a set; the value displays on the element in a diagram, in the <i>Name</i> compartment.		<a href="#">Cardinality</a> <sup>[1151]</sup>
<b>Port Size Customizable</b>	(For a Port element) Allow the Port to be re-sized from the default size. If the option is not set, the element size cannot be changed.		<a href="#">Port</a> <sup>[1384]</sup>
<b>Show Element Type</b>	(For a Port or Part element) Toggles between displaying and hiding the element type, as set using the <b>Set Property Type</b> option (above) or on the Property page of the element's Properties dialog.		<a href="#">Part</a> <sup>[1383]</sup> <a href="#">Port</a> <sup>[1384]</sup> <a href="#">The Property Page</a> <sup>[1386]</sup>
<b>Set Property Values</b>	(For a Part element) Set the property value for the Part, using the Set Property Values dialog.	<b>Ctrl+Shift+R</b>	<a href="#">Add Property Values</a> <sup>[1383]</sup>
<b>Edit Extension Points</b>	(For a Use Case element) Display the Use Case Extension Points dialog, which you use to insert the point at which the behavior should be inserted.		<a href="#">Use Case Extension Points</a> <sup>[1354]</sup>

Option	Action	Shortcut	See also
<b>Association Class</b>	(For a Class element) Connect the Class to a new Association.		<a href="#">Connect New Class to Association</a> 
<b>Use Rectangle (Circle, Actor) Notation</b>	(For an element that uses non-rectangular notation) Apply rectangle notation to the element (and switch back again).		<a href="#">Rectangle Notation</a> 
<b>Partition Activity</b>	(For an Activity element) Define an Activity Partition in the element.		<a href="#">Activity Partition</a> 
<b>Set Run State</b>	(For an Object or State element) Add a new instance variable to the element using the Define Run State dialog.	<b>Ctrl+Shift+R</b>	<a href="#">Define A Run-Time Variable</a> 
<b>Set Object State</b>	(For an Object or instance element) Set the state of the element based on the child states of its classifier.	<b>Ctrl+Shift+S</b>	<a href="#">Object State</a> 
<b>Override Attribute Initializers</b>	(For a Class element with attributes) Pre-define initial values for attributes that can be used to override existing defaults.	<b>Ctrl+Shift+R</b>	<a href="#">Display Inherited Attributes</a> 
<b>Convert to Instance (Property/Invocation)</b>	Convert this classifier to an instance, property or invocation, depending on the type of classifier selected (for example, SysML classifiers are always converted to properties).		<a href="#">Classifiers and Instances</a> 
<b>Convert Linked Element To Local Copy</b>	Convert the occurrence of the element on this diagram from a <b>link to the original element</b> to a <b>local copy</b> of the element. A copy of the element is also added to the diagram's parent package in the Project Browser.		
<b>Make Sender/Receiver</b>	(For Send and Receive Event elements) Toggle the element from a sender to a receiver and vice versa.		<a href="#">Receive</a>  <a href="#">Send</a> 
<b>Accept Time Event</b>	(For a Receive element) Change the notation from an Accept Event action to an Accept Time Event action.		<a href="#">Receive</a> 
<b>Define Concurrent Substates</b>	(For a State element) Define a set of substates that can be held simultaneously within that composite State.		<a href="#">Regions</a> 

Option	Action	Shortcut	See also
<b>Use State Label Notation</b>	Display State Label Notation for a State object (the element name is displayed on a box on top of the element rather than inside it).		
<b>Deep History</b>	(For a History pseudo-State) Change the type of a <b>shallow</b> History pseudo-state to a <b>deep</b> History pseudo-state.		<a href="#">History</a> 
<b>Set Attached Links</b>	(For a Note element) Attach the element to a connector, or several connectors.		<a href="#">Add a Note to a Connector</a> 
<b>Link to Diagram Note</b>	(For a Note element) Display the <b>diagram</b> notes as the text of the Note element.  The option simply deletes any current text and blocks the Note from being edited other than through the <b>Notes</b> field in the diagram Properties dialog.		<a href="#">Link Note to Internal Documentation</a> 
<b>Link to Connector Tagged Value</b>	(For a Note element) Display the Tagged Values assigned to the connector as the Note text, or hide the text.		
<b>Vertical Partition</b>	(For an Activity Partition element) Switch a horizontal Partition to a vertical Partition.		<a href="#">Partition</a> 
<b>Tagged Value</b>	Add a Tagged Value to the selected element.		<a href="#">Quick Start - Add Tagged Value To Elements</a> 

### Notes

- Context menus vary between element types, and few of the menu options shown here are present on all element context menus; **Tagged Value** is often the only option available
- If an element appearance is overridden by a Shape Script, several of the appearance options are disabled; for example, **Use Rectangle (Circle) Notation**

### Learn more

- [Shape Scripts](#) 

### 5.6.2.3 Changing Element Appearance

If you want to modify the appearance of a specific element on a diagram, you can use a range of facilities available through the element context menu on the diagram.

**Access**    **Right-click on element on diagram | Appearance**

#### Element Appearance Options

Option	Action	Shortcut	See also
<b>Align Text</b>	(Hyperlink elements) Select an option to justify the text of the hyperlink on the left, middle or right of the element space.		
<b>Default Appearance</b>	Override the <i>global</i> default appearance of all elements (which you set on the Standard Colors and Diagram Appearance pages of the Options dialog) with a different default for just the selected element on all diagrams in which it is found.  To change the appearance of the selected element on the <i>current diagram only</i> , use the Format toolbar.	<b>F4</b>	<a href="#">Standard Colors</a> <sup>[616]</sup> <a href="#">Diagram Appearance Options</a> <sup>[622]</sup> <a href="#">Set an Element's Default Appearance</a> <sup>[927]</sup> <a href="#">Format Toolbar</a> <sup>[785]</sup>
<b>Apply Image From Clipboard</b>	Paste the image held on the <b>clipboard</b> onto the selected element, as an alternative image.  This image is added to the <b>Image Library</b> under an automatically-assigned name. The image can be accessed from and renamed in the Image Manager.		<a href="#">Using the Image Manager</a> <sup>[860]</sup>
<b>Select Alternate Image</b>	Select an alternative image using the <b>Image Manager</b> .	<b>Ctrl+Shift+W</b>	<a href="#">Using the Image Manager</a> <sup>[860]</sup>
<b>Hide/Show Name Under Image</b>	Hide or redisplay the name label under an element with an alternative image.		
<b>Set Font</b>	Change the font type, size, color and effects for the text in an element.		<a href="#">Set Element Font</a> <sup>[949]</sup>
<b>Diagram Frame Appearance</b>	(Diagram Frame elements) Select from a number of options to set the appearance of a Diagram Frame dragged onto the diagram from the Project Browser.		<a href="#">Diagram Frame</a> <sup>[1296]</sup>

Option	Action	Shortcut	See also
<b>Show Labels</b>	Reveal any hidden labels on the element.		
<b>Copy Appearance to Painter</b>	Copy the default element appearance (set using the <b>Default Appearance</b> option, above) to the painter.  You then paste the default appearance using the <b>Paste Appearance</b> option on the Diagram toolbar.		<a href="#">Diagram Toolbar</a> [138]
<b>Copy Image to Clipboard</b>	Copy the element image to the clipboard. You can use the <b>Apply Image from Clipboard</b> option (above) to recover and use the image elsewhere.		

**Notes**

- You can also change the appearance (and other aspects) of several selected elements at once
- Context menus vary slightly between element types, and not all menu options shown here are present on all element context menus; for example, the **Alternate Image** option won't display for a Lifeline element

**Learn more**

- [Operations on Multiple Elements](#) [95]

**5.6.2.3.1 Set Element Font**

When you display an element on a diagram, the text font defaults to a global setting defined from the Diagram Appearance page of the Options dialog. You can change this global default setting if you need to. On the diagram, however, you might want to change the font only on a specific element or group of elements, to distinguish them from other elements. You have several options for doing this.

**Access** **Right-click on element | Appearance | Set Font**  
**Click on element | Format Toolbar | Font**  
**Click on element | F4 | Font Style [ ... ]**

**Font Dialog Options**

Field/Button	Action	See also
<b>Font</b>	Scroll through the list of font types and click on the name of the appropriate font, which is then displayed in the field at the top of the list.  This setting affects all text in the selected element except for the Note compartment text.	
<b>Font Style</b>	In the list of font styles click on the name of the appropriate style, which is then displayed in the field at the top of the list.	

Field/Button	Action	See also
	This setting affects all text in the selected element except for the Note compartment text.	
<b>Size</b>	<p>Scroll through the list of font sizes and click on the appropriate size, which is displayed in the field at the top of the list.</p> <p>Alternatively, type the font size directly into the field.</p> <p>This setting affects all text in the selected element.</p>	
<b>Strikeout</b>	Select this checkbox to show the element text with a line through it. (This checkbox is not currently available.)	
<b>Underline</b>	Select this checkbox to show the element text with an underline.	
<b>Color</b>	<p>Click on the drop-down arrow and click on a color in the list.</p> <p>(If you access the Font dialog through the Default Appearance dialog, the color you select will be overridden by the color in the <b>Font Color</b> field of the Default Appearance dialog, if different.)</p>	
<b>Script</b>	If Western script is not appropriate for the element text, click on the drop down arrow and select a different script type.	
<b>Sample</b>	Review the combined effects of the settings you have selected.	
<b>OK</b>	Click on this button to save your changes.	

### Notes

- If you set the element font through the Default Appearance dialog (**F4**) the changes can be overridden by any different values set through the other two access paths
- When it has been called directly on one or more selected elements (**Appearance | Set Font**), the Font dialog overrides the Default Appearance dialog

### Learn more

- [Set Default Fonts](#) <sup>[624]</sup>
- [Format Toolbar](#) <sup>[785]</sup>
- [Set an Element's Default Appearance](#) <sup>[927]</sup>

### 5.6.2.4 Operations on Multiple Elements

It is possible to perform various operations on two or more elements on a diagram at once, either to add the same feature or to standardize the size or orientation of the selected elements. This makes it much faster to update the diagram than if you had to modify each element separately.

To select the required elements, either click and drag the cursor over the group to highlight them, or press **Shift** and click on each element separately.

**Access** Right-click on one of the selected elements (for format/layout operations, right-click on the element to standardize the others against)

#### Options

Option	Detail	Shortcut	See also
<b>Align</b>	<p>Align elements against the 'master' element's:</p> <ul style="list-style-type: none"> <li>• left edge</li> <li>• right edge</li> <li>• top edge</li> <li>• bottom edge</li> <li>• center, vertically or</li> <li>• center, horizontally</li> </ul> <p>You can also edge-align groups of elements using the Diagram Toolbar; the four alignment buttons are made available when more than one element is selected in a diagram.</p>	<b>Ctrl+Alt+←</b> <b>Ctrl+Alt+→</b> <b>Ctrl+Alt+↑</b> <b>Ctrl+Alt+↓</b>	<a href="#">Operations on Elements</a> <sup>[939]</sup>  <a href="#">Diagram Toolbar</a> <sup>[138]</sup>
<b>Space Evenly</b>	(For three or more selected elements.) Automatically even out the spacing between the selected elements, either horizontally or vertically.	<b>Alt+-</b> <b>Alt+=</b>	
<b>Height/Width</b>	<p>Make all the selected elements match the 'master' element's height or width, or both.</p> <p>In Use Case diagrams, these options are blocked if the <b>Allow Elongated Use Cases</b> option is turned off on the Options dialog Objects page.</p>		<a href="#">Object Display Options</a> <sup>[637]</sup>
<b>Feature and Compartment Visibility</b>	Define the visibility of features and compartments on all selected elements.	<b>Ctrl+Shift+Y</b>	<a href="#">Feature Visibility</a> <sup>[845]</sup>
<b>Add Tagged Value</b>	Add the same Tagged Value to all selected elements.		<a href="#">Assign a Tagged Value to an Item</a> <sup>[1137]</sup>

Option	Detail	Shortcut	See also
<b>Autosize</b>	Automatically resize the selected elements to their default minimum size (element content permitting).	<b>Alt+Z</b>	<a href="#">Autosize Elements</a> <sup>[85]</sup>
<b>Appearance</b>	Set the same default appearance and font for all of the selected elements at once.	<b>F4</b>	<a href="#">Set an Element's Default Appearance</a> <sup>[92]</sup> <a href="#">Set Element Font</a> <sup>[94]</sup>
<b>Make Non-Selectable</b>	Make the selected elements on the diagram <b>non-selectable</b> .  To make the elements selectable again, either: <ul style="list-style-type: none"> <li>Right-click on <b>individual elements</b> and click on the <b>Selectable</b> menu option, or</li> <li>Right-click on the <b>diagram</b> and select the <b>Make All Elements Selectable</b> context menu option</li> </ul>		<a href="#">Operations on Elements</a> <sup>[94]</sup>
<b>Dockable</b>	Turn the <b>Docking</b> feature on or off for the selected elements. When two elements are dockable, they can be moved together to automatically join along their common edge.		<a href="#">Operations on Elements</a> <sup>[94]</sup>
<b>Code Generation</b>	Either: <ul style="list-style-type: none"> <li>Generate code for the selected elements together, or</li> <li>Synchronize code and model for each of the selected elements</li> </ul>	<b>Shift+F11</b> <b>Ctrl+R</b>	<a href="#">Generate a Group of Classes</a> <sup>[211]</sup>
<b>Transform</b>	Transform the selected elements using a transformation template in a single operation.	<b>Ctrl+H</b>	<a href="#">Transform Elements</a> <sup>[201]</sup>
<b>Copy</b>	Copy the selected elements to the clipboard.		
<b>Layout Selected Elements</b>	Automatically adjust the layout of the selected elements on the diagram.		<a href="#">Layout a Diagram Automatically</a> <sup>[89]</sup>
<b>Delete Selected Elements</b>	Delete the selected elements from the <b>diagram</b> only.	<b>Ctrl+D</b>	



#### 5.6.2.4.1 Position Elements

Use these sections of the Multiple Selection menu to size and position elements on the diagram, relative to each other.

##### Size

Use these options to make the selected element(s) wider, narrower, taller or shorter by one increment.

Option	Action	Shortcut	See also
<b>Wider</b>	Make all selected elements wider.	<b>Ctrl+→</b>	
<b>Narrower</b>	Make all selected elements narrower.	<b>Ctrl+←</b>	
<b>Shorter</b>	Make all selected elements shorter.	<b>Ctrl+↑</b>	
<b>Taller</b>	Make all selected elements taller.	<b>Ctrl+↓</b>	

##### Move

Use these options to move the selected element(s) left, right, up or down by one increment.

Option	Action	Shortcut	See also
<b>Right</b>	Move all selected elements to the right.	<b>Shift+→</b>	
<b>Left</b>	Move all selected elements to the left.	<b>Shift+←</b>	
<b>Up</b>	Move all selected elements upwards.	<b>Shift+↑</b>	
<b>Down</b>	Move all selected elements downwards.	<b>Shift+↓</b>	

### 5.6.3 Visual Representation

Each UML element has a default representation. However, you can:

- Change the appearance, features, position and - for some elements - orientation through a number of toolbars
- Display or hide various types of information held in the element, in compartments

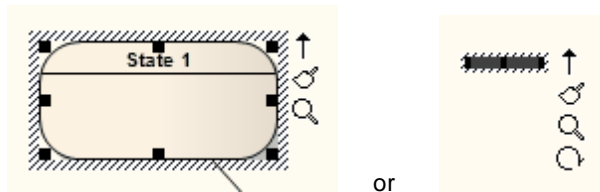
##### Learn more

- [Element Icons](#) <sup>954</sup>

- [Compartment](#)<sup>[955]</sup>

### 5.6.3.1 Element Icons

When you add an element to a diagram, or select an existing element, a number of small icons display off the right hand side of the element underneath the Quicklinker arrow.




These icons display small versions of the diagram toolbars or perform specific actions, to enable you to quickly edit the element you have highlighted.

You can prevent the display of these icons by deselecting the **Show buttons for selected Object on diagram** checkbox on the Objects page of the Options dialog.

#### Icon explanations

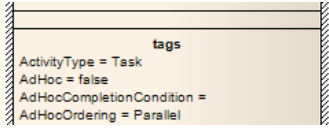
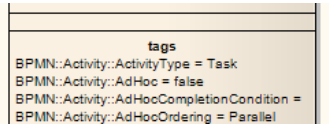
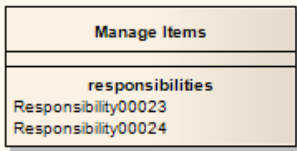
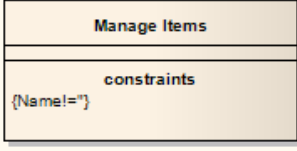
Icon	Description	See also
	Rotates a Fork/Join element from vertical to horizontal and vice-versa.	
	Displays the Format toolbar, for changing element appearance.	<a href="#">Format Toolbar</a> <sup>[785]</sup>
	Displays the Current Element toolbar, to edit the element's properties and features.	<a href="#">Current Element Toolbar</a> <sup>[141]</sup>
	When multiple elements are selected, displays the Diagram Toolbar for changing or aligning the elements together.	<a href="#">Diagram Toolbar</a> <sup>[138]</sup>
	Toggles an Activity Partition between vertical and horizontal.	<a href="#">Activity Partition</a> <sup>[1283]</sup>
	<p>Indicates that the Interaction Fragment currently contains messages; in this state you can:</p> <ul style="list-style-type: none"> <li>• Move the fragment vertically, but not horizontally; all elements and messages <b>below</b> the fragment move up or down by the same amount</li> <li>• Resize the fragment larger than the contained messages/elements, but not smaller</li> </ul> <p>This is the default condition when you open or reload a diagram containing a fragment. You can click on the icon to toggle to <b>'move freely'</b> mode (below).</p>	<a href="#">Combined Fragment</a> <sup>[1267]</sup>

Icon	Description	See also
	<p>Indicates that the Interaction Fragment can be moved independently of the Messages or other elements contained within the fragment.</p> <p>This is the default condition when you first create the fragment. You can click on the icon to toggle to '<b>contains messages</b>' mode (above).</p>	

### 5.6.3.2 Compartments

When you place an element on a diagram, you can display it with a number of compartments that reveal, not only the attributes and operations of the element, but also several other properties such as tags, constraints and tests, as described here.

#### Compartment Types

Images	Type	Detail	See also
 <ul style="list-style-type: none"> <li>Or, fully qualified, expanded format:</li> </ul> 	<b>Tags Compartment</b>	<p>The <b>tags</b> compartment lists all Tagged Values for an element as entered in the Tagged Values window.</p> <p>The <b>fully-qualified</b> option operates only on those Tagged Values that were created in Enterprise Architect release 7.1 or later; it does not expand Tagged Values created in earlier releases.</p>	<a href="#">Tagged Values</a> <sup>[1134]</sup>
	<b>Responsibilities Compartment</b>	The <b>responsibilities</b> compartment shows a list of responsibilities as entered on the Requirements page of the element Properties dialog.	<a href="#">Requirements</a> <sup>[961]</sup>
	<b>Constraints Compartment</b>	The <b>constraints</b> compartment shows a list of element constraints as entered in the Constraints page of the element Properties dialog.	<a href="#">Constraints</a> <sup>[963]</sup>
	<b>Testing Compartment</b>	The <b>testing</b> compartment lists all of the tests associated with an element as listed in the Testing window.	<a href="#">Working on Test Records</a> <sup>[2605]</sup>

Images	Type	Detail	See also
	<b>Maintenance Compartment</b>	The <b>maintenance</b> compartment lists all of the defects, changes, issues and tasks associated with an element, as listed in the Maintenance window.	<a href="#">Working on Maintenance Items</a> <sup>[2623]</sup>

#### Notes

- To set the visibility of the various compartments, see the *Feature Visibility* topic

#### Learn more

- [Feature Visibility](#) <sup>[845]</sup>

### 5.6.4 Element Property Displays

To define, assign and review the properties and features of elements, you can use the following:

Topic	Link
Properties Dialog	<a href="#">Properties Dialog</a> <sup>[956]</sup>
Properties Docked Window	<a href="#">Properties Window</a> <sup>[992]</sup>
Element Browser	<a href="#">The Element Browser</a> <sup>[989]</sup>
Scenarios & Requirements Window	<a href="#">The Scenarios &amp; Requirements Window</a> <sup>[992]</sup>
Select <Item> Dialog	<a href="#">Select &lt;Item&gt; Dialog</a> <sup>[994]</sup>

#### 5.6.4.1 Properties Dialog

This topic area covers element properties and their settings, responsibilities, constraints, connectors, scenarios, Tagged Values, associated files, object files and classifiers, and boundary element settings.

#### Topics

Topic	Detail	See also
<b>Displaying the</b>	To display the element Properties dialog, use any of the following methods:	

Topic	Detail	See also
<b>Properties dialog</b>	<ul style="list-style-type: none"> <li>Select an element in the Diagram View and select the <b>Element   Properties</b> menu option</li> <li>Right-click on an element in the Diagram View, and select the <b>Properties</b> context menu option</li> <li>Select an element in the Diagram View, and press ( <b>Alt+Enter</b> )</li> <li>Double-click on an element in the Diagram View</li> <li>Right-click on an element in the Project Browser, and select the <b>Properties</b> context menu option</li> </ul> <p>To suppress display of the Properties dialogue when placing a new element, uncheck the <b>Edit Object on New</b> option on the Objects page of the Options dialog.</p>	<a href="#">Object Display Options</a> <sup>[631]</sup>
<b>Properties dialog pages</b>	<p>The Class Properties dialog consists of the following pages:</p> <ul style="list-style-type: none"> <li>General</li> <li>Details</li> <li>Advanced Properties</li> <li>Templates</li> <li>Tagged Values</li> <li>Requirements</li> <li>Constraints</li> <li>Scenarios</li> <li>Files</li> <li>Links</li> </ul> <p>Properties also include Object files and classifiers, and the System Boundary element appearance.</p>	<a href="#">General Settings</a> <sup>[958]</sup> <a href="#">Details</a> <sup>[959]</sup> <a href="#">Advanced Properties</a> <sup>[961]</sup> <a href="#">Templates</a> <sup>[961]</sup> <a href="#">Tagged Values</a> <sup>[1134]</sup> <a href="#">Requirements</a> <sup>[961]</sup> <a href="#">Constraints</a> <sup>[963]</sup> <a href="#">Scenarios</a> <sup>[965]</sup> <a href="#">Associated Files</a> <sup>[988]</sup> <a href="#">Links</a> <sup>[964]</sup> <a href="#">Classifiers and Instances</a> <sup>[1009]</sup> <a href="#">System Boundary Properties</a> <sup>[1348]</sup>

### Notes

- There are several variations of the Properties dialog:
  - The dialog for a Table or Stored Procedure element has slight differences on the General page, and a Table (Stored Procedure) Details page instead of a Details page
  - The dialog for a Class element of a stereotype other than Table is as shown in General Settings
  - The dialog for an element of any other type does not have a Details page
  - Port and Part elements have a Property page
  - Activity elements have a Behavior page, and Action and Invocation elements (depending on their type) have other pages such as Effect, Trigger and Call pages
  - Action Pins have a Pin page
- In all cases, the Properties dialog is an expandable window, which you can stretch to enable longer entry and clearer inspection of the text field values
- The Tagged Values page of the element Properties dialog simply provides the Tagged Values window

within the frame of the Properties dialog

- If Tagged Values have been derived from a stereotyped element in a Profile, they are listed on a separate **page** having the Profile name; they are still listed on the Tagged Values window, in a separate **compartment**



#### Learn more

- [Working With Table Properties](#) <sup>[2340]</sup>
- [Port](#) <sup>[1384]</sup>
- [Part](#) <sup>[1383]</sup>
- [The Property Page](#) <sup>[1386]</sup>
- [Interactions and Activities](#) <sup>[1028]</sup>
- [Action](#) <sup>[1266]</sup>
- [Trigger](#) <sup>[1350]</sup>
- [Behaviour Calls](#) <sup>[1026]</sup>
- [Action Pin](#) <sup>[1277]</sup>

#### 5.6.4.1.1 General Settings

The General page of the element Properties dialog contains the following fields:

Field	Usage	See also
<b>Name</b>	Change the element's name.	
<b>Stereotype</b>	(Optional) Type the name of a stereotype for the element, or click on the drop-down arrow and select one.	
<b>Role</b>	Type the role played by the object represented by the element.	
<b>Alias</b>	Type an alias (alternative display name) for the object.	
<b>Keywords</b>	Enter free-text items such as keywords or context information; this can be filtered in Use Case Metrics and Search dialogs.	
<b>Author</b>	Enter or select the name of the original author.	
<b>Language</b>	Select the programming language for the object.	
<b>Status</b>	Indicate the current status of the element (such as Approved or Proposed).	
<b>Complexity</b>	Indicate the complexity of the element (used for project estimation). You normally select from three levels:	<a href="#">Object Display Options</a> <sup>[631]</sup>

Field	Usage	See also
	<ul style="list-style-type: none"> <li>• <b>Easy</b></li> <li>• <b>Medium</b></li> <li>• <b>Difficult</b></li> </ul> <p>You can change your user defaults to add <b>Extreme</b> and <b>Unknown</b> to this list of options.</p>	
<b>Version</b>	Enter the version of the current element.	
<b>Phase</b>	Indicate the phase this element is to be implemented in (for example, 1, 1.1, 2.0, ...).	
<b>Notes</b>	<p>Enter any notes text associated with the element, as described for the Notes window,</p> <p>You can format the notes text using the Notes toolbar at the top of the field,</p>	<a href="#">Notes</a>  <sup>[142]</sup> <a href="#">Notes Toolbar</a>  <sup>[143]</sup>

Further facilities are made available by clicking on the Advanced page.


#### Learn more

- [Advanced Settings](#)  <sup>[96]</sup>

#### **5.6.4.1.2 Details**

The Details page of the element Properties dialog enables you to define the structural and processing details for the selected Class element.

#### Reference

Field/Button	Usage	See also
<b>Scope</b>	Click on the drop-down arrow and select the element's scope (public, private, protected, package).	
<b>Persistence</b>	Click on the drop-down arrow and select the appropriate value (blank, persistent or transient).	
<b>Cardinality</b>	<p>The number of instances of the element that can exist - the value displays on the element in a diagram, in the <i>Name</i> compartment.</p> <p>Use the format defined in the Cardinality tab.</p>	<a href="#">Cardinality</a>  <sup>[15]</sup>

Field/Button	Usage	See also
<b>Abstract</b>	Select if the element is abstract.	
<b>Is Root</b>	Select if the element is a root element and cannot be descended from another.	
<b>Is Leaf</b>	Select if the element is final and cannot be a parent for other elements.	
<b>Is Specification</b>	Select if the element is a specification.	
<b>Is Active</b>	Select if the element is active; for example, an Active Class.	<a href="#">Active Classes</a> <sup>[1365]</sup>
<b>Attributes</b>	Define attributes for the Class. The Attributes Properties dialog displays.	<a href="#">Attributes</a> <sup>[999]</sup>
<b>Operations</b>	Define operations for the Class. The Operations Properties dialog displays.	<a href="#">General Properties of Operations</a> <sup>[1015]</sup>
<b>Concurrency</b>	Specify how concurrent activities should be processed.	
<b>Collection Classes</b>	Define Collection Classes (for generating code from Association connectors) that apply to this Class. The Collection Classes for Association Roles dialog displays.	<a href="#">Set Collection Classes</a> <sup>[2259]</sup>
<b>Template Parameters</b>	List the <i>Parameterized Class</i> template parameters. Click on the <b>Edit</b> button to edit a selected parameter, or click on the <b>Add</b> button to add a new parameter.	<a href="#">Parameterized Classes</a> <sup>[1365]</sup>
<b>Binding(s)</b>	List the binding expressions on a <i>binding</i> Class. Click on the <b>Edit</b> button to edit a selected binding expression, or click on the <b>Add</b> button to add a new binding expression.	<a href="#">Template Binding</a> <sup>[1444]</sup>

#### Notes

- When launched from MDG Integration, the **Attributes** and **Operations** buttons are not available



#### 5.6.4.1.3 Advanced Properties

The Advanced page displays the advanced properties of the element (where they exist) and enables you to reset the values of these properties.

##### How to

To set a value for an advanced (or custom) property

Step	Action	See also
1	Click on the data field to the right of the property name.	
2	Depending on the property, either: <ul style="list-style-type: none"><li>• Type the value in free text</li><li>• Click on the drop-down arrow and select the value from the list</li><li>• Click on the ( ... ) Browse button and search for the required value</li></ul>	
3	Click on the <b>Apply</b> button.	

#### 5.6.4.1.4 Templates

Use the Templates page to define

- Parameterized Class template parameters
- Binding expressions on a binding Class

In the Template Parameter(s) panel, click on the **Add** button to add a new parameter, or click on the **Edit** button to edit a selected existing parameter; in either case, the Template Parameter dialog displays.

In the Binding(s) panel, click on the **Add** button to add a new binding expression or click on the **Edit** button to edit a selected binding expression. When you click on the **Add** button, a short context menu displays from which you select the type of relationship to generate, and then you define the binding expressions.

If either a template parameter or a binding expression is no longer required, click on it and then click on the corresponding **Delete** button.

##### Learn more

- [Parameterized Classes \(Templates\)](#)<sup>[1365]</sup>
- [Template Binding](#)<sup>[1444]</sup>

#### 5.6.4.1.5 Requirements



You use the Requirements page of the element Properties dialog to create requirements that this element is designed to meet. Requirements are of two types:

- Internal requirements (responsibilities) and
- External requirements (system requirements, elements connected to the element by a Realize

connector)





Enterprise Architect shows both types, but you can only edit the internal type from the Requirements page.

You can show the requirements for an element on the diagram directly, using the Feature Visibility function; it is also possible to show inherited requirements in this way.

Field/Button	Usage	See also
<b>Requirement</b>	Enter the name and high level detail of the requirement.	
<b>Type</b>	Specify the type; for example, Functional or Non-functional: <ul style="list-style-type: none"> <li>• <b>Functional</b> requirements are things that the system must do, such as identify franked, unfranked and total credit for a dividend</li> <li>• <b>Non-functional</b> requirements are things that the system must be, such as reliable or cost effective</li> </ul>	
<b>Last update</b>	Specify the date of the last requirement update.	
<b>Status</b>	Specify the current status of the requirement.	
<b>Difficulty</b>	Identify the difficulty of implementing the current requirement.	
<b>Priority</b>	Specify how urgent the requirement is.	
<b>Stability</b>	Specify the estimated stability of the requirement.  This is an indication of the probability of the requirement - or understanding of the requirement - changing; high stability indicates a low probability of the requirement changing.	
<b>Notes</b>	Record details of the requirement.  You can format the notes text using the Notes toolbar at the top of the field.	<a href="#">Notes Toolbar</a> 
<b>Move External</b>	Make an internal responsibility into an external requirement.	<a href="#">Make Internal Requirement External</a> 
<b>New</b>	Create a new requirement.	

Field/Button	Usage	See also
<b>Save</b>	Save changes to requirements.	
<b>Delete</b>	Delete a selected requirement.	
<b>Defined</b>	List the defined requirements associated with this element.	

#### Learn more

- [Internal Requirements](#) 
- [External Requirements](#) 
- [Feature Visibility](#) 
- [The Element Browser](#) 

#### **5.6.4.1.5.1 External Requirements**

External requirements are those Requirement elements that have been connected to the current element using a *Realization* connector. By creating the connector from the element to the requirement, you create an expectation that the element must implement the requirement as part of the system solution.

In Enterprise Architect, linked requirements are shown in the Requirements page of the element Properties dialog, but they are marked external and cannot be directly edited (on selection, the page fields are grayed out).

Double-click an external requirement in the list to activate the Properties dialog for the associated requirement, where you can view and modify the requirement details and check the requirement hierarchy details.

#### Learn more

- [Create Requirements](#) 
- [Requirement Properties](#) 
- [Make Internal Requirement External](#) 

#### **5.6.4.1.6 Constraints**

If you need to define the conditions and rules under which an element operates and exists, you can do this by setting Constraints on the element, in conjunction with responsibilities (internal requirements). Typical constraints are pre- and post- conditions, which indicate things that must be true before the element is created or accessed and things that must be true after the element is destroyed or its action is complete.

You can show the constraints for an element directly on the diagram, using the Feature Visibility function; this also reveals any constraints inherited from parent elements.

**Access**    **Double-click element > Constraints**

#### Define constraints for an element

Field/Button	Description	See also
<b>New</b>	Click on this button to clear the fields ready to create a new constraint.	
<b>Constraint</b>	Type the value of the constraint.	
<b>Type</b>	Click on the drop-down arrow and select the appropriate type ( <b>Pre-condition</b> , <b>Post-condition</b> or <b>Invariant</b> ).	
<b>Status</b>	Click on the drop-down arrow and select the appropriate status.	
Notes	Type any additional information required.	
<b>Save</b>	Click on this button to save the constraint data.	
<b>OK</b>	Click on this button to close the dialog.	

#### Learn more

- [Feature Visibility](#)<sup>[845]</sup>
- [Requirements](#)<sup>[961]</sup>
- [The Element Browser](#)<sup>[989]</sup>

#### 5.6.4.1.7 Links

The Links page of the element Properties dialog displays a list of all relationships active for the current element.

#### Topics

Topic	Detail	See also
<b>Relationships Panel</b>	<p>The Relationships panel lists the relationships this element has, including:</p> <ul style="list-style-type: none"> <li>• Elements this element is related to</li> <li>• Element Stereotype (if any) of the element</li> <li>• Type of the related element</li> <li>• Connection or relationship type</li> <li>• Stereotype (if any) of the relationship</li> </ul>	
<b>Links Page</b>	From the Links page you can perform operations on a relationship, by right-clicking on the relationship to display the context menu.	<a href="#">Connector Properties</a> <sup>[1126]</sup>

Topic	Detail	See also
	<p>To:</p> <ul style="list-style-type: none"> <li>Hide the relationship on the diagram, click on the <b>Hide Relation</b> menu option; the option then changes to <b>Show Relation</b>, which you select to redisplay the relationship on the diagram</li> <li>Display the relationship Properties dialog, click on the <b>Relationship Properties</b> menu option</li> <li>Highlight the related element in the Project Browser, click on the <b>Locate Related Object</b> menu option</li> <li>Delete the relationship from the model and all diagrams, click on the <b>Delete Relationship</b> menu option; the system prompts you to confirm the deletion</li> </ul>	

#### 5.6.4.1.8 Scenarios

A scenario is a real-world sequence of operations that describes how an element works in real-time.



##### Topics

Topic	Detail	See also
<b>Where you can generate Scenarios</b>	<p>You can apply scenarios to any element - generally Use Cases - to describe functional behavior, business work flows and end-to-end business processes.</p> <p>You create scenarios through the Scenario page of the element Properties dialog; this page has two internal tabs, as described below.</p> <ul style="list-style-type: none"> <li>The Description tab enables you to create scenarios and provide a simple text description of each scenario, or of the structure of each scenario</li> <li>The Structured Specification tab (the default) enables you to create scenarios or select those you have created elsewhere and, for each scenario: <ul style="list-style-type: none"> <li>Create a series of steps for each part of the scenario</li> <li>Structure the scenario to show how the basic path diverges into the alternate paths and exception paths</li> <li>Generate a number of types of diagram from the structure</li> <li>Generate a structured scenario from an Activity diagram</li> <li>Generate a structured scenario from text on the clipboard; this option has a variation in the Description tab that enables you to translate scenario descriptions created prior to release 8.0 of Enterprise Architect, into structured scenarios in the latest release</li> </ul> </li> </ul>	<p><a href="#">Structured Specification Tab</a> [967]</p> <p><a href="#">Generate Diagrams</a> [976]</p> <p><a href="#">Generate Scenario from Activity diagram</a> [985]</p> <p><a href="#">Structured Specification Item Context Menu</a> [972]</p> <p><a href="#">Scenarios</a> [966]</p>

Topic	Detail	See also
<b>Description Tab</b>	<p>On the Scenarios page, click on the Description tab.</p> <p>The <b>Scenario</b> (name) field and the <b>Type</b> field both default to <b>Basic Path</b>, to enable you to define the basic path first; you can overtype the scenario name with more appropriate text if required.</p> <p>As you go on to create other scenarios, you set the type to <b>Alternate</b> or <b>Exception</b> as appropriate.</p> <p>Complete the fields as described in the table below.</p>	

### Reference

Field	Usage	See also
<b>Scenario</b>	Type in the name of the scenario (or, for existing scenarios, click on the drop-down arrow and select one from the list).	
<b>Type</b>	<p>Specify the type of scenario; the options are:</p> <ul style="list-style-type: none"> <li>• <b>Basic Path</b> - the direct set of steps for the scenario</li> <li>• <b>Alternate</b> - an alternative set of steps, in parallel with part of the basic path</li> <li>• <b>Exception</b> - the path the scenario follows if a step of the basic path does not produce an appropriate result</li> </ul>	
<b>Description</b>	<p>Record a textual description of how the user uses the current element.</p> <p>As for the Notes window, you can format the notes text using the Notes toolbar at the top of the field.</p> <p>As well as the Notes window facilities, you can also generate a structured specification from the text in this field; highlight the text, right-click on it and select the <b>Create Structure from Notes</b> context menu option.</p> <p>The text is copied to the Structured Specification tab for the current scenario, either as a new specification or as the continuation of an existing specification, with a new step created at each carriage return.</p> <p>Subsequent changes to the text on the Structured Specification tab are not reflected on the Description tab.</p>	<a href="#">Note Tab</a> <sup>[1142]</sup> <a href="#">Notes Toolbar</a> <sup>[1143]</sup> <a href="#">Structured Specification Tab</a> <sup>[967]</sup>
<b>New</b>	Clear the data fields so that you can enter data in them to create a new scenario.	

Field	Usage	See also
<b>Save</b>	Save a new scenario, or changes to an existing scenario.	
<b>Delete</b>	Delete a scenario selected from the Scenarios panel, below.	
<b>Scenarios</b>	Display a list of defined scenarios for this element.  You can change the order in which the scenarios are listed, using the  and  buttons.	

### Notes

- The Scenario page does not prevent you from creating more than one basic path, but it would be unusual to define more than one
- All the functions available on the Scenario page are also available through the Scenarios & Requirements window/view; use the **Browse Element** icon in the window toolbar to list and select the scenarios for the element

### Learn more

- [Scenarios & Requirements](#)<sup>[992]</sup>
- [The Element Browser](#)<sup>[989]</sup>

#### 5.6.4.1.8.1 Structured Specification Tab

The Structured Specification tab offers a wide range of facilities for generating and modifying scenario specifications, enabling you to define the structure, actions and interactions of the scenarios defined for an element such as a Use Case. These scenarios can be the main (basic) path, alternate paths or exception paths.

When you open the Structured Specification tab it defaults to the basic path so that if it does not already exist, you can create it. You can create alternate paths and exception paths as part of the process of adding them to steps of the basic path.

You can also create all three types of scenario paths on the Description tab, or in the Scenarios & Requirements window.

### Learn more

- [Scenarios](#)<sup>[965]</sup>
- [Scenarios & Requirements](#)<sup>[992]</sup>
- [Structured Specification Toolbar](#)<sup>[970]</sup>
- [Item context menu](#)<sup>[972]</sup>
- [Selected Text context menu](#)<sup>[974]</sup>
- [Entry Points context menu](#)<sup>[976]</sup>
- [Floating Toolbar](#)<sup>[976]</sup>

You can create a specification for a scenario in one of several ways:

- Enter the specification yourself, as described below
- Generate a specification from an Activity diagram created under a Use Case element
- Generate a specification from the notes text of the scenario in the Description tab
- Generate a specification from text held on the clipboard

### How to

To enter the specification yourself, starting with the basic path

Step	Action	See also
1	In the <b>Scenario</b> field, click on the drop-down arrow and select the <i>Basic Path</i> scenario.	
2	In the <i>new step</i> field in the <b>Action</b> column, type the text of the first step or action.	
3	Tab to the <b>Uses</b> column and, if necessary, type the name of each element used in this step; any elements that are listed in the Context References tab are highlighted in blue and underlined.  You can also manage the list of elements as context references, and add to the list directly; see the Notes below.	
4	Tab to the <b>Results</b> column and, if necessary, type the outcome of completing this step.	
5	Tab to the <b>State</b> column and, if necessary, type the name of the state into which the step moves the action.	
6	When you move out of the <b>Action</b> column, the next <i>new step</i> field displays underneath. Repeat steps 2 to 5 as many times as is necessary. The Structured Specification tab should now resemble the following illustration:	








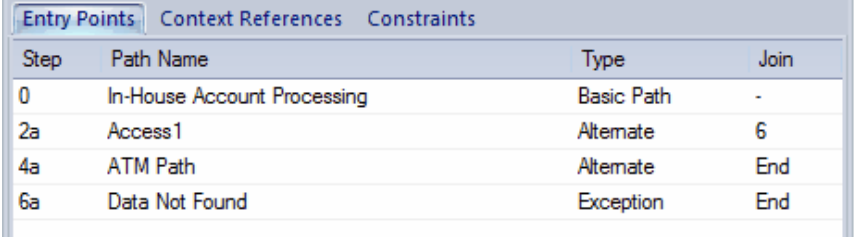





- Repeat the **Scenario Steps** procedure for each scenario you have created; you can now adapt, enhance and interrelate the scenario specifications using the facilities of the Structured Specification tab:
  - Structured Specification Toolbar
  - Item context menu
  - Selected Text context menu
  - Entry Points context menu
  - Floating Toolbar




#### Learn more

- [Generate a Scenario from an Activity diagram](#)<sup>[985]</sup>
- [Scenarios](#)<sup>[965]</sup>
- [Structured Specification Item Context Menu](#)<sup>[972]</sup>
- [Generate Diagrams](#)<sup>[976]</sup>
- [Context References Tab](#)<sup>[987]</sup>
- [Structured Specification Toolbar](#)<sup>[970]</sup>
- [Scenario Constraints Tab](#)<sup>[988]</sup>
- [Structured Specification Entry Points Context Menu](#)<sup>[976]</sup>
- [Structured Specification Floating Toolbar](#)<sup>[976]</sup>
- [ScenarioStep Class](#)<sup>[2905]</sup>

The icons on the Structured Specification toolbar offer the following facilities:

Icon	Action	See also
	Save changes to the scenario specification.	
	Return to the basic path specification (if another specification is currently displayed).	
	<p>(Only enabled when the basic path is displayed - you cannot add an alternate path to another alternate path or an exception path.)</p> <p>Create a branch from the selected step to an alternate path scenario - select the path from the displayed list. If the appropriate scenario does not yet exist, double-click on the <i>new path</i> line and type the scenario name, then click off the line and back on to it.</p> <p>Click on the <b>OK</b> button.</p> <ul style="list-style-type: none"> <li>• An entry for this alternate path displays in the Entry Points tab, as <b>Step a</b> of the basic path step it branches from; in the <b>Join</b> column, click on the drop-down arrow and select the number of the step at which action flows back to the basic path, or select <b>End</b> if the path terminates separately from the basic path</li> <li>• You cannot add more than one branch - whether alternate or exception - to a step</li> <li>• You cannot join the end of the branch - whether alternate or exception - to more than one step of the basic path. If you change the join point of</li> </ul>	

Icon	Action	See also
	<p>the branch, any duplicates of that branch are updated to remain in synchrony with the join point</p> 	
	<p>(Only enabled when the basic path is displayed - you cannot add an exception path to another exception path or an alternate path.)</p> <p>Create a branch from the selected step to an exception path scenario - select the path from the displayed list. If the appropriate scenario does not yet exist, double-click on the <i>new path</i> line and type the scenario name, then click off the line and back on to it.</p> <p>Click on the <b>OK</b> button.</p> <ul style="list-style-type: none"> <li>An entry for this exception path displays in the Entry Points tab, as <b>Step a</b> of the basic path step it branches from; in the <b>Join</b> column, click on the drop-down arrow and select the number of the step at which action flows back to the basic path, or select <b>End</b> if the path terminates separately from the basic path</li> <li>You cannot add more than one branch - whether alternate or exception - to a step</li> <li>You cannot join the end of the branch - whether alternate or exception - to more than one step of the basic path. If you change the join point of the branch, any duplicates of that branch are updated to remain in synchrony with the join point</li> </ul>	
	Display the Manage Uses Context List dialog, which you use to set the <b>Uses</b> column entries to linked context reference lists.	<a href="#">Structured Specification Selected Text Context Menu</a> [974]
	Move the currently-selected step one place up (including any <i>new step ...</i> entry).	
	Move the currently-selected step one place down (including any <i>new step ...</i> entry).	
	Display a list of diagrams that you can generate from the scenario; select the type of diagram that you want to generate.	<a href="#">Generate Diagram</a> [976]

Icon	Action	See also
	Generate Test Cases based on this Use Case scenario; you can generate either internal Test Cases or External Test Cases.	<a href="#">Generate Test Cases</a> <sup>[986]</sup>
	Delete the selected step from the scenario.	
	Display the Help topic for this tab.	

To display the context menu for a structured specification item, right-click on a step or blank line on the Structured Specification tab. The following options are available:

Option	Action	Shortcut	See also
<b>Create Structure From Clipboard Text - New Line Delimited</b> <b>Create Structure From Clipboard Text - Sentence Delimited</b>	Generate a set of steps from a text description or list captured on the clipboard; a new step is generated: <ul style="list-style-type: none"> <li>• After each carriage return in the captured text (New Lines), or</li> <li>• For each sentence in the text; that is, after each full stop/space/capital letter combination (Sentences)</li> </ul> If a set of steps is already displayed, it is overwritten by the generated steps.		
<b>Create Structure From Generated Activity Diagram</b>	Generate a set of steps from an Activity Diagram created for a Use Case. If a set of steps is already displayed, it is overwritten by the generated steps.		<a href="#">Generate Scenario from Activity Diagram</a> <sup>[985]</sup>
<b>Add Alternate Path</b>	Create a branch from the selected basic path step to an alternate path.		<a href="#">Alternate path</a> <sup>[970]</sup>
<b>Add Exception Path</b>	Create a branch from the selected basic path step to an exception path.		<a href="#">Exception path</a> <sup>[971]</sup>
<b>Insert Step Above</b>	Insert a <i>new step...</i> line above the currently-selected step (press ( <b>Esc</b> ) to return this new line to the end of the scenario).	<b>Shift+Insert</b>	
<b>Insert Step Below</b>	Insert a <i>new step...</i> line below the currently-selected step (press ( <b>Esc</b> ) to return this new line to the end of the scenario).	<b>Insert</b>	

Option	Action	Shortcut	See also
<b>Insert End Step</b>	Insert a <i>new step...</i> line at the end of the scenario.	<b>Ctrl+N</b>	
<b>Set Step As 'User'</b> <b>Set Step As 'System'</b>	Switch the entity responsible for performing the action of the selected step between user and system.		
<b>Link Step to Use Case</b>	<p>Either include the actions of an existing Use Case element, extend an existing Use Case element, or invoke a Use Case as the action of the selected step.</p> <p>Selecting the appropriate sub-option displays the Select Use Case dialog, which you use to browse for and select the required Use Case element.</p> <p>The appropriate <i>includes</i>, <i>extends</i> or <i>invokes</i> stereotyped connector is created between the current element and the selected Use Case.</p> <p>For the <i>include</i> and <i>extend</i> actions, any text in the <b>Action</b> field is overwritten by the link to the Use Case; for the <i>invoke</i> action, the following link is added to the end of the <b>Action</b> text:</p> <p>( <i>Invokes: &lt;Use Case Name&gt;</i> )</p>		<a href="#">Select Use Case</a> 994
<b>Merge With Step</b>	<p>Merge the selected step with another.</p> <p>A list of the other steps in the scenario displays; click on the step to merge with the selected step.</p>		
<b>Move After Step</b>	<p>Move the selected step to another position in the scenario.</p> <p>A list of the other steps in the scenario displays; click on the step after which to position the selected step.</p>		
<b>Delete</b>	<p>Delete the selected step; any subsequent steps are moved up one place.</p> <p>A prompt displays to confirm the deletion.</p>		

To display this context menu, *highlight* the text in a user-editable field within a step on the **Structured Specification** tab. The following options are available:

Option	Action	Shortcut	See also
<b>Create</b>	Create a glossary definition or a new element based on the highlighted text.		<a href="#">Glossary definition</a> <sup>[1142]</sup> <a href="#">New element</a> <sup>[903]</sup>
<b>Link Step to Use Case</b>	<p>Either incorporate the actions of an existing Use Case element, extend an existing Use Case element or invoke a Use Case element, as the action of the selected step. Selecting the appropriate sub-option displays the Select Use Case dialog, which you use to browse for and select the required Use Case element.</p> <p>Any text in the Action field is overwritten by the link to the Use Case, except for the invoke action where the following link is added to the end of the Action text:</p> <p>( <i>Invokes: &lt;Use Case Name&gt;</i> )</p>		<a href="#">Select Use Case</a> <sup>[994]</sup>
<b>Link to existing Element</b>	<p>(Uses and Results fields only.) Create a Realization or Dependency relationship to a Requirement, Feature or other element elsewhere in the model.</p> <p>You select the element and connector types from submenu options, which then display the Select Element dialog, which you use to browse for and select the required element.</p>		<a href="#">Select Element</a> <sup>[994]</sup>
<b>Insert context reference</b>	<p>Add a reference to an element stored elsewhere in the model, and create an entry for the element in the Context References tab.</p> <p>Selecting this option displays the Select Element dialog, which you use to browse for and select the required reference element.</p>		<a href="#">Context References</a> <sup>[987]</sup> <a href="#">Select Element</a> <sup>[994]</sup> <a href="#">Structured Specification Floating Toolbar</a> <sup>[976]</sup>
<b>Insert glossary definition</b>	<p>Insert an existing glossary term at the cursor position. To select the term, double-click on it in the displayed list.</p> <p>When you select the term it is inserted into the field as highlighted and underlined text, which displays the definition when you move the cursor over it. If you highlighted part of the original text, the term overwrites that text.</p>		
<b>Manage Uses Context List</b>	(On highlighted text or a specific cursor position in the Uses column.)		<a href="#">Context References Tab</a> <sup>[987]</sup>

Option	Action	Shortcut	See also
	<p>Display the Manage Uses Context List dialog, which enables you to:</p> <ul style="list-style-type: none"> <li>• Browse to or type in a context reference item</li> <li>• Re-sort the list with up and down arrow keys</li> <li>• Remove an entry from the list</li> <li>• Enable or disable the Uses Context List facility across the model</li> </ul> <p>You can resize the dialog to show the full details of context reference items.</p> <p>If you enable the Uses Context List facility, you do not necessarily have to use the dialog to add context list items; you can type or insert an element name from the Context References tab in the Uses column, and that name is added to the Uses Context List.</p> <p>If you disable the Uses Context List facility, the context list items in the Uses column are hidden.</p>		
<b>Split Step</b>	<p>Splits the selected step into two consecutive steps.</p> <p>The option is available only if you highlight a <i>portion</i> of the text in the selected field. The new step takes the highlighted text as its <b>Action</b> text.</p>		
<b>Search for &lt;text&gt;</b>	Displays a sub-menu of options for locating the selected text in a number of locations.		<a href="#">Code editor context menu</a> <sup>[2152]</sup>
<b>Undo</b>	Undo any unsaved changes you have just made in the step.		
<b>Cut</b>	Perform simple editing operations on the highlighted text.		
<b>Copy</b>			
<b>Paste</b>			
<b>Delete</b>			
<b>Select All</b>			

Option	Action	Shortcut	See also

The Entry Points tab shows how the basic path, alternate path and exception path scenarios for the element are organized and interrelated. If an alternate path or exception path has been defined but has not yet been added to the basic path, it is not listed on this tab.

You can switch focus between the Entry Points tab and the Structured Specification tab by pressing ( **Alt+Q** ).

To display the context menu for this tab, highlight an entry and right-click on it. The following options are available:

Option	Action	Shortcut	See also
<b>Edit Path</b>	Display the steps of the scenario in the Structured Specification tab, with the first step highlighted.		
<b>Join with Step</b>	(Available only if the basic path scenario is displayed in the Structured Specification tab. Not available to edit the basic path scenario.)  Highlight the <b>Join</b> field and its drop-down arrow. Click on the drop-down to define or change the step number at which the alternate or exception path rejoins the basic path. Select <b>End</b> if the path does not rejoin the basic path steps.		
<b>Remove Entry Point</b>	(Available only if the basic path scenario is displayed in the Structured Specification tab. Not available to delete the basic path.)  Delete the relationship between the selected path and the basic path, and remove the entry from the Entry Points tab.		

Wherever a reference to another element exists on the Scenario tab (that is, where the text is highlighted and underlined), if you hover the cursor over the element name a short floating toolbar displays.

#### Use to

- Display the element Properties dialog
- Locate the element in its parent diagram
- Locate the element in the Project Browser

#### **5.6.4.1.8.2 Generate Diagrams**

If you have created a structured scenario, you can generate any of the following diagrams from that scenario:


- Activity
  - With ActivityParameter
  - With Action
  - With Action Pin



- Rule Flow
- State Machine
- Sequence
- Robustness

### How to

To generate the required diagram

Icon	Step	Action	See also
	1	Create the scenario structure on the Structured Specification tab.	<a href="#">Structured Specification Tab</a> <sup>[967]</sup>
	2	Click on the Generate Diagram icon in the toolbar on the tab.	
	3	<p>Click on the type of diagram to generate.</p> <p>Enterprise Architect generates the diagram and notifies you that generation is complete. Close the Properties dialog to review the diagram.</p> <p>If the diagram being generated already exists under the selected element, a prompt displays to overwrite or synchronize with that diagram.</p> <p>Select the appropriate radio button to:</p> <ul style="list-style-type: none"> <li>• Overwrite the existing diagram (delete the existing diagram and elements, and create a new diagram and elements) or</li> <li>• Synchronize the elements in the existing diagram with the scenario steps (however, Sequence and Robustness diagrams cannot be synchronized)</li> </ul>	

### Notes

- The **Synchronize elements in existing diagram** option enables the **Preserve Diagram Layout** checkbox, which you can select to preserve the existing arrangement of elements and connectors on the diagram; any new elements are added to the diagram in the default position, and you manually position them in the diagram as required; if you do not select the checkbox, the diagram is recast in the default layout

It is recommended that you uncheck the **Preserve Diagram Layout** checkbox if you are synchronizing elements with scenario steps:

- When new steps have been added or existing steps have been deleted or moved within the Use Case
- For the first time in a Use Case that has been imported from XML with the **Strip GUIDs** option selected
- For the first time in a Use Case that has been copied and pasted in the Project Browser, or
- For the first time in a Use Case whose containing package has been copied and pasted in the Project Browser

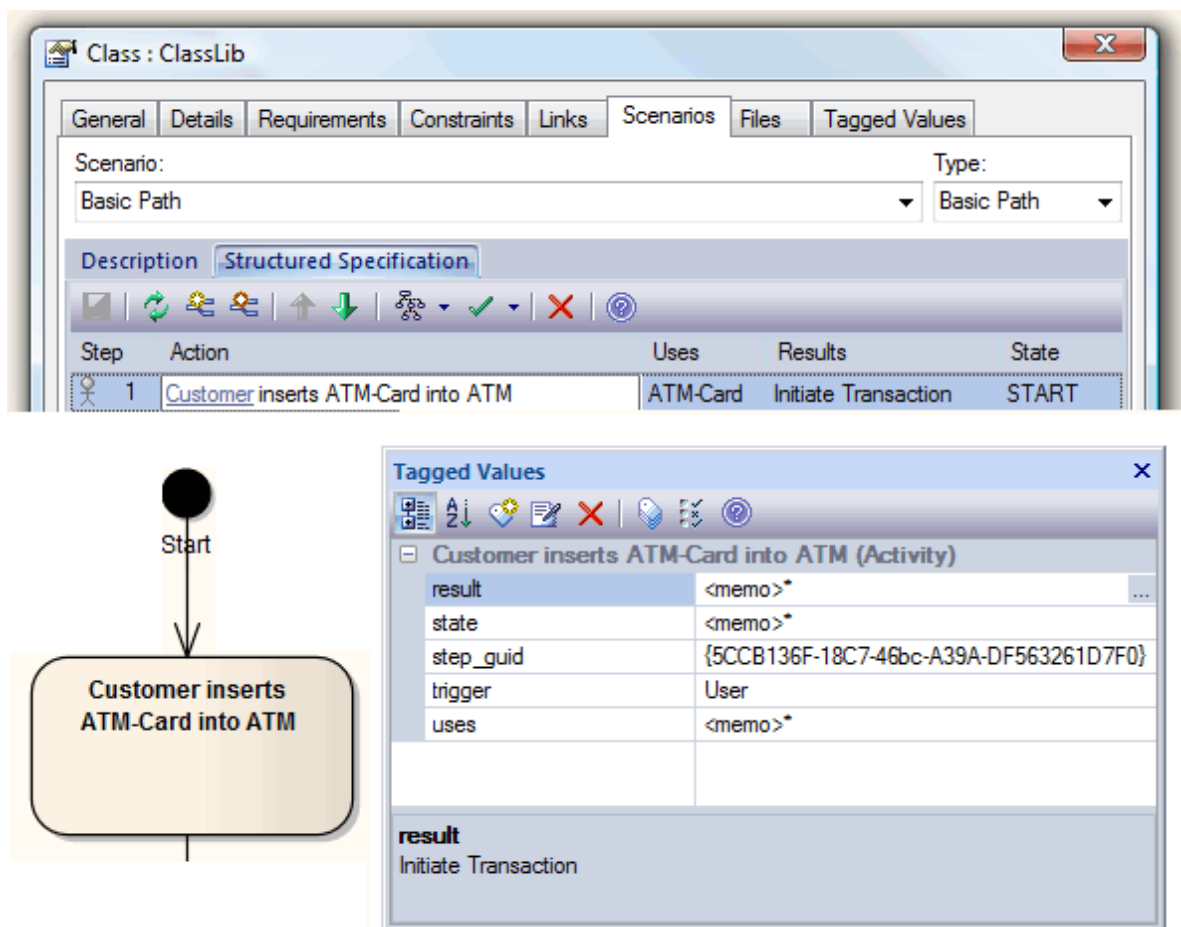
### Learn more

- [Generate Activity Diagram](#) <sup>[978]</sup>
- [ActivityParameter](#) <sup>[978]</sup>
- [Action](#) <sup>[978]</sup>
- [Action Pin](#) <sup>[978]</sup>
- [Generate Rule Flow Diagram](#) <sup>[980]</sup>
- [Generate State Machine Diagram](#) <sup>[980]</sup>
- [Generate Sequence Diagram](#) <sup>[982]</sup>
- [Generate Robustness Diagram](#) <sup>[984]</sup>
- [Import from XML](#) <sup>[478]</sup>
- [Copy Element Between Packages](#) <sup>[915]</sup>
- [Copy A Package](#) <sup>[775]</sup>

An Activity is generated as a child of the selected element, to act as a container for the diagram.

- The scenario steps are modeled as Activities
- The values of the **Uses**, **Results** and **State** columns for each step are added as Tagged Values of the corresponding Activity

### Example



### Topics

Topic	Detail	See also
<b>Activity with ActivityParameter</b>	<ul style="list-style-type: none"> <li>The values of the Uses and Results columns are modeled as ActivityParameters</li> <li>The value of the State column is added as a Tagged Value of the Activities</li> <li>ActivityParameters are added to the Project Browser and not to the diagram</li> </ul>	
<b>Activity with Action</b>	<ul style="list-style-type: none"> <li>The scenario steps are modeled as Actions</li> <li>The values of the Uses, Results and State columns are added as Tagged Values of the Actions</li> </ul>	
<b>Activity with ActionPin</b>	<ul style="list-style-type: none"> <li>The scenario steps are modeled as Actions</li> <li>The values of the Uses and Results columns are modeled as Input Pins and Output Pins respectively</li> <li>The value of the State column is added as a Tagged Value of the Actions</li> </ul>	

Topic	Detail	See also
	<ul style="list-style-type: none"><li>ActionPins are added to the Project Browser and not to the diagram</li></ul>	

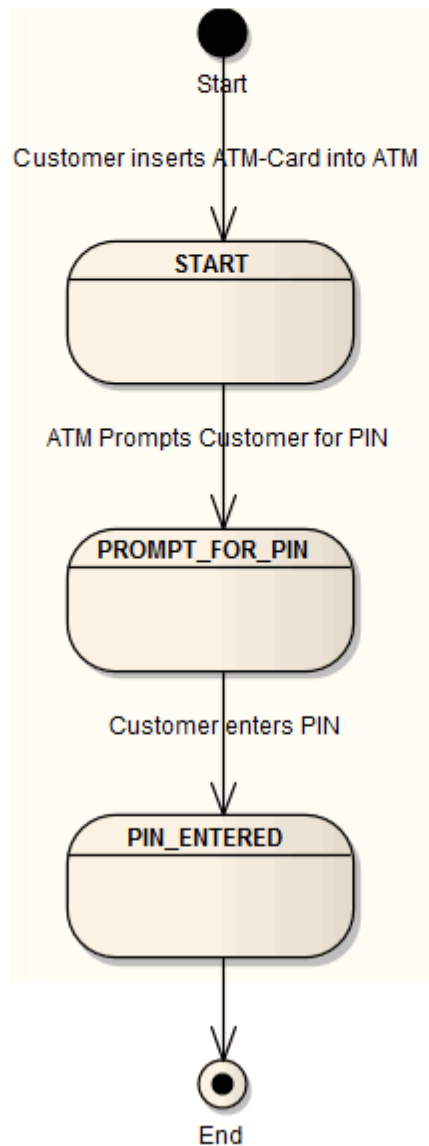
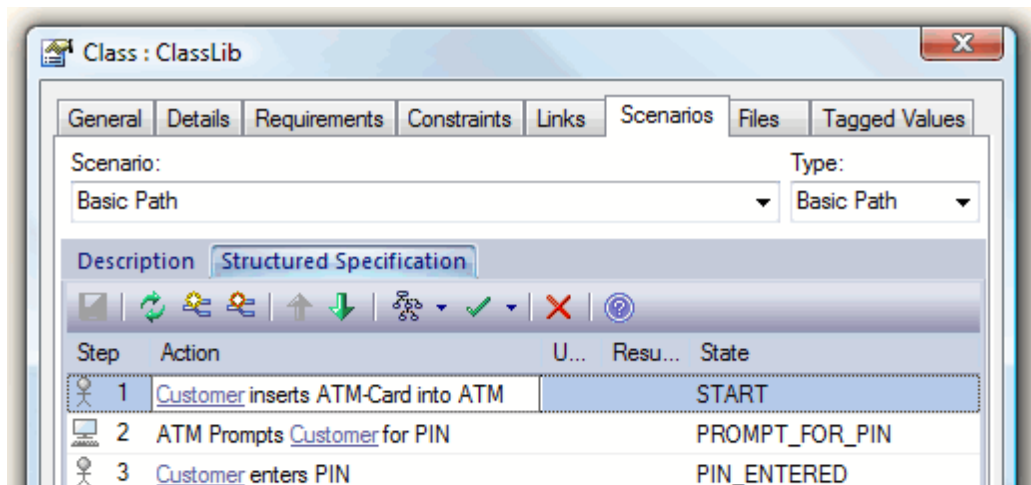
A Rule Flow Activity is created as a child of the selected element, to act as a container for the diagram.

- The scenario steps are modeled as RuleTasks.
- The values of the **Uses**, **Results** and **State** columns are added as Tagged Values of the RuleTasks

A StateMachine is created as a child of the selected element, to act as a container for the diagram.

- Each value in the *State* column is modeled as a State.
- The scenario steps become the Transition connectors between the States
- The values of the *Uses* and *Results* columns are added as Tagged Values of the Transitions

### Example

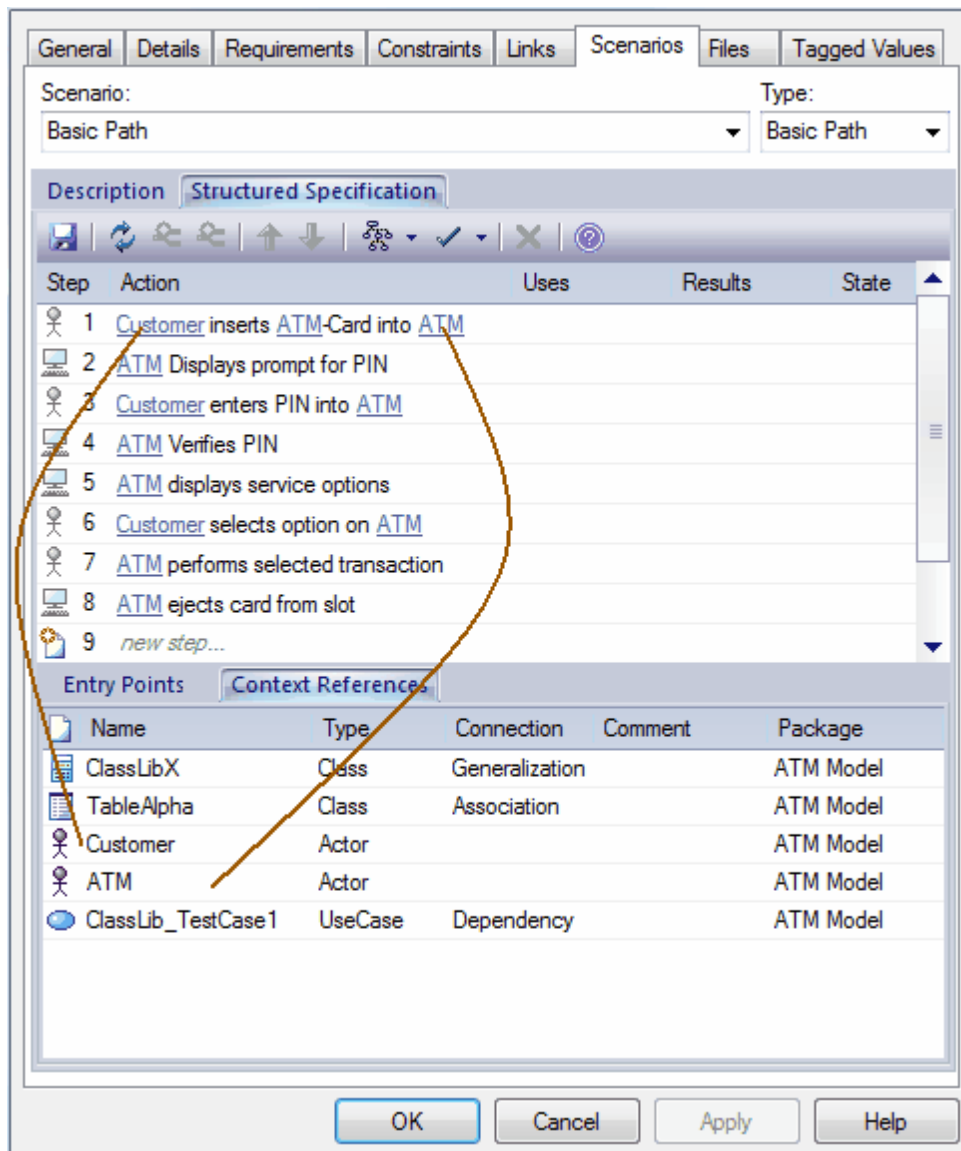


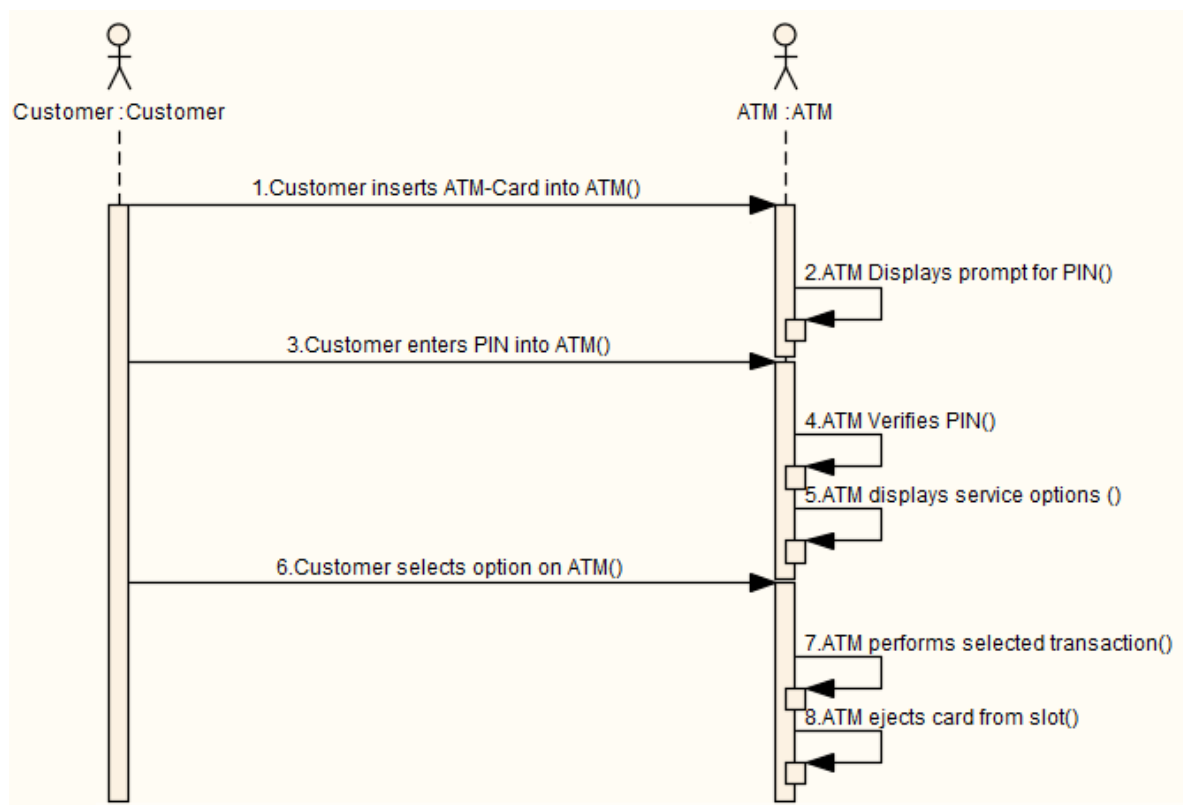
An Interaction is created as a child of the selected element, to act as a container for the diagrams - the Basic, Alternate and Exception paths are modeled as separate Sequence diagrams under the Interaction.

### Topics

Topic	Detail	See also
<b>Usage</b>	<p>All the elements involved in the scenario should be identified in the Context Reference tab. That is, relationships must already exist between the scenario parent element and the other elements named in the scenario.</p> <ul style="list-style-type: none"> <li>Each Context Reference element named in a scenario step is modeled as a Lifeline</li> <li>The step itself becomes the Message between an originator and its destination(s)</li> <li>The first Context Reference element in a step is treated as the originator</li> <li>The subsequent Context Reference element(s) become the destination(s)</li> </ul> <p>Because the diagram generator acts on element names in the step, you should take care to avoid using the element names as normal text. For example, in step 1 in the dialog below, the term ATM-Card is interpreted as a reference to the ATM element, and two <i>Customer inserts ATM-Card into ATM</i> messages are generated for the step. (To avoid confusion, in the diagram the first Message has been deleted.)</p> <p>If the <b>Show Sequence Numbering</b> checkbox on the Diagram Sequence page of the Options dialog is not selected, the Message name on the Sequence diagram is prefixed by the step number, as shown in the diagram below. If the checkbox is selected, the message sequence number is shown instead of the step number.</p> <p>The values of the Uses, Results and State columns are added as Tagged Values of the Message.</p>	<a href="#">Context Reference Tab</a> <sup>[987]</sup> <a href="#">Sequence</a> <sup>[629]</sup>



### Example



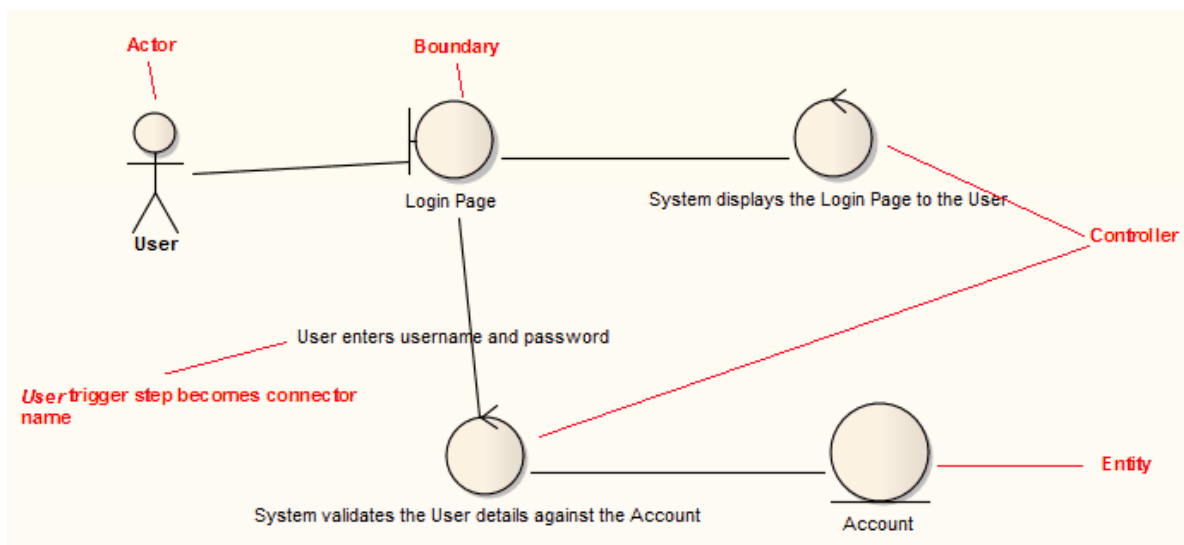


A Collaboration is created as a child of the selected element, to act as a container for the Robustness diagram.

### Example

Step	Action
 1	System displays the <a href="#">Login Page</a> to the <a href="#">User</a>
 2	<a href="#">User</a> enters username and password
 3	System validates the <a href="#">User</a> details against the <a href="#">Account</a>





### Notes

- All the elements involved in the scenario should be identified in the Context Reference tab; that is, relationships must already exist between the scenario parent element and the other elements named in the scenario
- Any values in the **Uses**, **Results** and **State** columns are ignored and not represented in the diagram
- Each UI element in a step becomes a Boundary element; a Dependency relationship is created from this Boundary element to the UI element (this connector is not shown on the diagram)
- Each Actor referenced in a step is dropped into the Robustness diagram as a simple link
- Each Class referenced in a step is dropped into the Robustness diagram as a simple link, and is given the stereotype *entity*
- Each step with a *System* trigger becomes a Controller; alternate/exception path Controllers are displayed with a red background color
- Each step with a *User* trigger becomes the name of the Association between Controllers

### Learn more

- [Collaboration](#) <sup>[1367]</sup>
- [Robustness Diagram](#) <sup>[1800]</sup>
- [Context Reference Tab](#) <sup>[987]</sup>
- [UI Elements](#) <sup>[1992]</sup>
- [Boundary Elements](#) <sup>[1997]</sup>
- [Dependency](#) <sup>[1404]</sup>
- [Control](#) <sup>[1999]</sup>
- [Associate](#) <sup>[1393]</sup>

#### 5.6.4.1.8.3 Generate Scenario From Activity Diagram

You can generate a range of diagrams from a scenario in an element. Conversely, you can also generate a structured scenario within an element from an Activity diagram, reverse engineering the steps from the diagram elements (effectively either regenerating the scenario within the Use Case, or transferring a scenario into another Use Case).

**How to**

To generate the scenario from the Activity diagram

Step	Action	See also
1	Open the element Properties dialog, select the Scenarios tab, and select the Structured Specification tab.	
2	Right-click in the empty space within the tab, and select the <b>Create Structure From Generated Activity Diagram</b> context menu option.  The Select an Activity (generated from a UseCase Scenario) containing the Diagram dialog displays.	<a href="#">Select &lt;Item&gt; Dialog</a> <sup>[994]</sup>
3	Search for and select the Activity containing the required diagram.  Enterprise Architect validates the diagram (displaying the results in the Output window) and, if the diagram is valid, generates the scenario steps in the Structured Specification tab (replacing any existing scenario steps).	<a href="#">Output Window</a> <sup>[169]</sup>

**Notes**

- The source Activity diagram must be generated from another Use Case Scenario
- Any existing scenario steps are deleted and replaced by the generated scenario
- This facility does not operate on the enhanced Activity diagrams generated from a Use Case - those generated with ActivityParameters, Actions and Action Pins

**Learn more**

- [Generate Diagrams](#) <sup>[976]</sup>

**5.6.4.1.8.4 Generate Test Cases**

When you select a scenario and click on the **Test Case Generation** icon in the window toolbar, Enterprise Architect prompts you to select to generate either an Internal Test Case or an External Test Case.

**Topics**

Topic	Detail	See also
<b>Internal Test Case</b>	<p>A test is generated for the basic path and each alternate and exception path in the scenario, and added to the selected element.</p> <p>In addition, for each step in the basic, alternate and exception paths that has a value in the <b>Results</b> column, a test is generated and added to the selected element.</p> <p>A diagram is created under the selected element, to which the selected element and a Note (showing the element's Structured Specification) is added.</p>	<a href="#">Show Test Scripts compartment</a> <sup>[2618]</sup> <a href="#">Working on Test Records</a> <sup>[2605]</sup> <a href="#">Move Or Copy Tests Between Categories</a> <sup>[2610]</sup>

Topic	Detail	See also
	<p>The created tests are displayed in the <i>Test Scripts</i> compartment of the selected element on the diagram.</p> <p>Another way to view these tests is to click on the element and display the Testing window.</p> <p>These generated tests are written to the Scenario test tab of the Testing window; you can move the tests to another test-type tab if required.</p>	
<b>External Test Case</b>	<p>A Test Case element is created, linked to the selected element using the Trace connector.</p> <p>A diagram is created under this Test Case element, to which the selected element, the Test Case element and a Note (showing the element's Structured Specification) is added.</p> <p>The created tests are displayed in the <i>Test Scripts</i> compartment of the Test Case element on the diagram.</p> <p>A test case is generated and added to the Test Case element for the basic path, and for each alternate and exception path.</p> <p>In addition, for each step in the basic, alternate and exception paths that has a value in the <b>Results</b> column, a test is generated and added to the Test Case element.</p> <p>You can review the tests within the Test Case element using the Testing window, as for Internal Test Cases.</p>	<p><a href="#">Show Test Scripts compartment</a><sup>[2616]</sup></p> <p><a href="#">Test Case</a><sup>[2010]</sup></p> <p><a href="#">Trace</a><sup>[1446]</sup></p>

#### 5.6.4.1.8.5 Context References Tab

On the Scenarios page of the element Properties dialog, the Context References tab displays a list of all elements that are either:

- Connected to the current element by any connector, on the current diagram or another, or
- Defined as a cross reference (or custom reference) on the current element

#### Topics

Topic	Detail	See also
<b>Usage</b>	<p>This page enables you to add custom references; right-click anywhere in the list and select the <b>Add Reference</b> context menu option.</p> <p>The Select Element dialog displays, in which you can locate and select the required cross reference element or elements.</p> <p>For each cross reference you can also use context menu options to delete the entry in the list, or to open the <b>Comment</b> field so that you can add or edit comment text.</p> <p>For each element in the <b>Context References</b> list, wherever the name of that element appears in the structured specification, the name is</p>	<p><a href="#">Select &lt;Item&gt; Dialog</a><sup>[994]</sup></p>

Topic	Detail	See also
	highlighted and underlined; press ( <b>Ctrl</b> ) + <b>click</b> on the highlighted name to view the element Properties dialog.	

#### Learn more

- [Custom/Cross Reference](#)<sup>[916]</sup>

#### **5.6.4.1.8.6 Scenario Constraints Tab**

The Structured Specification Constraints tab is a simple link to the Constraints tab of the element overall. It lists existing constraints; if you select the toolbar icons to add or edit a constraint, control switches to the element Constraints tab.

#### Learn more

- [Constraints](#)<sup>[963]</sup>

#### **5.6.4.1.9 Associated Files**

An element can be linked to files external to the repository, using the Files page of the element's Properties dialog.

Field/Button	Usage	See also
<b>File Path</b>	Specifies the directory path and name of the file.	
<b>Type</b>	Display the local file or web address.	
<b>Last Write</b>	Display the date and time the file was last updated.	
<b>Size</b>	Display the size of the file.	
<b>Notes</b>	Indicates extra information about the file.	
<b>Files</b>	Display the list of linked files.	
<b>Launch</b>	Open the selected file.  Local files open with their default application and web files open in the default browser.	

#### Notes

- Linked files are a good way to link elements to additional documentation and/or source code
- You can also insert hyperlinks in diagrams to other files, and launch them directly from the diagram

#### Learn more

- [Hyperlinks](#)<sup>[2002]</sup>
- [The Element Browser](#)<sup>[989]</sup>

### 5.6.4.2 The Element Browser

As you develop an element in your model, you can define a range of added-on properties of that element through the appropriate dialogs, pages and tabs. It is possible to neatly summarize all of these properties for a selected element, using the Element Browser. The properties listed - where they exist in the element - are grouped by type and include:


- Operations
- Attributes
- Linked Features (Linked Operations and Attributes)
- Tagged Values
- Constraints
- Internal Requirements (Responsibilities)
- Attached Files
- Relationships
- Scenarios
- Maintenance Items (sub-grouped as Defects, Changes, Issues and Tasks)
- Testing Items (sub-grouped as Unit, Integration, System, Acceptance and Scenario tests)
- Project Management Items (sub-grouped as Resource Allocation, Effort, Risk and Metrics records)

Having listed these properties, you can also perform a number of operations on them, including dragging and dropping many of them onto other elements.

Access    **Element | Element Browser**    (Alt+9)

#### Options

Option	Action	See also
<b>Display element Properties dialog</b>	Double-click on the element name at the top of the dialog (the root node).	<a href="#">Properties Dialog</a> <sup>[956]</sup>
<b>Add items of a specific type</b>	<p>Right-click on the property type name and select the <b>Add New &lt;object&gt;</b> context menu option. (This option is not applicable to Relationships or Linked Features, which cannot be created without using information external to the selected element.)</p> <p>The appropriate window or dialog displays, or the element Properties dialog displays on the appropriate page. Add new or edit existing items</p>	

	<p>as necessary.</p> <p>For example, if you right-click on the <i>Attributes</i> folder and select the <b>Add New Attributes</b> context menu option, the &lt;element name&gt; Attributes dialog displays and you can add to or edit the attributes in the element.</p>	
<b>Edit specific items</b>	<p>Expand the appropriate folder and double-click on the required <b>item</b>.</p> <p>The Properties dialog, docked window or view displays with the focus on the selected item. You can also go on to select and edit other items within the window or dialog.</p>	
<b>Drag a relationship onto a diagram</b>	<p>Click on the relationship in the Element Browser and drag it onto the diagram.</p> <p>If the element at one or both ends of the relationship is not already on the diagram, those elements are also added to the diagram.</p> <p>If the relationship is already on the diagram, but hidden, it becomes visible again.</p> <p>Alternatively:</p> <ul style="list-style-type: none"> <li>• Right-click on the <i>Relationships</i> folder name and select the <b>Place All Related Elements in Diagram</b> context menu option, or</li> <li>• Right-click on the relationship and select the <b>Place Related Elements in Diagram</b> context menu option</li> </ul>	
<b>Hide/Show a Relationship in the current diagram</b>	<p>Right-click on the relationship in the Element Browser and select the <b>Hide Relationship</b> or <b>Show Relationship</b> context menu option as appropriate.</p>	
<b>Display the Properties dialog of the other element in the relationship</b>	<p>Right-click on the relationship in the Element Browser and select the <b>View Related Element Properties</b> context menu option.</p>	
<b>Locate the related element in the Project Browser</b>	<p>Right-click on the relationship in the Element Browser and select the <b>Locate Related Element</b> context menu option.</p>	
<b>Locate the related element in other diagrams</b>	<p>Right-click on the relationship in the Element Browser and select the <b>Find in all Diagrams</b> context menu option.</p>	
<b>Open an associated file</b>	<p>Double-click on the required file path within the <i>Files</i> folder.</p> <p>If the file can:</p>	<a href="#">Associated Files</a> 

	<ul style="list-style-type: none"> <li>• Be opened within Enterprise Architect, the file opens on a separate tab in the Diagram View workspace</li> <li>• Not be opened within Enterprise Architect, the file opens in the default Windows viewer/editor for the file type</li> </ul>	
<b>Display the source code for a feature</b>	<p>Right-click on an attribute or operation in the <i>Attributes</i> or <i>Operations</i> folder or in the <i>Linked Features</i> folder, then click on the <b>View Source Code</b> context menu option.</p> <p>Alternatively:</p> <ul style="list-style-type: none"> <li>• Click on the attribute or operation and press <b>Ctrl+E</b>, or</li> <li>• In the <i>Linked Features</i> folder, double-click on the attribute or operation</li> </ul> <p>If no code has been generated for the selected feature, nothing happens.</p> <p>If code has been generated for the feature, the code is displayed. The source code viewer in which the source code displays depends on which editor you select as the default, either for the project as a whole or for a specific programming language; if you select the Enterprise Architect internal editor, the code displays in the Source Code Viewer with the cursor positioned on the selected feature.</p>	<a href="#">Options - Code Editors</a> <sup>[2250]</sup> <a href="#">Editing Source Code</a> <sup>[2146]</sup>
<b>Add a property to another element</b>	<p>Click on the property in the Element Browser and drag it onto the other element in a diagram. The properties you can do this for are:</p> <ul style="list-style-type: none"> <li>• Attributes</li> <li>• Operations</li> <li>• Constraints</li> <li>• Requirements</li> <li>• Tagged Values</li> <li>• Maintenance Items</li> <li>• Test Items</li> </ul>	
<b>Attach a Recording Marker to an Operation</b>	<p>Right-click on the operation and select the <b>Visual Analyzer   Set Breakpoint</b> context menu option, and the option to set the required type of recording marker.</p>	<a href="#">Place Recording Markers</a> <sup>[2538]</sup>

#### Learn more

- [Connector Properties](#) <sup>[1126]</sup>
- [Operations](#) <sup>[1014]</sup>
- [Attributes](#) <sup>[999]</sup>
- [Connect to Element Feature \(Linked Operations and Attributes\)](#) <sup>[1110]</sup>
- [Tagged Values](#) <sup>[1134]</sup>
- [Constraints](#) <sup>[963]</sup>

- [Requirements](#)<sup>[967]</sup> (Responsibilities)
- [Links](#)<sup>[964]</sup>
- [Scenarios](#)<sup>[965]</sup>
- [Working on Maintenance Items](#)<sup>[2623]</sup>
- [Working on Test Records](#)<sup>[2605]</sup>
- [Project Management](#)<sup>[509]</sup>

### 5.6.4.3 Properties Window

The Properties window provides a convenient way to view (and in some cases edit) common properties of elements. When an element is selected, the Properties window shows the element's name, stereotype, version, author, dates and other pertinent information.

**Access**    **Element | Properties Window**    (Alt+1)

#### Topics

Topic	Detail	See also
<b>Properties Sections</b>	<p>The Properties window is divided into three expandable sections:</p> <ul style="list-style-type: none"> <li>• &lt;Element type&gt; Settings - for the basic element details</li> <li>• Project - for general housekeeping settings</li> <li>• Advanced - only active for generalizable elements</li> </ul>	

#### Notes

- When you click on a field name, a brief explanation of that field displays at the bottom of the Properties window, unless you have selected the **Hide Properties Info Section** checkbox on the General page of the Options dialog
- If you click on the field value for an editable field, a drop-down arrow displays that enables you to select a different value; one of the values offered might display a more advanced list or a dialog for adding new objects or selecting multiple objects, as the **Stereotype** field does

#### Learn more

- [General Options](#)<sup>[607]</sup>
- [Apply Stereotypes](#)<sup>[1453]</sup>

### 5.6.4.4 Element Scenarios & Requirements


The Scenarios & Requirements window provides a convenient way to quickly add, view, edit and delete rules applied to an element. The window shows details of the entities that impose such rules or restrictions on the element, namely:

- The element's internal responsibilities or requirements
- The element's internal constraints (not external Constraint notes)
- The element's scenarios




**Access** **Element | Scenarios & Rules** (Ctrl+Shift+3)

### Topics

Topic	Detail	See also
<b>Usage</b>	<p>The Scenarios &amp; Requirements window is typically used to examine Use Case and Test Case elements, and any other elements that realize an external Requirement.</p> <p>For convenience, you can display the window as either a dockable window around the edge of your workspace, or as a view in the center of your workspace (in which the Scenarios, Constraints and Requirements are shown on separate tabs).</p> <p>Use the  icon in the window toolbar to toggle between these display options.</p>	<a href="#">Notes</a> <sup>[1324]</sup>

### How to

To review an internal requirement, constraint or scenario for an element

Step	Action	See also
<b>1</b>	<p>Select the element in the Project Browser or diagram and select the <b>Scenarios &amp; Rules</b> menu option.</p> <p>The Scenarios &amp; Requirements window displays.</p>	
<b>2</b>	<p>Either:</p> <ul style="list-style-type: none"> <li>Click on the required item in the list panel on the left of the window, or</li> <li>If the Element Browser is not already displayed, click on the  icon and select the required item from the <i>Constraints</i>, <i>Requirements</i> or <i>Scenarios</i> folder</li> </ul> <p>The appropriate screen displays, with the details of the selected item in the fields.</p>	<a href="#">Element Browser</a> <sup>[989]</sup>
<b>3</b>	<p>The toolbar icons, from left to right, enable you to:</p> <ul style="list-style-type: none"> <li>Add a new rule to the current screen (click on the appropriate folder to create an item of a different type)</li> <li>Save the new or edited item</li> <li>Save the current (edited) item as a new item</li> <li>Delete the current item</li> <li>Sort the contents of the selected folder into alphabetical order</li> </ul>	<a href="#">Notes Toolbar</a> <sup>[1143]</sup> <a href="#">Converting an Internal Responsibility Into an External Requirement</a> <sup>[1771]</sup> <a href="#">Scenarios</a> <sup>[965]</sup> <a href="#">Constraints</a> <sup>[963]</sup>

Step	Action	See also
	<ul style="list-style-type: none"> <li>• (Print icon unavailable)</li> <li>• Display the Element Browser window, or highlight the current item in the Element Browser window</li> <li>• Switch between a docked window display and a workspace view display</li> </ul> <p>You can also add or edit formatted notes in the <b>Notes</b> field, using the Notes toolbar at the top of the field.</p> <p>The list panel also provides a context menu that provides options for adding a new item to a folder, deleting the selected item or, for a requirement, converting an internal responsibility into an external Requirement element.</p>	<a href="#">Requirements</a> <sup>[96]</sup>

#### 5.6.4.5 Select <Item> Dialog

The Select <Item> dialog is a multi-purpose browser and search tool for locating model items such as Classifier elements, properties, attributes and behaviors. The <Item> in the dialog title changes to represent the type of item the original operation is working on.

##### Topics

Topic	Detail	See also
<b>Usage</b>	<p>The dialog is called in a range of operations; for example, setting:</p> <ul style="list-style-type: none"> <li>• The base type or classifier for an Object, Swimlane or Lifeline</li> <li>• Classifiers for the return types for operations</li> <li>• Classifiers for generalization sets</li> <li>• The Information Item or classifier elements conveyed or realized on an Information Flow</li> <li>• The associated behavior for a behavior call</li> <li>• The type and return type for operation parameters</li> <li>• Activities for State transitions</li> <li>• Use Case Scenarios</li> <li>• Pattern element defaults</li> <li>• Activities from which to generate code</li> <li>• The values of Tagged Values</li> </ul>	<a href="#">Using Classifiers</a> <sup>[1010]</sup> <a href="#">Objects</a> <sup>[1379]</sup> <a href="#">Swimlanes</a> <sup>[852]</sup> <a href="#">Lifeline</a> <sup>[1321]</sup> <a href="#">Using Information Flows</a> <sup>[1411]</sup> <a href="#">General Properties of Operations</a> <sup>[1015]</sup> <a href="#">Generalization Sets</a> <sup>[1120]</sup> <a href="#">Associate with Different Behaviour</a> <sup>[1027]</sup> <a href="#">Insert Operation Parameters</a> <sup>[1041]</sup> <a href="#">Transitions</a> <sup>[1446]</sup> <a href="#">Generate Scenarios from Activity Diagram</a> <sup>[985]</sup> <a href="#">Use a Pattern</a> <sup>[1467]</sup> <a href="#">Assigning Tagged Values</a> <sup>[1137]</sup>

<b>Multiple Selection</b>	<p>Where an operation permits the selection of multiple items, the Select &lt;Item&gt; dialog is automatically enabled to support this; to select the items, press ( <b>Ctrl</b> ) as you click on each item.</p> <p>If you want to cancel a selected item, press ( <b>Ctrl</b> ) and click on the item.</p> <p>Having selected an item, you can continue to expand and browse the hierarchy, and/or search for items; the dialog retains the existing selections until you click on the <b>OK</b> button.</p>	
---------------------------	--	--

### Select a required item

Step	Action	See also
<b>1</b>	<p>During an operation, when it is necessary to locate an element or feature, click on the ( ... ) (browse) button.</p> <p>The Select &lt;Item&gt; dialog displays.</p>	
<b>2</b>	<p>If required, in the <b>Go To Namespace</b> field select a namespace to reduce the scope of the displayed hierarchy.</p> <p>The dialog opens the section of the hierarchy associated with that namespace, and closes all previously-open sections associated with other namespaces.</p>	
<b>3</b>	<p>You can either:</p> <ul style="list-style-type: none"> <li>Expand the selected area of the hierarchy on the Browse tab, or any other package, and locate the required item (go to step 5) or</li> <li>Click on the Search tab and, in the <b>Find</b> field, type a partial or complete text string to search for the item</li> </ul>	
<b>4</b>	<p>On the Search tab, you can filter the search further by selecting the <b>Whole Word</b> and <b>Match Case</b> checkboxes.</p> <p>Each list entry shows the name of the item, the type, any stereotype the item has, the immediate package in which the item is held, and any successive parent packages (the package path).</p> <p>You can either:</p> <ul style="list-style-type: none"> <li>Select the item immediately on the Search tab or</li> <li>Right-click on one item or a group of items and select the <b>Locate item(s) in tree</b> context menu option; this redisplay the Browse tab and highlights each selected item in the &lt;namespace&gt; hierarchy</li> </ul>	
<b>5</b>	Click on the required item.	
<b>6</b>	Click on the <b>OK</b> button.	

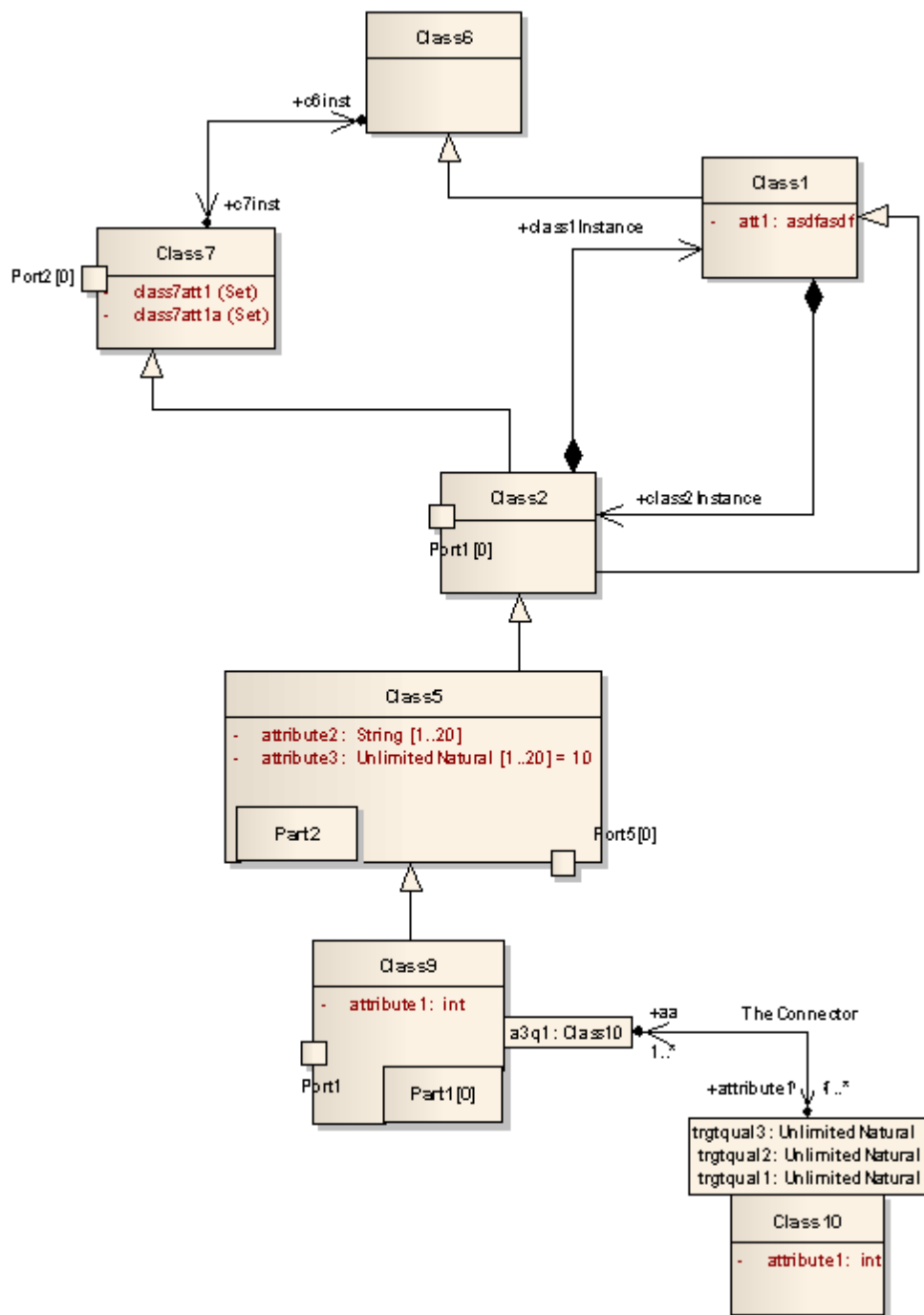
**Notes**

- When you have selected an item, the Select <Item> dialog retains the context and item; next time you display the dialog, if the context is similar the dialog opens to the same Namespace and item - for example, if you have selected an activity for a State transition and you start to do the same for another transition, the dialog opens to the activity you previously selected
- If the context is totally dissimilar, the dialog opens with the Namespace <any> and a collapsed model hierarchy
- If the available items do not meet your requirements, you can create a new item and define the appropriate properties - click on the **Add New** button; the appropriate Add <Item> dialog displays, on which you define the required item
- The **Add New** button is not always available, depending on the context and the type of item being searched for

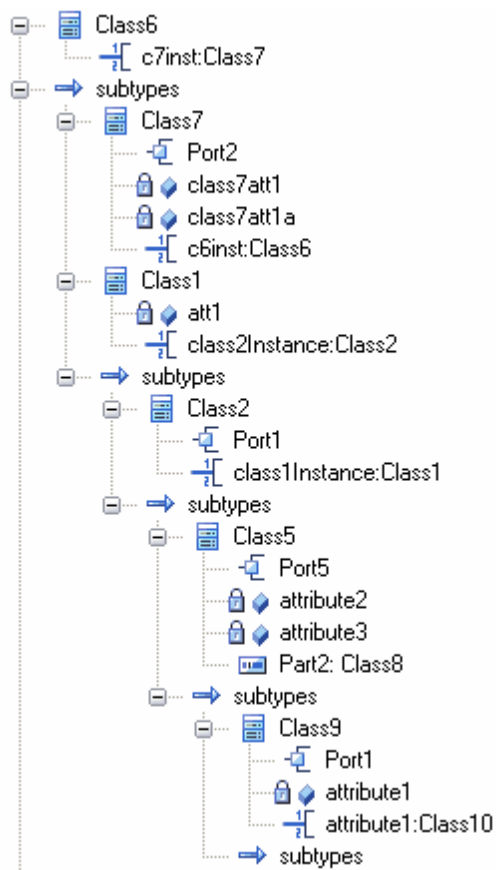
**5.6.4.5.1 Select Property Dialog**

The Select Property dialog is a specific instance of the Select <Item> dialog; it is used to select Ports, Parts, Attributes and Association Ends as *redefined* or *subsetting* properties, from a *hierarchical list* of Classes and their contained properties in the model. For example, consider the section of the model below:

**Example**



This would be represented in the dialog as follows:



Locate and click on the required object to select it, then click on the **OK** button (to select several objects at once, press and hold ( **Ctrl** ) while you click on each object).

#### Notes

- AssociationEnds should be owned by the Class to be listed in the dialog

#### Learn more

- [Select <Item> Dialog](#)<sup>[994]</sup>

#### 5.6.4.5.2 Set Feature Dialog

The Set Feature dialog is the Set Operation dialog used to change the operation represented by an Action on a diagram.

As the Set Operation or Set Attribute dialog, it is also used to set the *Value* operation or attribute for Tagged Values of type RefGUID or for the target of a hyperlink in a diagram.

#### Locate and select a value operation or attribute

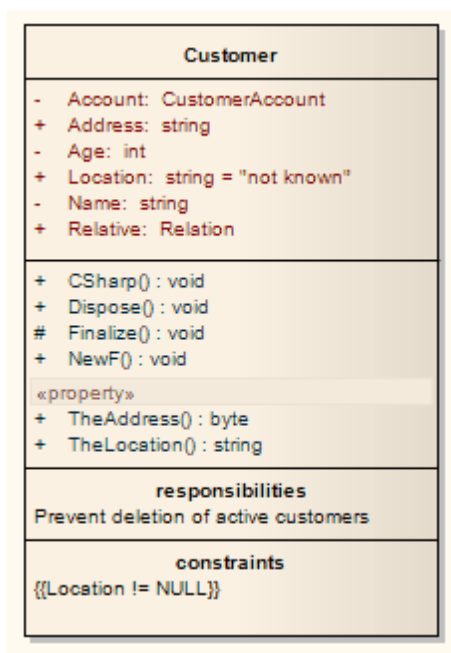
Step	Action	See also
1	The Set Operation (or Set Attribute) dialog displays, with the model hierarchy opened at the point at which you selected the original operation or attribute.	
2	If required, in the <b>In Namespace</b> field, click on the drop-down arrow and select another model that contains the required operation or attribute.  The package hierarchy for that model displays.	
3	Browse through the hierarchy, or use the Search tab to locate the required operation or attribute, then double-click on the item to select it.	<a href="#">Select &lt;Item&gt; Dialog</a> <sup>[994]</sup>

#### Learn more

- [Class Operations In Diagrams](#)<sup>[1275]</sup>
- [Pre-defined Structure Types - RefGUID](#)<sup>[1624]</sup>
- [Hyperlinks](#)<sup>[2002]</sup>

### 5.6.5 Attributes

To represent the properties or internal data elements of an element, you create features of the element called **attributes**. Not all element types support attributes, and others have restrictions; for example, attributes of Interfaces must have **Public** scope. In the Project Browser, elements with attributes (typically Classes) have their attributes listed under the element name, each preceded by a blue box (◆). When depicted in diagrams, the attributes are listed in colored text in the first properties compartment of the element - the default color is red (for example, **Age : int**). In the following Customer Class example, the customer **Name** and **Address** are attributes.



Attributes have several important characteristics, such as **type**, **scope** (visibility), **derivation** and **notes**.

**Access** **Right-click element in diagram/Project Browser | Attributes (F9)**

### Maintain Attributes

Action	Detail	See also
<b>Create and Modify Attributes</b>	On the <Element name> Attributes dialog, you create and modify attributes on the four dialog pages.	<a href="#">General Properties of Attributes</a> <sup>[1001]</sup> <a href="#">Attributes Dialog - Detail</a> <sup>[1003]</sup> <a href="#">Attributes Dialog - Constraints</a> <sup>[1004]</sup> <a href="#">Attribute Tagged Values</a> <sup>[1005]</sup>
<b>Move and Copy Attributes</b>	You can copy or move existing attributes between elements, if the target element also supports attributes.	<a href="#">Copy Features Between Elements</a> <sup>[932]</sup> <a href="#">Move Features Between Elements</a> <sup>[933]</sup> <a href="#">The Element Browser</a> <sup>[989]</sup>
<b>Define Properties</b>	You can create property implementations specific to your coding languages for each attribute.	<a href="#">Create Properties</a> <sup>[1006]</sup>
<b>Identify Inherited Attributes</b>	The attributes of an element can be inherited from one or more successive <b>parent</b> elements in the model. On a diagram, you can show which attributes are inherited from which 'owning' element in the hierarchy.	<a href="#">Display Inherited Attributes</a> <sup>[1007]</sup>

### Notes

- If you make changes and do not save them, the **Cancel** button prompts you to save or cancel the changes, whilst the **Close** button closes the dialog immediately and does not save the changes
- If you are creating many attributes, go to the Attribute/Operations page of the Options dialog (**Tools | Options | Source Code Engineering | Attribute/Operations**) and **de-select** the **After save, re-select edited item** checkbox; now, when you create an attribute and click on the **Save** button, the dialog fields clear ready for you to enter the details of the next attribute - this helps you when you want to create attributes quickly and might not necessarily want to fully define each one as you create it
- If the parent element provides source or target roles for a connector, the connector can be attached to a specific attribute (feature)



- An alternative method of creating relationships between elements and attributes is to drag an attribute onto an Activity diagram, which generates an Object element named after the attribute; you then create relationships to and from this Object element

#### Learn more



- [Options - Attribute/Operations](#) <sup>[2253]</sup>
- [Connect to Element Feature](#) <sup>[1110]</sup>
- [Create Object From Attribute](#) <sup>[1008]</sup>

### 5.6.5.1 General Properties of Attributes


This topic describes the fields and options of the General page of the Attributes dialog.

#### Reference

Field	Usage	See also
<b>Name</b>	Display the name of the attribute; for a new attribute, type the name (with no spaces).	
<b>Type</b>	<p>Display the attribute's type; if necessary, either:</p> <ul style="list-style-type: none"> <li>• Click on the drop-down arrow and select a different type, or</li> <li>• Click on the ( ... ) button to open the Select &lt;Item&gt; dialog and select or define a different attribute classifier type that might not be in the <b>Type</b> drop-down list</li> </ul> <p>The type can be defined by the code language (data type) or by a classifier element; when you click on the drop-down arrow, the first set of values in the list provides the data types, and the second (longer) set provides the possible classifiers.</p> <p>To add new code language data types that can be displayed in this list, see the <i>Data Types</i> topic.</p>	<a href="#">Data Types</a> <sup>[1171]</sup> <a href="#">Instance Classifier</a> <sup>[994]</sup>
<b>Initial Value</b>	Display an optional initial value; if necessary, type in or browse for a new initial value.	
<b>Stereotype</b>	<p>Define the optional stereotype of the attribute; if necessary, either:</p> <ul style="list-style-type: none"> <li>• Type a different stereotype name</li> <li>• Click on the drop-down arrow and select a stereotype, or</li> <li>• Click on the ( ... ) button and browse for the stereotype</li> </ul>	
<b>Alias</b>	Display an optional alias for the attribute; if necessary, type in a new alias.	
<b>Scope</b>	Define the attribute as <b>Public</b> , <b>Protected</b> , <b>Private</b> or <b>Package</b> ; if necessary, click on the drop-down arrow and select a different	

Field	Usage	See also
	scope.	
<b>Static</b>	Indicate that the attribute is a static member.	
<b>Const</b>	Indicate that the attribute is a constant.	
<b>Is Literal</b>	<p>(For Enumeration elements) Defaults to selected, to define the attribute as an enumeration literal.</p> <p>Deselect the checkbox to define the attribute as a normal element attribute.</p> <p>In the <i>Attributes</i> compartment on the diagram, the enumeration literals are listed separately, above the normal attributes (ensure that the <b>Stereotype</b> field for the normal attribute is not set to <b>enum</b>).</p>	
<b>Containment</b>	Define the containment type ( <b>By Reference</b> , <b>By Value</b> or <b>Not Specified</b> ); if necessary, click on the drop-down arrow and select a different containment type.	
<b>Property</b>	Indicate that the attribute has automatic property creation; when you select this checkbox, the Create Property Implementation dialog displays.	<a href="#">Create Properties</a> 
<b>Notes</b>	Enter any free text notes associated with the attribute; you can format the notes text using the Notes toolbar at the top of the field.	<a href="#">Notes Toolbar</a> 

### Topics

Topic	Detail	See also
<b>Usage</b>	<p>To change the position of a selected attribute in the list at the bottom of the dialog, click on the <b>Scroll Up</b> or <b>Scroll Down</b> (hand) buttons.</p> <p>To review an existing attribute, click on the attribute name in the list.</p> <p>You can re-organize and filter the information shown for each attribute, using the List Header facilities for reported information.</p> <p>To delete an existing attribute, click on the attribute name in the list and click on the <b>Delete</b> button.</p> <p>To create a new attribute, either:</p> <ul style="list-style-type: none"> <li>Click on the <b>New</b> button to clear the fields for new information, or</li> <li>Click on an existing attribute name in the list, and click on the <b>Copy</b> button; a prompt displays for you to enter the name of the copy</li> </ul>	<a href="#">List Header</a> 

Topic	Detail	See also
	If you have changed the attribute details, click on the <b>Save</b> button to save the changes.	

#### Notes



- By default, the attributes are listed in alphabetical order; before changing this sequence, you must deselect the **Sort Features Alphabetically** checkbox on the Objects page of the Options dialog (**Tools | Options | Objects**)

### 5.6.5.2 Attributes Dialog - Detail

This topic describes the fields and options of the Detail page of the Attributes dialog.

#### Reference

Field	Action	See also
<b>Multiplicity</b>		
<b>Lower bound</b>	Define a lower limit to the number of elements allowed in the collection.	
<b>Upper bound</b>	Define an upper limit to the number of elements allowed in the collection.	
<b>Allow Duplicates</b>	Indicate that duplicates are allowed.  This field maps to the UML property <i>isUnique</i> , with a value <b>FALSE</b> .	
<b>Ordered Multiplicity</b>	Indicate that the collection is ordered.	
<b>Redefined Property</b>	Review the redefined properties for the attribute.  You can add redefined properties by clicking on the <b>Add</b> button to display the Select Property dialog.	<a href="#">Select Property dialog</a> <sup>[996]</sup>
<b>Subsetted Property</b>	Review the subsetted properties for the attribute.  You can add subsetted properties by clicking on the <b>Add</b> button to display the Select Property dialog.	

Field	Action	See also
<b>Collection</b>		
<b>Attribute is a Collection</b>	Indicate that the attribute is a collection (array), so that it can contain multiple concurrent values rather than a single value.	
<b>Container Type</b>	Enter the name of the container type.	
<b>Other</b>		
<b>Transient</b>	(For Java code) indicate that the attribute can change regardless of what the code is performing.	
<b>Derived</b>	Indicate that the attribute is derived from one or more other attributes; for example, full name can be derived from the <i>last name</i> and <i>first name</i> attributes.	
<b>isID</b>	When selected ( <b>true</b> ), enables the attribute to be used to uniquely identify an instance of the containing Class. Defaults to unselected ( <b>false</b> ).	
<b>Qualifiers</b>	Click on this button to add Qualifiers to the attribute. The Qualifiers dialog displays.	<a href="#">Qualifiers</a>  <a href="#">Qualifiers dialog</a> 
<b>Save</b>	Click on this button to save changes to this page.	

### 5.6.5.3 Attributes Dialog - Constraints

Attributes can have constraints associated with them; typically these indicate such things as maximum value, minimum value and length of field.

These constraints are managed on the Constraints page of the Attributes dialog.

#### Topics

Topic	Detail	See also
<b>Usage</b>	Select the Constraints page to define the constraints. <ul style="list-style-type: none"> <li>To review an existing constraint, click on the constraint name in the panel at the bottom of the dialog</li> <li>To delete an existing constraint, click on the constraint name in the panel and click on the <b>Delete</b> button</li> </ul>	

Topic	Detail	See also
	<ul style="list-style-type: none"><li>To create a new constraint, click on the <b>New</b> button</li></ul> <p>Review, edit or complete the fields as indicated in the table below.</p> <p>If you have created or edited the data, click on the <b>Save</b> button to save the changes.</p>	

#### Reference


Field	Usage	See also
<b>Constraint</b>	Type in the constraint name.	
<b>Type</b>	Type in the constraint type.	
<b>(Notes)</b>	Type in any comments or notes concerning the constraint.	

#### 5.6.5.4 Attribute Tagged Values

You can define Tagged Values for an attribute. Tagged Values are a convenient means of extending the properties a model element supports. This in turn can be used by code generators and other utilities to transform UML models into other forms.

#### How to

To add a Tagged Value to an attribute

Step	Action	See also
1	Click on the Tagged Values tab of the Attribute Properties dialog.	
2	Click on the <b>New tag</b> button (  ). The Tagged Value dialog displays.	
3	In the <b>Tag</b> field, type the tag name or click on the drop-down arrow and select a defined tag.	
4	If appropriate, in the <b>Value</b> field type a specific value for the tag.	
5	Click on the <b>OK</b> button to confirm the operation.	

Step	Action	See also
	The tag name and value are displayed under the attribute in the Tagged Values tab.	

### Notes

- Tagged Values are supported for attributes, operations, objects and connectors
- You can define custom tags by creating a Custom Tagged Value Type


### Learn more

- [Creating a Custom Tagged Value Type](#) 

## 5.6.5.5 Create Properties

Enterprise Architect has capabilities for automatically creating properties in various languages. Property creation is controlled from the General page of the Attributes dialog.

### Topics

Topic	Detail	See also
<b>Usage</b>	<p>On the General page, select the <b>Property</b> checkbox.</p> <p>The Create Property Implementation dialog immediately displays.</p> <p>The Language panel defaults to the Class language; however, you can change this and generate the properties for any language, including Action Script.</p> <p>Each language has slightly different syntax and generates slightly different results</p> <ul style="list-style-type: none"> <li>• Java and C++ generate get and set functions</li> <li>• C# and VB.Net create property functions</li> <li>• Delphi creates get and set functions as well as a specialized Delphi property Tagged Value</li> </ul> <p>Type in the required details and click on the <b>OK</b> button.</p> <p>Enterprise Architect generates the required operations and properties to comply with the selected language.</p> <p>Note that get and set functions are stereotypes with «<i>property get</i>» «<i>property set</i>» making it easy to recognize property functions; you can also hide these specialized functions by deselecting the <b>Property Methods</b> checkbox in the Features tab of the Diagram Properties dialog for a specific diagram (select the <b>Diagram   Properties</b> menu option). This makes it easier to view a Class, uncluttered by many <i>get</i> and <i>set</i> methods.</p> <p>Note that for Delphi you must enable the Tagged Values compartment to see the generated properties.</p>	<a href="#">Compartments</a> 

Topic	Detail	See also

### 5.6.5.6 Display Inherited Attributes

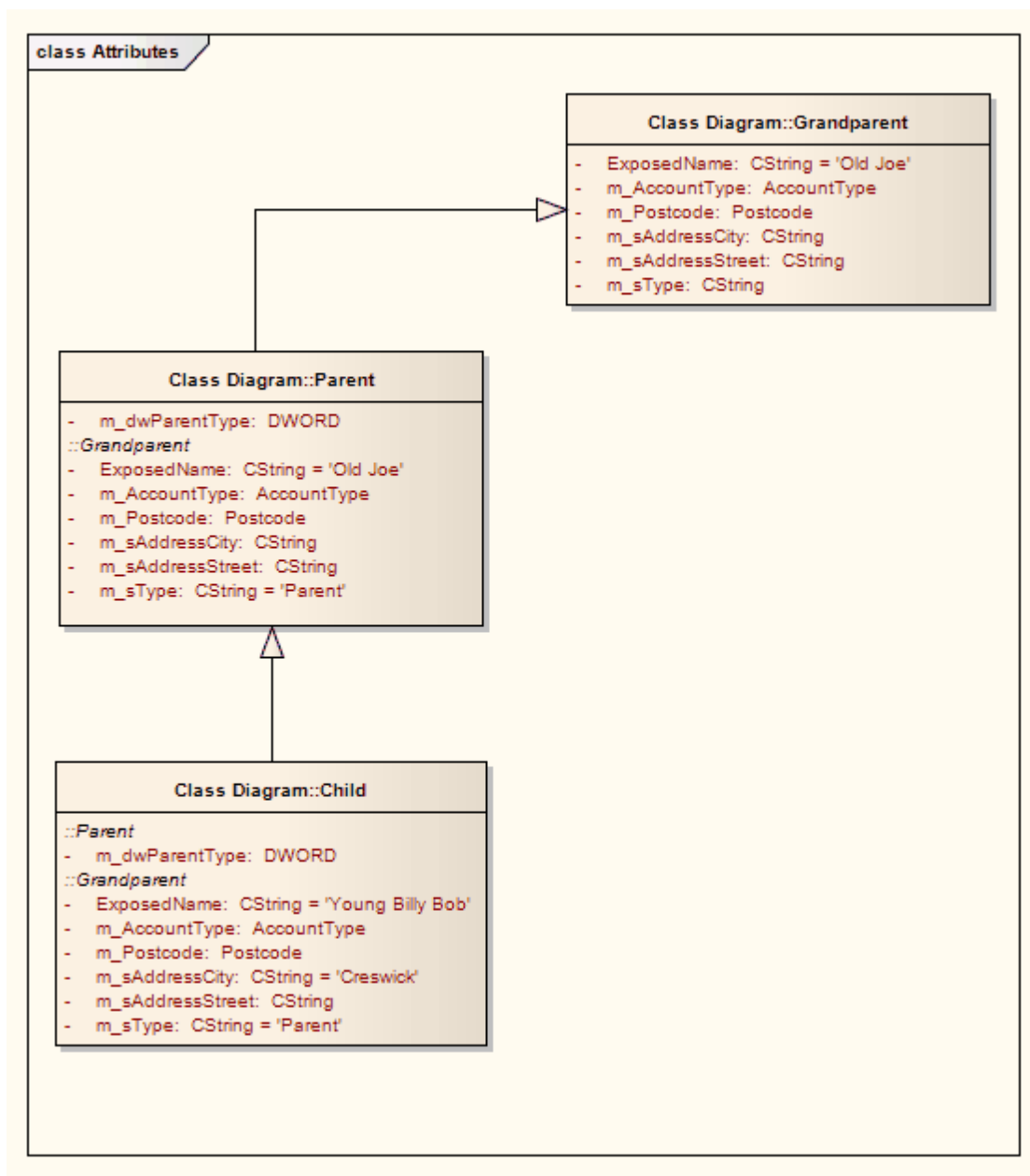
When displaying a Class with attributes in a diagram, you can also show the inherited attributes from all parents in the elements type hierarchy (ancestors).

**Access**    **Right-Click element (on diagram) context menu | Feature Visibility (Ctrl+Shift+Y)**

#### Topics

Topic	Detail	See also
<b>Override Attribute Initializers</b>	<p>Note that for elements that have attributes you can also override an inherited attribute's initial value, using the element context menu option <b>Advanced   Override Attribute Initializers</b> to display the Override Attribute Initializers dialog.</p> <p>In this dialog, select the variable name and enter a new initial value.</p> <p>If required, type a note in the <b>Note</b> field.</p> <p>When you display inherited attributes, Enterprise Architect merges the list of attributes from all ancestors and merges the attribute initializers, so that the final child Class displays the correct attribute set and initial values.</p>	

#### Example



#### Learn more

- [Feature Visibility](#)<sup>[845]</sup>

#### 5.6.5.7 Create Object From Attribute

If you drag an attribute from the Project Browser onto an *Activity* diagram, the attribute generates an Object element of the same name.

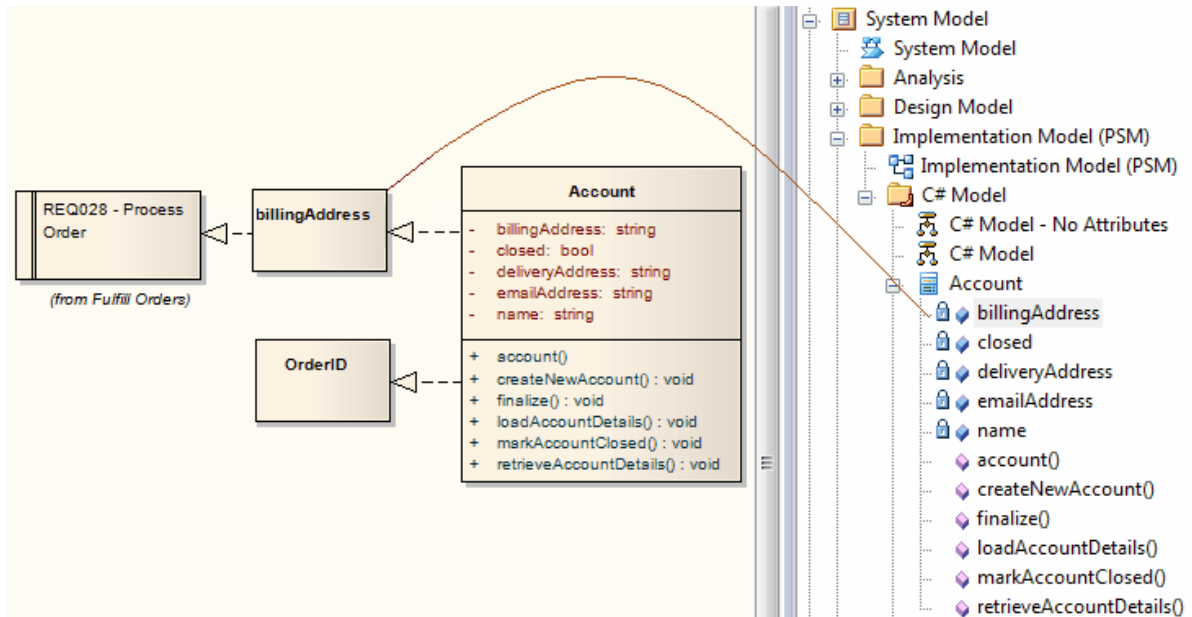
This is very useful for creating connectors between elements and specific attributes. For example, a Class element of stereotype *table* defines its fields as attributes; Requirement elements that define requirements for those fields can then be linked to the appropriate table fields via the attribute Object elements.



### Example

In the following diagram, the *billingAddress* Object was generated by dragging the *billingAddress* attribute from the Account Class in the Project Browser onto the diagram.

The user then created *Realize* relationships between the *Account* element and the *billingAddress* element, and between the *billingAddress* element and the *REQ028 Requirement* element.



### Notes

- You can also create this relationship more directly by linking an attribute in an element to another element, or linking two attributes in different elements

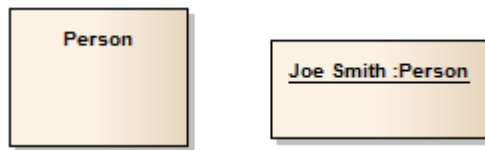
### Learn more

- [Connect to Element Feature](#) <sup>1110</sup>

## 5.6.6 Classifiers and Instances

Some types of element (such as Classes) model **classifications**, which provide a specification of an abstract concept. For example, the concept of a *Building* can be represented by a set of Classes that define types of building such as *Bungalow*, *Miner's Cottage*, *Office Block*, *Shop* and so on. Other types of element (such as Objects) model **instances** of such classifications, such as *23 High Street Shop* or *45 Village Green Cottage*. The instance elements represent actual objects in a run-time scenario, based on the specification of the classifier element. For example:

- An **Object element** represents a man called *Joe Smith*, as a specific instance of the concept of 'a person' defined by the **Class element** *Person*; the object is rendered as a rectangle showing the name of the classifier - if you name the object it becomes *<ObjectName> :<ClassifierName>* or, in this example, *Joe Smith :Person*



If you develop a *Person* Class with attributes such as *Age*, *Name*, *Address* and *Sex*, and functions such as *GetAge* and *GetName*, then when you derive an object from that Class the object takes on all the *Person* Class behavior and state (as well as inherited state and behavior from *Person's* **parent elements**, if any).

Associating an instance with a classifier greatly increases the descriptive power of the model in capturing the functionality and responsibility of objects and their associated state at run-time. This is a powerful means of moving your model from the analysis phase into detailed design.

### Notes

- An object is an instance of a classifier at runtime, so the object's **features** are actually those of the classifier; in the context menu for a classified object, if you select the **Attributes** or **Operations** menu options, the Attributes or Operations dialog displays for the **classifier**, not the **object**.
- Similarly, if you have set the classifier for an object in a **Sequence diagram**, when you add a **message** and display its Message Properties dialog, the **Message** drop-down list is drawn from the operations of the target object's **classifier**, not from the object itself

In this way you can associate Sequence diagram objects with Classes and use the defined behavior of the **Class** to model actual behavior at run time; the same concept applies to Messages for a **State Flow** connector

- In the Message dialog, you can also select to include messages defined in the **inheritance hierarchy** of the classifier for the target object

### Learn more

- [Classify an Object](#) <sup>[1010]</sup>
- [Message \(Sequence Diagram\)](#) <sup>[1418]</sup>

#### 5.6.6.1 Classify an Object

Objects can be created and classified in a number of ways. You can define a classifier first and then create instances of that classifier. Alternatively, you might develop several separate objects that turn out to have a similar specification, which you then formally define as a classifier. You would then classify the existing objects as instances of that new classifier.

#### Methods of Classifying an object

Method	Detail	See also
<b>Drop Classifier Onto Diagram</b>	<p>Drag an existing classifier element from the Project Browser onto a diagram. Depending on how you configure the Paste Element dialog, you can <b>create</b> an associated object as an instance, or the instance is <b>automatically generated</b>.</p> <p>If your classifier contains Property elements, you also have the option of adding some or all of these properties to the instance as 'slots'.</p>	<p><a href="#">Paste from Project Browser</a> <sup>[833]</sup></p> <p><a href="#">Add Property Slots to</a></p>

Method	Detail	See also
		<a href="#">Instance</a> <sup>[1011]</sup>
<b>Classify Existing Object</b>	<p>(Where both object and classifier already exist, separately.)</p> <p>On a diagram, right-click on the object and select the <b>Advanced   Instance Classifier</b> context menu option. (Or click on the object and press ( <b>Ctrl + L</b> ).)</p> <p>The Select Classifier dialog displays. Browse or search through the model for the classifier element to associate with the object, and double-click on it.</p> <p>The object is associated with the classifier and displays the naming convention described above.</p>	<a href="#">Select &lt;Item&gt; Dialog</a> <sup>[994]</sup>
<b>Classify new or existing object from classifier Associations</b>	<p>A classifier is usually a component of a model structure, and can have Associations with other classifiers.</p> <p>If an object is classified, any other objects related to that object can be classified by the related elements of the classifier. For example:</p> <ul style="list-style-type: none"> <li>• Class A is associated with Class B and Class C</li> <li>• Object 1 is classified by Class A</li> <li>• If you create an Association between Object 1 and Object 2, the Choose an Association Connector dialog offers the option of setting either Class B or Class C as the classifier of Object 2</li> </ul>	<a href="#">Classify Object Via Classifier Associations</a> <sup>[1012]</sup>

**Learn more**

- [Classifiers and Instances](#) <sup>[1009]</sup>

**5.6.6.1.1 Add Property Slots to Instance**

A classifier element can have a range of characteristics that are automatically applied to an instance based on that classifier. You can also, optionally, add any **Property elements** (Ports, Parts and Expose Interfaces) on the classifier to the instance, as 'slots' for the definition of each Property. This provides some flexibility in creating variations of the design defined in the classifier. For example, models of car might have doors of the same specification, but in a two-door or four-door configuration.

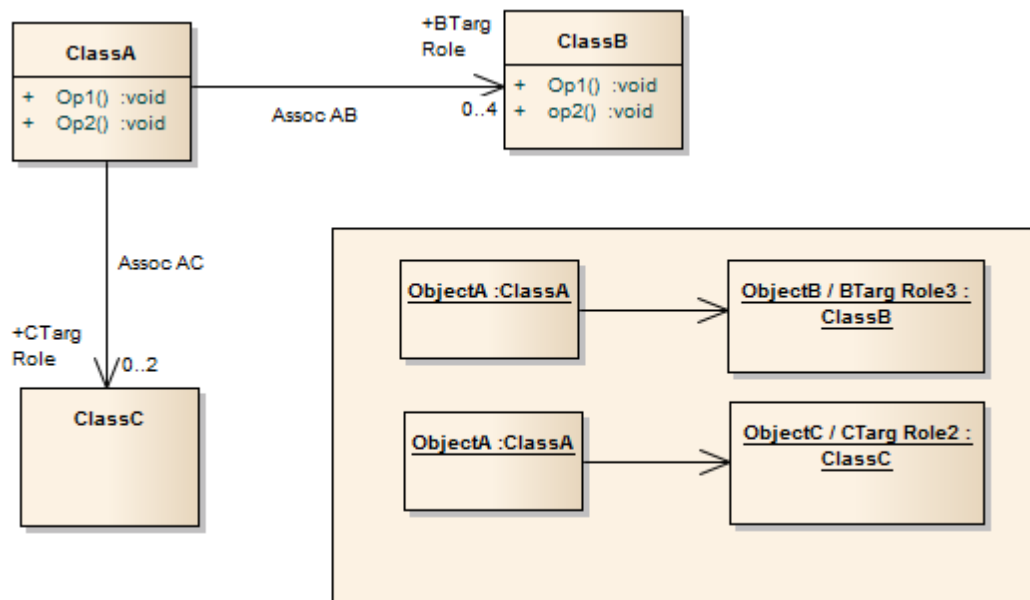
**Add classifier Property elements to an instance**

Step	Action	See also
<b>1</b>	Create and save the classifier element with the configuration of Property elements you need.	<a href="#">Part</a> <sup>[1383]</sup> <a href="#">Port</a> <sup>[1384]</sup> <a href="#">Expose Interface</a> <sup>[1375]</sup>
<b>2</b>	Create or assign an instance of the classifier on your work diagram.	<a href="#">Classifiers and Instances</a> <sup>[1009]</sup>

Step	Action	See also
3	<p>Right-click on the instance and select the <b>Structural Elements</b> context menu option.</p> <p>The Structural Elements dialog displays, listing the Property elements defined for the classifier.</p>	<a href="#">Manage Structural Elements</a> <sup>[93]</sup>
4	<p>Select the check box against each Property to reproduce in the instance element, and click on the <b>Close</b> button.</p> <p>Each Property element is shown in the top left of the instance; drag the elements into the arrangement you want. The Property element name within the instance is prefixed with a / character.</p>	

#### 5.6.6.1.2 Classify Object Via Classifier Associations

When building an instance model from a predefined Class model, Enterprise Architect can assist in setting classifiers and role names for objects involved in a relationship. For example, if ClassA has association relationships with ClassB and ClassC, you can create an instance of ClassA (ObjectA), then draw an association to a new unclassified object and have the choice of selecting ClassB or ClassC as the classifiers of the new object.



**Access** Click on the Classified object on a diagram and:

Drag an Association to an existing, unclassified object or  
 Use the Quick Linker arrow to create a Link to an existing or new unclassified object, or  
 Right-click on existing Association | Advanced | Instance Classifier (Ctrl+L)

**Choose an Association Connector dialog**

Field/Button	Detail	See also
<b>Classifier Name</b>	Displays the name (if defined) of each Association connector issuing from the source object's classifier.  If the connector has no name, the value <b>&lt;anonymous&gt;</b> displays.	
<b>Association</b>	Represents the source and target classifier elements in each Association connector issuing from the source object's classifier.  If you select one of these, the target classifier becomes the classifier for the target object element.  Select <b>&lt;none&gt;</b> if you want to create the Association, but not based on the classifier relationships.	
<b>&lt;Source Object Name&gt; Role</b>	Identifies any Roles defined in the Association for the source classifier element, according to the source role multiplicity setting; if you select one of these roles, this will become the specific Role for the source object, as shown on the name of the element on the diagram.  If the multiplicity is not set, or the upper bound is * (asterisk), just the role name displays in the list.	
<b>&lt;Target Object Name&gt; Role</b>	Identifies any Roles defined in the Association for the target classifier, according to the target role multiplicity setting; if you select one of these roles, it will become the specific Role for the target object as shown on the name of the element in the diagram (and on the Binding page of the Association connector Properties dialog).  If the multiplicity is not set, or the upper bound is * (asterisk), just the role name displays in the list.	
<b>OK</b>	Click on the Association and source and target roles, then click on this button to: <ul style="list-style-type: none"> <li>• Create the Association between the two objects based on the relationship between the two classifiers, and</li> <li>• Make the target classifier the classifier for the target object</li> </ul> If you selected the <b>&lt;none&gt;</b> entry, a simple Association is created and the target object is unclassified.	
<b>Cancel</b>	Cancel the dialog without changing any classifier settings.	

**Notes**

- When creating a new association or object, the **Properties** dialog might appear before the **Choose an Association Connector** dialog; complete any Properties fields as required, and click on the **OK** button

Learn more

- [Classifiers and Instances](#) <sup>[1009]</sup>
- [Using Classifiers](#) <sup>[1010]</sup>

## 5.6.7 Behavior

Enterprise Architect enables you to define an element's behavior through the element's operations and parameters. You can also define the behavior of more specific behavioral elements such as Activities, Interactions, Actions and Interaction Occurrences, through the Behavior and Call tabs of the element Properties dialogs.

Learn more

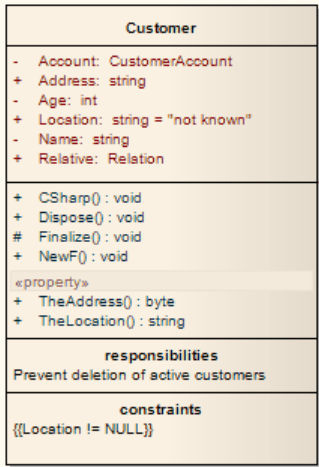
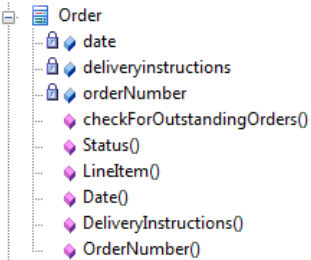
- [Operations](#) <sup>[1014]</sup>
- [Interactions and Activities](#) <sup>[1026]</sup>
- [Behavior Calls](#) <sup>[1026]</sup>
- [Behavior Parameters](#) <sup>[1028]</sup>
- [Behavior Call Arguments](#) <sup>[1027]</sup>

### 5.6.7.1 Operations

Operations are features of a Class or other element that represent the behavior or services an element supports. For a Customer Class, *UpdateCustomerName* and *GetCustomerAddress* can be operations. Operations have several important characteristics, such as type, scope (visibility), static, abstract and notes.

Topics

Images	Topic	Detail	See also
	<b>Create and Modify Element Operations</b>	<p>In the Diagram view or Project Browser, either:</p> <ul style="list-style-type: none"> <li>• Right-click on the element to be edited, and from the context menu select the <b>Operations</b> menu option, or</li> <li>• Click on the element and press ( <b>F10</b> )</li> </ul> <p>The &lt;Element name&gt; Operations dialog displays.</p> <p>You can also copy or move existing Operations between elements.</p> <p>These facilities are only available if the target element supports operations.</p>	<p><a href="#">General Properties of Operations</a> <sup>[1015]</sup></p> <p><a href="#">Copy Features Between Elements</a> <sup>[932]</sup></p> <p><a href="#">Move Features Between Elements</a> <sup>[933]</sup></p> <p><a href="#">The Element Browser</a> <sup>[989]</sup></p>

Images	Topic	Detail	See also
 <p>The diagram shows a class named <b>Customer</b>. It has several attributes: <code>- Account: CustomerAccount</code>, <code>+ Address: string</code>, <code>- Age: int</code>, <code>+ Location: string = "not known"</code>, <code>- Name: string</code>, and <code>+ Relative: Relation</code>. It also has several operations: <code>+ CSharp() : void</code>, <code>+ Dispose() : void</code>, <code># Finalize() : void</code>, and <code>+ NewF() : void</code>. There is a property section with <code>+ TheAddress() : byte</code> and <code>+ TheLocation() : string</code>. A responsibilities section states "Prevent deletion of active customers". A constraints section shows <code>{{[Location != NULL]}}</code>.</p>	<b>How Operations Appear in Diagrams</b>	<p>In diagrams, elements with operations (typically Classes) list the operations in the second properties compartment, in colored text, as shown on the left. The default color is dark green; for example:</p> <p><code>Finalize() : void</code></p> <p>Some characteristics display in shorthand form; for example, <i>static</i> displays as \$, <i>abstract</i> as *.</p>	
 <p>The Project Browser shows a tree structure. Under the <b>Order</b> class, there are several items: <code>date</code>, <code>deliveryinstructions</code>, <code>orderNumber</code>, <code>checkForOutstandingOrders()</code>, <code>Status()</code>, <code>LineItem()</code>, <code>Date()</code>, <code>DeliveryInstructions()</code>, and <code>OrderNumber()</code>. Each operation is preceded by a pink diamond icon.</p>	<b>Operations in the Project Browser</b>	<p>In the Project Browser, Classes with operations have the operations listed beneath the Class name, each preceded by a pink box (◆).</p> <p>Right-click on an operation and select the <b>Operation Properties</b> context menu option, to open the Operations dialog and edit details for the feature.</p> <p>From the Project Browser, you can drag operations onto new elements to give them the same operations.</p>	

### Notes

- If the parent element provides source or target roles for a connector, the connector can be attached to a specific operation

### Learn more

- [Behavior Parameters](#) <sup>[1028]</sup>
- [Connect to Element Feature](#) <sup>[1110]</sup>

#### 5.6.7.1.1 General Properties of Operations

**Access** In diagram or Project Browser:

Click on element | Element | Operations (F10) or  
Right-click on element | Features & Properties | Operations

The Operations dialog has seven pages:



- General
- Parameters

- Behavior
- Advanced
- Tagged Values
- Pre-conditions and Post conditions (that is, Constraints)

The General page of the Operations dialog enables you to define new operations and set the most common properties, including name, access type and return.

Field/Button	Usage	See also
<b>Name</b>	Display the selected operation name.	
<b>Parameters</b>	Display the parameter list. See the <i>Parameters Dialog</i> topic for information regarding what this string can contain.	<a href="#">Define Parameters</a> <sup>[1028]</sup>
<b>Edit Parameters</b>	Open the Parameters dialog.	
<b>Return Type</b>	(This option is not shown for <i>State</i> or <i>State Machine</i> elements.) Display the data type returned by the operation.  The type can be defined by the code language (data type) or by a classifier element; when you click on the drop-down arrow, the set of values in the list provides the appropriate data types.  To select or define possible classifiers, either click on the <b>Select Type</b> option in the list, or click on the ( ... ) (Select) button to display the Select <Item> dialog.  To add new code language data types that can be displayed in this list, see the <i>Data Types</i> topic.	<a href="#">Instance Classifier</a> <sup>[994]</sup> <a href="#">Data Types</a> <sup>[1171]</sup>
( ... ) (Return Type Browse button)	Open the Select <Item> dialog to select the operation return type. (Not shown for <i>State</i> or <i>State Machine</i> elements.)	<a href="#">Instance Classifier</a> <sup>[994]</sup>
<b>Action</b>	Define the action of the operation: <b>do</b> , <b>exit</b> or <b>entry</b> . (Displayed for operations of <i>State</i> or <i>State Machine</i> elements.)	<a href="#">State</a> <sup>[1329]</sup> <a href="#">State Machine</a> <sup>[1338]</sup>
<b>Scope</b>	Click on the drop-down arrow and select one of: <ul style="list-style-type: none"> <li>• <b>Public</b></li> <li>• <b>Protected</b></li> <li>• <b>Private</b></li> <li>• <b>Package</b></li> </ul>	



Field/Button	Usage	See also
<b>Stereotype</b>	Specify an optional stereotype for this operation.	
<b>Alias</b>	Define an optional alias for the operation.	
<b>Notes</b>	Enter free text notes. You can format this text if necessary, using the Notes toolbar at the top of the field.	<a href="#">Notes toolbar</a> 
<b>Virtual</b>	If the operation's language is set to C++, map to the C++ <i>Virtual</i> keyword. Otherwise this option is <i>Abstract</i> , pertaining to an abstract function. (This option is not shown for <i>State</i> or <i>State Machine</i> elements.)	
<b>Static</b>	Indicate that the operation is a static member. (This option is not shown for <i>State</i> or <i>State Machine</i> elements.)	
<b>Up/Down Buttons</b>	Change the order of operations in the list.	
<b>New</b>	Create a new operation.	
<b>Copy</b>	Copy the currently selected operation.	
<b>Save</b>	Save a new operation, or save modified details for existing operation.	
<b>Delete</b>	Delete the currently selected operation.	
<b>Name/ Parameters/ Return Type</b>	List the existing operations for the selected element. You can re-organize and filter the information shown for each operation, using the List Header facilities for reported information.	<a href="#">List Header</a> 

### Notes

- The General page can vary according to the type of element you are adding an operation to; if defining operations for a data modeling table, see the *Indexes*, *Triggers* and *Check Constraints* topics
- If you make changes and do not save them, the **Cancel** button prompts you to confirm or cancel the changes, whilst the **Close** button closes the dialog immediately and does not save the changes
- If you are creating many operations, go to the Attribute/Operations page of the Options dialog (**Tools** |

**Options | Source Code Engineering | Attribute/Operations**) and **de-select** the **After save, re-select edited item** checkbox; now, when you create an operation and click on the **Save** button, the dialog fields clear ready for you to enter the details of the next operation - this helps you when you want to create operations quickly and might not necessarily want to fully define each one as you create it

#### Learn more

- [Operation Parameters](#) <sup>[1029]</sup>
- [Operation Behavior](#) <sup>[1018]</sup>
- [Operation Constraints](#) <sup>[1023]</sup>
- [Advanced Properties of Operations](#) <sup>[1020]</sup>
- [Operation Tagged Values](#) <sup>[1022]</sup>
- [Indexes](#) <sup>[2365]</sup>
- [Triggers](#) <sup>[2368]</sup>
- [Default Constraints](#) <sup>[2371]</sup>
- [Check Constraints](#) <sup>[2370]</sup>

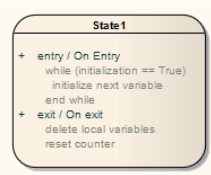
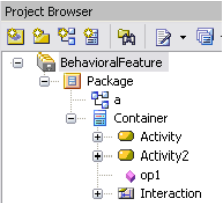
#### 5.6.7.1.2 Operation Behavior

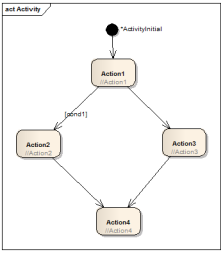
This topic illustrates how to elaborate a method's function in a diagram. The Behavior page of the Operations dialog enables you to enter free text to describe the functionality of an operation. Use pseudo-code, structured English or just a brief description.

You can also use the Behavior page to formally describe a method or State action and have the text appear under the method/action name in a diagram.

**Access** **Project Browser** **Operation context menu | Operation Properties | Behavior**

#### Topics

Images	Topic	Detail	See also
	<b>State Operations - Behaviour</b>	<p>A State's <i>do</i>, <i>entry</i> and <i>exit</i> operations optionally refer to other behaviors such as Activities, Interactions or Operations.</p> <p>Click on the <b>Element</b> button to display the Select Behavior dialog, and locate and select the required behavior.</p>	<a href="#">Select &lt;Item&gt; Dialog</a> <sup>[994]</sup>
	<b>Associate with Behaviour</b>	<p>A Class operation can be associated with a behavior elsewhere in the model; to do this:</p> <ol style="list-style-type: none"> <li>1. Display the operation's Properties dialog..</li> <li>2. Select the Behavior page, click on the <b>Element</b> button and select the required behavior from the Select Behavior dialog; the behavior displays in the <b>Behavior</b> field.</li> </ol> <p>The operation is then set as the specification of the associated behavior.</p> <p>In behavioral code generation, the behavior of the</p>	<a href="#">Generate From Behavioral</a>

Images	Topic	Detail	See also
 <pre> graph TD     Start(( )) --&gt; *ActivityInitial  Action1[Action1]     Action1 --&gt; cond1  Action2[Action2]     Action1 --&gt; Action3[Action3]     Action2 --&gt; Action4[Action4]     Action3 --&gt; Action4 </pre>		<p>associated behavioral element is generated as the <i>operation</i>'s code; in the illustration on the left, <i>Op1</i> is associated with the <i>Activity</i> Activity.</p>	<a href="#">Models</a> <sup>2121</sup>
	<b>Example Code</b>	<p>The generated code for <i>Op1</i> is as follows:</p> <pre> package Package; public class Container {     public Container(){     }      public void finalize() throws Throwable     {      }      public void op1(){         /*Activity element(Activity1)'s behavior         operation(op1)'s code*/         //Action1;         if (cond1)         {             //Action2;         }         else         {             //Action3;         }         //Action4;     }     /*Activity element(Activity1)not rendered     public void Activity2()     {         // behavior is a Activity     }      public void Interaction()     {         // behavior is a Interaction     } } </pre>	

### Learn more

- [Initial Code](#) <sup>1019</sup>

#### 5.6.7.1.2.1 Initial Code

**Access** Project Browser Operation context menu | Operation Properties | Behavior

### Topics

Topic	Detail	See also
<b>Usage</b>	<p>On the Behavior page of the Operations dialog, use the <b>Initial Code</b> field to enter code to be inserted into an operation body when the operation is first generated to file.</p> <p>After this point, forward code generation and synchronization do not replace the existing operation code with the <b>Initial Code</b> field.</p> <p>By default, the <b>Initial Code</b> field also is not imported into the model during reverse engineering, but you can select to import the field by selecting the <b>Include method bodies in model when reverse engineering</b> checkbox on the Options dialog.</p> <p>This field is most useful when combined with Design Patterns; elements within a pattern often require the same stub code.</p> <p>Notice that the language-specific patterns available from <a href="http://www.sparxsystems.com/resources/developers/uml_patterns.html">www.sparxsystems.com/resources/developers/uml_patterns.html</a> include initial code for some of the defined operations; this helps speed up the process of applying patterns from model to implementation.</p> <p>The <b>Initial Code</b> section is also useful for ensuring that the generated code is directly compilable.</p>	<p><a href="#">Options - Attributes/Operations</a> <sup>2253</sup></p> <p><a href="#">Design Patterns</a> <sup>1464</sup></p> <p><a href="http://www.sparxsystems.com/resources/developers/uml_patterns.html">www.sparxsystems.com/resources/developers/uml_patterns.html</a></p>

#### 5.6.7.1.3 Advanced Properties of Operations

In defining the behaviors of the model elements you create, you can also define how the operations translate into methods in code. For this, you set options on the Advanced tab of the Operation Properties dialog.


**Access**    **In diagram or Project Browser:**

**Click on element | Element | Operations (F10) > General | select operation > Advanced, or Right-click on element | Features & Properties | Operations > General | select operation > Advanced**

#### Available Options

Field/Button	Usage	See also
<b>Concurrency</b>	Click on the drop-down arrow and select the concurrency of the operation - <b>Sequential</b> , <b>Guarded</b> or <b>Synchronous</b> .	
<b>Pure</b>	<p>For C++, select this checkbox to indicate that the method statement is pure virtual syntax. For example:</p> <pre>virtual void myFunction( ) = 0;</pre> <p>(This option is not shown for operations of <i>State</i> or <i>State Machine</i> elements.)</p>	
<b>Const</b>	Select this checkbox to indicate that the return type of the method is constant.	

Field/Button	Usage	See also
	(This option is not shown for operations of <i>State</i> or <i>State Machine</i> elements.)	
<b>Is Query</b>	Select this checkbox to indicate that the method does not modify the object.  (This option is not shown for operations of <i>State</i> or <i>State Machine</i> elements.)	
<b>Return Array</b>	Select this checkbox to indicate that the method's returned value is in the form of an array.  (This option is not shown for operations of <i>State</i> or <i>State Machine</i> elements.)	
<b>Synchronized</b>	For Java, select this checkbox to set a code engineering flag that relates to multi threading.  (This option is not shown for operations of <i>State</i> or <i>State Machine</i> elements.)	
<b>Advanced</b>	Select the checkbox for each coding convention to assign as a Tagged Value (corresponding to a coding keyword).	<a href="#">Modeling Conventions</a> [2080]
<b>Redefined Operation</b>	<p>If the parent element is the <b>source</b> of a <b>Generalize</b> connector, and you want the selected operation to redefine an operation in the <b>target</b> element, click on the <b>Add</b> button.</p> <p>The Select Operation dialog displays, showing the operations in the target element. Click on the appropriate operation; if you want to redefine more than one operation in the target element, press ( <b>Ctrl</b> ) as you click on each one.</p> <p>Click on the <b>OK</b> button.</p> <p>A statement displays in the <b>Redefined Operation</b> field of the format:</p> <pre>&lt;Package&gt;::&lt;Target Element&gt;.&lt;Operation&gt;</pre> <p>On the element in the diagram, the Operation compartment shows the complete statement, of the format:</p> <pre>&lt;Operation&gt;:{redefines &lt;Package&gt;::&lt;Target Element&gt;.&lt;Op</pre> <p>If you want to change the redefined operation(s) at any point, again click on the <b>Add</b> button and repeat the process; the original selection is replaced by the new selection.</p>	<a href="#">Generalization</a> [1409]  <a href="#">Select &lt;Item&gt; Dialog</a> [994]
<b>Raised Exception</b>	If you want the code to check for an exception and, if it occurs, skip to the code defined by a Classifier element, you can identify that element in this field.	

Field/Button	Usage	See also
	<p>Click on the <b>Add</b> button. The Select Classifier dialog displays; browse or search through the model hierarchy for the appropriate Classifier.</p> <p>Click on the <b>OK</b> button; the name of the Classifier displays under the <b>Exception</b> heading.</p> <p>If you want to change the Classifier at any point, again click on the <b>Add</b> button and repeat the process; the original selection is replaced by the new selection.</p>	<a href="#">Select &lt;Item&gt; Dialog</a> 
<b>Close</b>	Click on this button to save the field values and close the dialog.	
<b>Cancel</b>	Click on this button to close the dialog without saving the field values.	

#### Learn more

- [General Properties of Operations](#) 

#### 5.6.7.1.4 Operation Tagged Values

Operations can have Tagged Values associated with them. Tagged Values offer a convenient extension mechanism for UML elements, so you can define any tags you like and then assign values to them using this form.

Tagged Values are written to the XML output, and can be input to other third party tools for code generation or other activities.

**Access** [Project Browser Operation context menu](#) | [Operation Properties](#) | [Tagged Values](#)

#### Add a Tagged Value for an operation

Step	Action	See also
1	Click on the Tagged Values page of the operation Properties dialog.	
2	Click on the <b>New Tags</b> button. The Tagged Value dialog displays.	
3	In the <b>Tag</b> field, type the tag name (or select a defined tag from the drop-down list). In the <b>Value</b> field type the initial tag value.	
4	Click on the <b>OK</b> button to confirm the operation.	

Step	Action	See also

#### Notes

- Tagged Values are supported for attributes, operations, objects and connectors
- You can define custom tags using a predefined Tagged Value Type

#### Learn more

- [Create a Custom Tagged Value](#)<sup>[626]</sup>

#### 5.6.7.1.5 Operation Constraints

**Access** **Project Browser Operation context menu | Operation Properties | Constraints**

#### Topics

Topic	Detail	See also
<b>Usage</b>	<p>You can define pre- and post- conditions for an Operation; for each type, give the condition a name and a type, and enter notes.</p> <p>Constraints define the contractual behavior of an operation, what must be true before they are called and what is true after. In this respect they are related to the state model of a Class and can also relate to the guard conditions that apply to a transition.</p>	

#### 5.6.7.1.6 Override Parent Operations

In Enterprise Architect, you can automatically override methods from parent Classes and from realized Interfaces.

**Access** **Element | Advanced | Overrides & Implementations (Ctrl+Shift+O)**

#### Topics

Topic	Detail	See also
<b>Usage</b>	<p>In the Override Operations/Interfaces dialog, check the operations/interfaces to automatically override and click on the <b>OK</b> button.</p> <p>Enterprise Architect generates the equivalent function definitions in your child Class.</p> <p>You can configure Enterprise Architect to display this dialog each time you add</p>	

Topic	Detail	See also
	a Generalization or Realization connector between Classes, and review their possible operations/interfaces to override/implement; to do this, open the Links page of the Options dialog (select the <b>Tools   Options   Links</b> menu option).	

#### 5.6.7.1.7 *Display Inherited Operations*

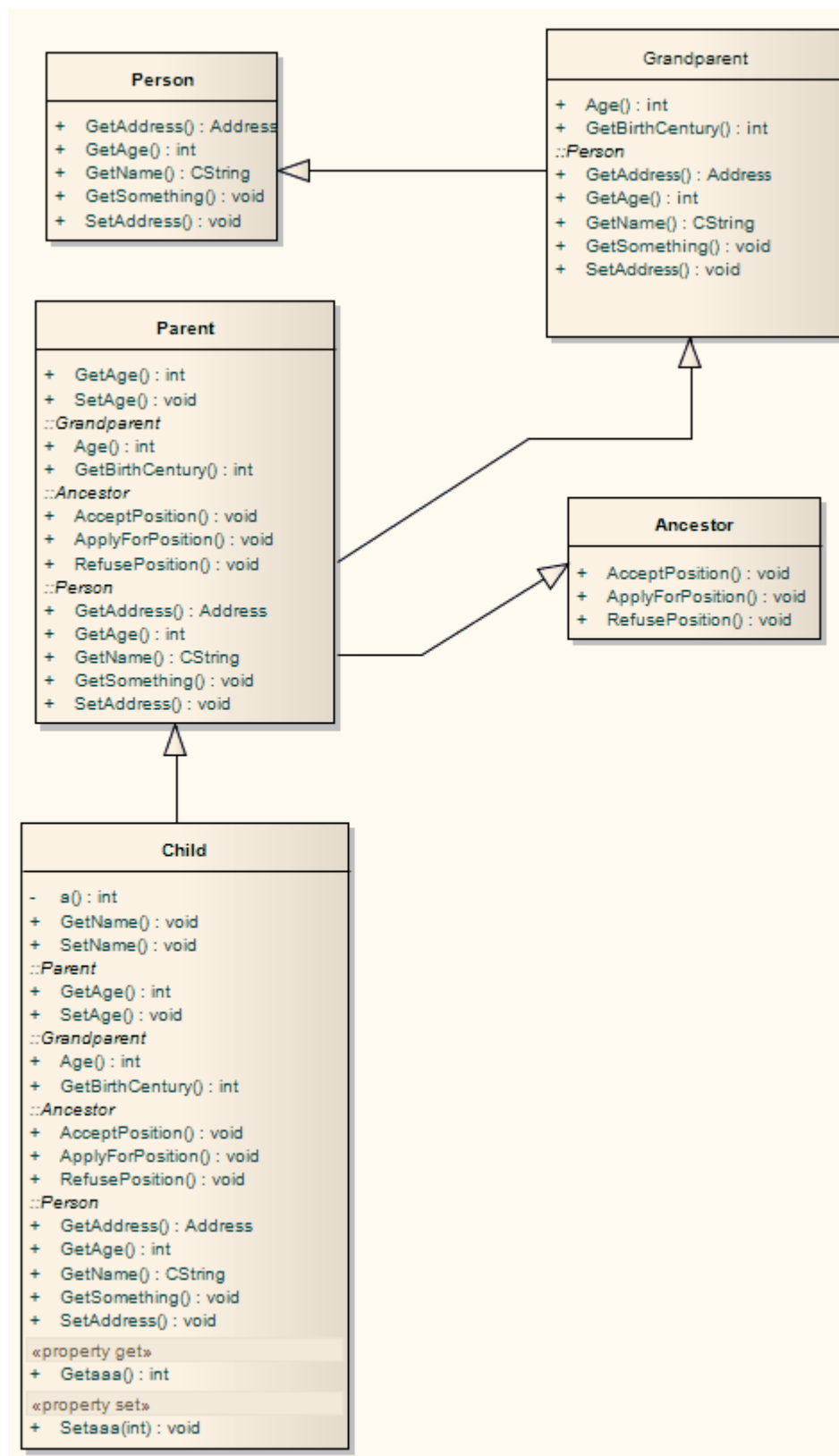
You can configure an element in a diagram to display the complete operation set obtained from all ancestors in the element's type hierarchy, as well as those directly owned.

Access   **Element | Feature Visibility**   ( **Ctrl+Shift+Y** )

#### Example

The following diagram illustrates this behavior when enabled for each element in a simple hierarchy.








### 5.6.7.2 Interactions and Activities

The behavioral aspects of Interactions and Activities are modeled using the Behavior page of the element Properties dialog, which enables you to assign parameters and return types to the elements.

#### Topics

Topic	Detail	See also
<b>Usage</b>	<p>Use the <b>Edit Parameters</b> button to edit an element's parameters. When you create a new parameter using the dialog, it internally creates an Activity Parameter Node for an Activity or an Interaction Parameter for an Interaction. In the <b>Return</b> field, click on the drop-down arrow and select the return type of the element.</p> <p>The <b>Specification</b> field is populated automatically when an operation is associated with the activity as a behavior.</p>	<a href="#">Define Parameters</a>  <sup>[1028]</sup> <a href="#">Activity Parameter Nodes</a>  <sup>[1281]</sup> <a href="#">Operation Behaviour</a>  <sup>[1018]</sup>

### 5.6.7.3 Behavior Calls

A behavior call is the invocation of a behavior. You can represent an invocation with a CallOperation Action (Operation), CallBehavior Action (Activity) or Interaction Occurrence (Interaction) element. You model the properties of the behavior call using the Call tab of the element Properties dialog.





#### Use to

- Edit Arguments
- Re-associate the call with a different behavior
- Synchronize the arguments with the parameters in the associated behavior

#### Topics

Topic	Detail	See also
<b>Usage</b>	<p>Click on the <b>Edit</b> button to create and delete arguments, and relate them to a corresponding parameter in the associated behavior.</p> <p>Click on the ( ... ) (<b>Select Behavior</b>) button to re-associate the invocation with a different behavior or to remove any association with the current behavior.</p> <p>The Interaction Occurrence panel is displayed only for Interaction Occurrence elements. It enables you to enter the return value and attribute of the behavior call.</p>	

#### Learn more

- [Action Types](#)  <sup>[1268]</sup>
- [Interaction Occurrence](#)  <sup>[1317]</sup>
- [Arguments](#)  <sup>[1027]</sup>
- [Associate with Different Behavior](#)  <sup>[1027]</sup>

- [Synchronize Arguments](#) <sup>[1027]</sup>
- [Code Generation - Activity Diagrams](#) <sup>[2134]</sup>

#### 5.6.7.3.1 Associate with Different Behaviors

On the Call tab of the Behavior Call Properties dialog, when you click on the ( ... ) (Select Behavior) button the Select <Item> dialog displays, listing all available behaviors in the model.

##### Topics

Topic	Detail	See also
<b>Usage</b>	<p>Select <b>&lt;none&gt;</b> to remove any existing association between an invocation and a behavior, or select another classifier to re-associate the invocation with a different behavior.</p> <p>The <b>Synchronize with Parameters</b> button is enabled only if a valid behavior is identified in the <b>Behavior</b> field.</p>	

##### Learn more

- [Select <Item> Dialog](#) <sup>[994]</sup>

#### 5.6.7.3.2 Synchronize Arguments

##### Topics

Topic	Detail	See also
<b>Usage</b>	<p>On the Call tab of the element Properties dialog, click on the <b>Synchronize with Parameters</b> button to synchronize the number of arguments in the invocation element with the number of parameters in the associated behavior.</p> <p>This automatically creates or deletes arguments based on the number of parameters in the behavior.</p> <p>If any arguments are to be deleted, Enterprise Architect prompts you to confirm the operation. Click on the <b>Yes</b> button to confirm.</p>	

##### Notes

- The **Synchronize with Parameters** button is enabled only if the invocation is associated with a valid behavior, as identified in the **Behavior** field

#### 5.6.7.4 Behavior Call Arguments

You define the arguments of a Behavior Call using the Arguments dialog, which you display by clicking on the **Edit** button on the Call tab of the element Properties dialog.

##### Define the arguments of a Behavior Call

Step	Action	See also
1	In the <b>Name</b> field, type the name of an argument to map to the behavior.	
2	In the <b>Parameters</b> field, click on the drop-down arrow and select a behaviors parameter from the list of parameters associated with the behavior.	
3	In the <b>Value</b> field, set any required value.	
4	If a diagram is displayed, and if required, select the <b>Show in current diagram</b> checkbox to add an Action Pin on the diagram.	<a href="#">Action Pin</a> [1277]
5	Click on the <b>Save</b> button.	
6	If appropriate, click on the <b>New</b> button and repeat steps 1 to 5 for another argument: parameter mapping.	

#### Notes

- If you attempt to map a newly created argument to a parameter that is already associated with a different argument, Enterprise Architect identifies the mapping and prompts you to confirm that you intend to change the association

#### Learn more

- [Behavior Calls](#) [1026]

### 5.6.7.5 Behavior Parameters

This topic area describes the facilities for defining, editing and extending the parameters of behavioral operations, Activities and Interactions.

#### Learn more

- [Define Parameters](#) [1028]
- [Parameter Tagged Values](#) [1030]
- [Operation Parameters By Reference](#) [1031]

#### 5.6.7.5.1 Define Parameters

Within the properties of an operation, Activity or Interaction, you can define parameters to control how that object and its subordinate objects behave.

The order or sequence of the parameters in the object properties is reproduced in code. You can set and manage the order using facilities available within the Parameter page of the object Properties dialog.

#### Reference

Field	Usage	See also
<b>Name</b>	Type the parameter name.	
<b>Type</b>	Select the data type of the parameter.  Alternatively, click on the ( ... ) button and select the element classifier to define the type.	
<b>Default</b>	Indicates an optional default value for the parameter.	
<b>Stereotype</b>	Indicates a stereotype name, specification of a Stereotype name can also be chosen via the drop-down list.	
<b>Kind</b>	Indicate the way a parameter is passed to a function: <ul style="list-style-type: none"><li>• <b>In</b> = By Value</li><li>• <b>InOut</b> = By Reference</li><li>• <b>Out</b> is passed by Reference, but only the return value is significant</li></ul>	
<b>Fixed</b>	Set the parameter to <i>const</i> , even if passed by reference.	
<b>Alias</b>	Indicate an optional alias for the parameter.	
<b>Add new to end</b>	Place new parameters at the end of the list instead of the start.	
<b>Multiplicity</b>	Display the Multiplicity dialog, to specify the multiplicity of the parameters.	
<b>Notes</b>	Indicate any additional notes on the parameter.	

Below is information regarding the Multiplicity dialog.

#### Reference

Field	Usage	See also
<b>Lower bound</b>	Define a lower limit to the number of elements allowed in the collection.	
<b>Upper bound</b>	Define an upper limit to the number of elements allowed in the collection.	

Field	Usage	See also
<b>Allow Duplicates</b>	Indicate that duplicates are allowed.  This maps to the UML property <i>isUnique</i> , value <b>FALSE</b> .	
<b>Multiplicity is Ordered</b>	Indicate that the collection is ordered.	

### Notes

- To organize parameters into the required sequence:
  - Use the **Up** and **Down** buttons above the parameter list
  - Select the **Add new to end** checkbox to force new parameters to appear at the end of the list instead of the top
  - Apply the List Header facilities from the column headings at the top of the parameter list
- Set the amount of parameter detail to display in a specific diagram using the **Show Parameter Detail** drop-down list on the Diagram Properties dialog; the setting applies only to the current diagram, and the default is to show the type only

### Learn more

- [Parameter Tagged Values](#)<sup>[1030]</sup>
- [Operation Parameters by Reference](#)<sup>[1031]</sup>
- [Visible Class Members](#)<sup>[831]</sup>
- [List Header](#)<sup>[677]</sup>

#### 5.6.7.5.1.1 Parameter Tagged Values

Behavioral parameters can have Tagged Values associated with them. Tagged Values offer a convenient extension mechanism for UML elements; you can define any tags you like and then assign values to them using this form.

Tagged Values are written to the XML output, and can be input to other third party tools for code generation or other activities.

### How to

To add a Tagged Value for a parameter

Step	Action	See also
<b>1</b>	Double-click on the operation, Activity or Interaction containing the parameter in a diagram or in the Project Browser.  The Properties dialog displays.	

Step	Action	See also
2	Click on the Tagged Values tab, which shows the Tagged Values for the selected object and its parameters.	
3	Click on the required parameter in the Parameters compartment of the Tagged Values tab, and click on the <b>New Tags</b> button.  The Tagged Value dialog displays.	
4	In the <b>Tag</b> field, type the tag name (or select a defined tag from the drop-down list).  In the <b>Value</b> field type the initial tag value.	
5	Click on the <b>OK</b> button to confirm the Tagged Value.	

### Notes

- Tagged Values are supported for attributes, operations, objects and connectors
- Custom tags can be created using a predefined Tagged Value Type

### Learn more

- [Create a Custom Tagged Value](#)<sup>[1628]</sup>

#### 5.6.7.5.2 Operation Parameters by Reference

**Access**   **Tools | Options | Objects:** Feature reference indicator

### Topics

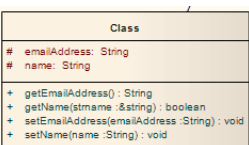
Image	Topic	Detail	See also
	<b>Usage</b>	<p>You can select to highlight parameters declared as 'Kind: <i>inout</i>' or 'Kind: <i>out</i>' with an additional user-defined prefix or suffix.</p> <p>If you select the <b>Highlight References</b> checkbox, you can also indicate whether a prefix or suffix should be used, and the actual reference character to use. In the example image, the <b>&amp;</b> character has been set as a prefix.</p> <p>When you declare a parameter of type <i>inout</i>, it is assumed you are passing the parameter by reference rather than by value. If you have elected to highlight references, then this is displayed in the Diagram View.</p> <p>The example to the left shows that, in the <i>getName</i></p>	<a href="#">Object Display Options</a> <sup>[631]</sup>

Image	Topic	Detail	See also
		operation, the parameter <i>strName</i> is a string reference, and is highlighted using the chosen character and position.	

### Notes

- This facility currently applies to operations only

## 5.6.8 In-place Editing Options


This topic explores the tasks that can be performed using in-place editing of elements on a diagram in Enterprise Architect. The tasks include:

- [View Properties](#) <sup>[1033]</sup>
- [Edit Element Item Name](#) <sup>[1034]</sup>
- [Edit Feature Stereotype](#) <sup>[1035]</sup>
- [Edit Feature Scope](#) <sup>[1036]</sup>
- [Edit Attribute Keyword](#) <sup>[1037]</sup>
- [Edit Operation Parameter Keyword](#) <sup>[1038]</sup>
- [Insert Operation Parameter](#) <sup>[1041]</sup>
- [Edit Parameter Kind](#) <sup>[1039]</sup>
- [Edit Custom Compartment](#) <sup>[1039]</sup>
- [Insert New Feature](#) <sup>[1040]</sup> (Attribute or Operation)
- [Insert Maintenance Feature](#) <sup>[1042]</sup>
- [Insert Testing Features](#) <sup>[1043]</sup>
- [Delete Selected from Model](#) <sup>[1033]</sup>

### 5.6.8.1 In-place Editing Tasks

#### How to

To use the in-place editing menu options

Images	Step	Action	See also
	1	Open the diagram containing the element.	
	2	Click on the element, and on the item to manipulate within the element.  The item line is highlighted in a lighter shade (the default is white), to indicate that it has been selected.	
	3	Edit and manipulate the items in the element, either by pressing the appropriate keyboard keys or by right-clicking on the highlighted item and choosing a task from the	



Images	Step	Action	See also
		<b>Element Items</b> context menu. The available commands are listed in the table below.	

### Reference

Menu option...	Usage	Shortcut	See also
<b>Edit Selected</b>	Change the name, scope or stereotype of the element or element item.	<b>F2</b>	
<b>View Properties</b>	Display the dialog containing details of the element.	<b>Enter</b>	
<b>Insert New After Selected</b>	Insert a new item in the element.	<b>Insert</b>	
<b>Find in Project Browser</b>	Locate the item in the Project Browser.		
<b>Add Attribute</b>	Add an attribute to the element.	<b>Ctrl + Shift + F9</b>	
<b>Add Operation</b>	Add an operation to the element.	<b>Ctrl + Shift + F10</b>	
<b>Add Other</b>	Insert a feature on the specific element item, such as Maintenance features and Testing features.	<b>Ctrl + F11</b>	
<b>Delete Selected from Model</b>	Delete the selected item from the model.	<b>Delete</b>	
<b>View Source code</b>	Display the source code for the element, in the default external editor.	<b>F12</b>	<a href="#">Editing Source Code</a> <sup>[2146]</sup>
<b>Visual Analyzer</b>	Set a breakpoint on a highlighted operation (including a breakpoint, start recording marker, end recording marker or stack auto-capture marker.)		<a href="#">Breakpoint and Marker Management</a> <sup>[2224]</sup> <a href="#">Marker Types</a> <sup>[2538]</sup>
<b>Set Type</b>	Set a classifier type for the selected item.	<b>Ctrl + L</b>	<a href="#">Select &lt;Item&gt; Dialog</a> <sup>[994]</sup>

Menu option...	Usage	Shortcut	See also
	Navigate Diagram Selection, to navigate the diagram between elements without having to use the mouse.	<b>Ctrl + Shift + arrow key</b>	
	Toggle element highlight option on and off.	<b>Shift + Enter</b>	

Other options that are available while editing element attributes or operations in a diagram include:

Menu options	Usage	Shortcut	See also
	Accept current changes.	<b>Enter</b>	
	Accept current changes and open a new slot to add a new item.	<b>Ctrl + Enter</b>	
	Abort the edit, without save.	<b>Esc</b>	
	Move the cursor between segments of a feature definition.	<b>Tab</b> or <b>Shift+Tab</b>	
	Display the context menu for in-place editing.	<b>Shift + F10</b>	
	Invoke the Select <Item> dialog.	<b>Ctrl+Shift+Space</b>	<a href="#">Select &lt;Item&gt; Dialog</a> <sup>[994]</sup>
	Display an autocompletion list for a feature classifier.	<b>Ctrl+Space</b>	



#### Notes

- Most of the in-place editing menu commands have keyboard alternatives; for many of them, if the selected item happens to be off-screen when you press the appropriate keys, the diagram automatically scrolls to show the whole element, so that you can see what you are changing

### 5.6.8.2 Edit Element Item Name

#### How to


To change the name of the element, or the name of an operation or attribute, directly from the diagram, follow the steps below


Images	Step	Action	See also
	1	Open the diagram containing the element.	
	2	Click on the element and on the name to change within the element.  The item line is highlighted in a lighter shade (the default is white), to indicate that it has been selected.	
	3	Right-click on the item.  The context menu displays.	
	4	Select the <b>Edit Selected</b> menu option, or press ( <b>F2</b> ) to enable you to edit the item directly from the diagram.  The name of the attribute or operation is highlighted.	
	5	Delete or type over the name.	
	6	Press ( <b>Enter</b> ) to accept the change, or ( <b>Esc</b> ) to cancel the change.	

### 5.6.8.3 Edit Feature Stereotype

#### How to

To change the stereotype of an operation or attribute directly from the diagram

Images	Step	Action	See also
	1	Open the diagram containing the element.	
	2	Click on the element, and on the item to edit within the element.  The item line is highlighted in a lighter shade (the default is white), to indicate that it has been selected.	
	3	Right-click on the item.  The context menu displays.	
	4	Select the <b>Edit Selected</b> menu option (or press ( <b>F2</b> ) ) to enable you to edit the attribute or operation directly from the diagram.  The name of the item is highlighted.	

Images	Step	Action	See also
	5	Move the cursor to the position before the name or within the existing attribute or operation stereotype (denoted by «stereotype name»).	
	6	Delete or type over the previous name to change the stereotype name of the attribute or operation.	
	7	Press ( <b>Enter</b> ) to accept the change or ( <b>Esc</b> ) to cancel the change.	



#### Notes

- You can assign multiple stereotypes by including a comma-separated list inside the stereotype markers

### 5.6.8.4 Edit Feature Scope

#### How to

To rapidly change the scope of an attribute or operation directly from the diagram

Images	Step	Action	See also
	1	Open the diagram containing the element.	
	2	Click on the element and on the item to edit within the element. The item line is highlighted in a lighter shade (the default is white), to indicate that it has been selected.	
	3	Right-click on the item. The context menu displays.	
	4	Select the <b>Edit Selected</b> menu option (or press ( <b>F2</b> ) ) to enable you to edit the attribute or operation directly from the diagram. The name of the item is highlighted.	
	5	Move the cursor to the scope of the item and delete the previous entry.	
	6	Reassign the entry by typing in one of the following symbols: <ul style="list-style-type: none"> <li>+ indicates that the scope is Public</li> <li>- indicates that the scope is Private</li> </ul>	


Images	Step	Action	See also
		<ul style="list-style-type: none"> <li>• ~ indicates that the scope is Package</li> <li>• # indicates that the scope is Protected</li> </ul>	
	7	<p>Press ( <b>Enter</b> ) to save the change, or ( <b>Esc</b> ) to cancel the change.</p> <p>The diagram is updated to reflect the changes (also see the <i>catalogNumber</i> attribute in the screen illustrations to the left).</p>	

### 5.6.8.5 Edit Attribute Keyword

You can add features such as attribute keywords and classifiers directly to an element, using the **Element Keywords and Classifiers** menu. This enables you to rapidly assign details element item by element item, directly from a diagram.

#### How to

To add features directly to an element

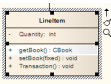
Images	Step	Action	See also
	1	In Enterprise Architect, open the diagram containing the element.	
	2	<p>Click on the element, and on the attribute to edit within the element.</p> <p>The item line is highlighted in a lighter shade (the default is white), to indicate that it has been selected.</p>	
	3	<p>Right-click on the item.</p> <p>The context menu displays.</p>	
	4	<p>Select the <b>Edit Selected</b> menu option (or press ( <b>F2</b> ) ) to enable you to edit the attribute directly from the diagram.</p> <p>The name of the attribute is highlighted.</p>	
	5	Right-click on the attribute name to display the context menu.	
	6	<p>From the context menu, you can:</p> <ul style="list-style-type: none"> <li>• Change the attribute classifier to static or fixed - select the <b>static</b> or <b>fixed</b> menu options as appropriate; the diagram is updated to reflect the changes</li> <li>• Display the Class properties - click on the <b>Goto Definition</b> menu option; Enterprise Architect locates the Class in the Project Browser and opens its Properties dialog</li> </ul>	<a href="#">Properties Dialog</a> <sup>[95]</sup>

Images	Step	Action	See also
		If the data type is a raw data type, Enterprise Architect displays the message: <i>The data type is a raw data type</i>	

### 5.6.8.6 Edit Operation Parameter Keyword

You can directly edit operation classifiers by element, using the in-place editing menu. This enables you to rapidly assign parameter keywords.

#### Directly edit operation classifiers by element


Images	Step	Action	See also
	1	Open the diagram containing the element.	
	2	Click on the element, and on the operation to edit within the element.  The item line is highlighted in a lighter shade (the default is white), to indicate that it has been selected.	
	3	Right-click on the item.  The context menu displays.	
	4	Select the <b>Edit Selected</b> menu option (or press ( <b>F2</b> ) ) to enable you to edit the operation directly from the diagram.  The name of the operation is highlighted.	
	5	Right-click on the data type of a parameter to display the context menu.	
	6	From the context menu you can: <ul style="list-style-type: none"> <li>Change the operation classifier by clicking on the appropriate menu option - <b>static</b>, <b>isquery</b>, <b>abstract</b> or <b>fixed</b>; the diagram is updated to reflect the changes</li> <li>Display the Class properties - click on the <b>Goto Definition</b> menu option</li> </ul> <p>If the data type is Class, Enterprise Architect locates the Class in the Project Browser and opens its Properties dialog</p> <p>If the data type is a raw data type, Enterprise Architect displays the message <i>This data type is a raw data type</i></p> <p>If the data type is not defined in the model, the message is:</p>	<a href="#">Properties Dialog</a> 9561

Images	Step	Action	See also
		<i>The data type is not defined in the model</i>	

### 5.6.8.7 Edit Parameter Kind

You can edit operation parameter kinds such as ( *in* ), ( *inout* ) and ( *out* ) directly from a diagram element by element, using the **Element Keywords and Classifiers** menu. This enables you to rapidly assign the parameter directly from a diagram.

#### Edit operation parameter kinds directly from a diagram

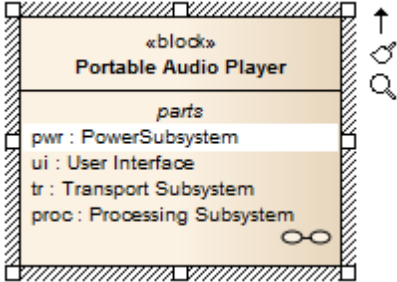
Images	Step	Action	See also
	1	In Enterprise Architect, open the diagram containing the element.	
	2	Click on the element, and on the operation to edit within the element.  The item line is highlighted in a lighter shade (the default is white), to indicate that it has been selected.	
	3	Right-click on the item.  The context menu displays.	
	4	Select the <b>Edit Selected</b> menu option (or press <b>F2</b> ) to enable you to edit the item directly from the diagram.  The name of the item is highlighted.	
	5	Right-click on the item name to display the context menu.	
	6	Select the appropriate menu option for the parameter kind value: ( <b>in</b> ), ( <b>inout</b> ) and ( <b>out</b> ).  The diagram is updated to reflect the change.	

### 5.6.8.8 Edit Custom Compartment

Items listed within a Custom Compartment can also be in-place edited. They have a similar in-place editing menu as other compartment items such as Operations and Attributes within a Class.

For example: a SysML Block element can have a *parts* custom compartment. In-place editing can be performed upon the Parts listed within this compartment.

#### In-place edit an item in a Custom Compartment

Step	Action	See Also
1	Open the diagram containing the element.	
2	<p>Click on the element and on the item to change within the custom compartment.</p> <p>The item line is highlighted in a lighter shade (the default is white), to indicate that it has been selected.</p> 	
3	<p>Right-click on the item.</p> <p>The context menu displays.</p>	
4	<p>Select the <b>Edit Selected</b> menu option, or press ( <b>F2</b> ) to enable you to edit the item directly from the diagram.</p> <p>The name of the item is highlighted.</p>	
5	Delete or type over the name.	
6	Press ( <b>Enter</b> ) to accept the change, or ( <b>Esc</b> ) to cancel the change.	

#### Notes

- Custom Compartments such as Parts and Ports within a SysML Block also provide a **Set Type** command on the right-click menu to set the classifier type of the selected item.

#### Learn more


- [In-place Editing Tasks](#)<sup>[1032]</sup>
- [Add Custom Compartments to Element](#)<sup>[1604]</sup>

### 5.6.8.9 Insert New Feature

#### How to

To add attributes and operations to a Class diagram element




Images	Step	Action	See also
	1	Open the diagram containing the element to which you are adding an attribute or operation.	
	2	Click on the element and, within the element, on the item after which to insert the operation or attribute.  The item line is highlighted in a lighter shade (the default is white), to indicate that it has been selected.	
	3	Either: <ul style="list-style-type: none"> <li>Press ( <b>Insert</b> ) or</li> <li>Right-click on the selected element item to display the context menu and select the <b>Insert New After Selected</b> menu option</li> </ul> Enterprise Architect inserts a new data line in the diagram, underneath the selected item.	
	4	Type in the relevant information for the attribute or operation.  Use the ( <b>Tab</b> ) and ( <b>Tab+Shift</b> ) keys to move the cursor through the segments of the attribute or operation, and select the classifier from either the: <ul style="list-style-type: none"> <li>Autocompletion list (press the ( <b>Ctrl+Space</b> ) keys) or</li> <li>Select &lt;Item&gt; dialog (press the ( <b>Ctrl+Shift+Space</b> ) keys)</li> </ul>	<a href="#">Autocompletion List</a> <sup>[2160]</sup> <a href="#">Select &lt;Item&gt; Dialog</a> <sup>[994]</sup>
	5	Press ( <b>Enter</b> ) to accept the change or ( <b>Esc</b> ) to cancel the change.  The diagram is updated to reflect the changes.	

#### 5.6.8.10 Insert Operation Parameter

You can add operation parameters to an operation through the in-place editing options, using hotkey commands or menu shortcuts.

##### Add parameters to operations in a Class diagram element

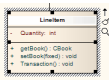
Images	Step	Action	See also
	1	Open the diagram containing the element.	
	2	Click on the element, and on the operation to update within the element.  The item line is highlighted in a lighter shade (the default is white), to indicate that it has been selected.	
	3	Press ( <b>F2</b> ), or right-click on the selected item to display the context menu and select the <b>Edit Selected</b> option.	

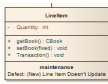
Images	Step	Action	See also
	4	Move the cursor inside the parameter brackets and type the parameter name, followed by a colon (for example, <b>bks:</b> for a vector containing books).	
	5	<p>Give the parameter a type.</p> <p>Place the cursor after the colon at the end of the name and add the classifier; either:</p> <ul style="list-style-type: none"> <li>Press ( <b>Ctrl+Space</b> ) to invoke the classifier autocompletion list</li> <li>Press ( <b>Ctrl+Shift+Space</b> ) to select a classifier from the Select &lt;Item&gt; dialog or</li> <li>Right-click the mouse to display the inline editing options context menu and select the <b>Insert Classifier</b> option</li> </ul> <p>The Select &lt;Item&gt; dialog displays.</p>	<a href="#">Autocompletion List</a> <sup>[2160]</sup> <a href="#">Select &lt;Item&gt; Dialog</a> <sup>[994]</sup>
	6	<p>Press ( <b>Enter</b> ) to accept the change or ( <b>Esc</b> ) to cancel the change.</p> <p>The diagram is updated to reflect the changes.</p>	

### 5.6.8.11 Insert Maintenance Feature

#### How to

To rapidly assign maintenance details such as Defects, Changes, Issues and Tasks directly to an element from a diagram

Images	Step	Action	See also
	1	Open the diagram containing the element.	
	2	<p>Click on the element name.</p> <p>The name is highlighted in a lighter shade (the default is white), to indicate that it has been selected.</p>	
	3	<p>Either:</p> <ul style="list-style-type: none"> <li>Press ( <b>Ctrl+F11</b> ) or</li> <li>Right-click on the highlighted name to display the context menu, and select the <b>Add Other</b> option</li> </ul> <p>The Insert Feature dialog displays.</p>	
	4	Click on the appropriate radio button option to associate the required	

Images	Step	Action	See also
		maintenance feature with the element item.	
	5	Click on the <b>OK</b> button. The <Maintenance Feature> details for <element> dialog displays.	
	6	Complete the fields to define the maintenance activity, and then click on the <b>Apply</b> button. To create a subsequent maintenance activity of this type, click on the <b>New</b> button.	
	7	When you have defined all of the maintenance activities of this type, click on the <b>OK</b> button. The maintenance details are added to the element.	

### Notes

- To ensure that the maintenance items are visible in the diagram element, as shown in step 7, select the **Maintenance** checkbox on the Elements tab of the Diagram Properties dialog


### Learn more


- [Show Maintenance Script in Diagram](#) 

## 5.6.8.12 Insert Testing Features

### How to

To rapidly add testing features such as Unit, Integration, System, Acceptance and Scenario tests to an element directly from a diagram

Images	Step	Action	See also
	1	Open the diagram containing the element.	
	2	Click on the element. The element name is highlighted in a lighter shade (the default is white), to indicate that it has been selected.	
	3	Either: <ul style="list-style-type: none"> <li>Press ( <b>Ctrl+F11</b> ) or</li> <li>Right-click on the highlighted name to display the context menu and select the <b>Add Other</b> option</li> </ul>	

Images	Step	Action	See also
		The Insert Feature dialog displays.	
	4	Click on the appropriate radio button option to associate the required testing feature with the element.	
	5	Click on the <b>OK</b> button.  The Testing window opens, showing the appropriate panel for the type of test selected.	
	6	Complete the fields to define the test activity, and then click on the <b>Save</b> icon in the window toolbar.  The test is added to the element.	<a href="#">Working on Test Records</a> <sup>[2605]</sup>
	7	To create a subsequent test activity of this type, click on the <b>New</b> icon.  To add items for other types of test, click on the appropriate tab.	

### Notes

- To ensure that the test items are visible in the diagram element, as shown in step 7, select the **Testing** checkbox on the Elements tab of the Diagram Properties dialog

### Learn more







- [Show Test Script Compartments](#) <sup>[2618]</sup>

## 5.6.9 Linked Documents

When you are developing elements in your model, you might want to provide extensive documentation on an element, to provide more structured text than you can create in the element **Notes** field. In this case, you can create a formatted document that is linked to the element. You can create a linked document from scratch, or from one of a range of Linked Document Templates that you define with the Linked Document Template Editor. The linked documents are held within the model.

### Working with Linked Documents

Task	Detail	See also
<b>Create linked documents</b>	<p>You can create a linked document either:</p> <ul style="list-style-type: none"> <li>Within a Document Artifact element attached to your model element, or</li> <li>Directly on the UML element</li> </ul> <p>The advantages of using a Document Artifact to host the</p>	<p><a href="#">Create Document Artifact</a> <sup>[1046]</sup></p> <p><a href="#">Create Linked Document on an Element</a> <sup>[1047]</sup></p>

Task	Detail	See also
	<p>linked document include:</p> <ul style="list-style-type: none"> <li>• <b>Visibility</b> of the Document Artifact and hence linked document in the Project Browser</li> <li>• As you can only have one linked document in an element, hosting each of several documents in its own Document Artifact and connecting the <b>several</b> Document Artifacts to <b>one</b> model element means that you have <b>broader scope</b> for providing information</li> <li>• Similarly, associating <b>several</b> elements with <b>one</b> document by connecting them to the host Document Artifact element means that you can <b>share information</b> rather than duplicating it</li> </ul> <p>The advantages of using a linked document directly on an element include:</p> <ul style="list-style-type: none"> <li>• More tightly binding the content of the document to the element, so there is less ambiguity in what the information refers to</li> <li>• More tightly binding the document itself to the element, so there is less risk of the document being changed, moved or deleted without reference to the element it belongs to</li> </ul>	
<b>Create linked document templates</b>	<p>You can base your linked document on one of a range of system-provided templates, each of which represents a project document type.</p> <p>You can also create and use your own linked document templates in the Resources window, within the <i>Document Generation &gt; Linked Document Templates</i> folder.</p>	<a href="#">Create Linked Document Templates</a> 
<b>Edit linked documents</b>	<p>You can define the style, format and layout of your linked document both in the template from which you create the document and in the document itself.</p>	<a href="#">Edit Linked Document Templates</a>  <a href="#">Edit Linked Documents</a> 
<b>Create hyperlinks in linked documents</b>	<p>Linked documents are intimately associated with specific elements, but you can also create hyperlinks in the document directly to other elements in the model. Having followed the link, you can perform any normal operation on that element.</p> <p>Similarly, you can create <b>new</b> elements or diagrams to link to.</p> <p>You can also create hyperlinks to other forms of documentation.</p>	<a href="#">Hyperlink From Linked Document</a>  <a href="#">Create Element From Document</a> 
<b>Include linked documents in RTF reports</b>	<p>Linked documents, however they are created, are rendered into document reports when you select the <b>Linked Document</b> checkbox in the Document Template Designer.</p>	<a href="#">Reporting Linked Documents</a> 

Task	Detail	See also
<b>Delete linked documents</b>	You can completely remove a linked document from an element, or you can remove the <i>contents</i> of the document and replace them with the contents of an external file.	<a href="#">Replace or Delete Documents</a> <sup>[1097]</sup>

### Notes

- You can create a Document Artifact element to contain a linked document in **all** editions of Enterprise Architect
- You can create and link a document to any UML element in the model, in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect
- When you have saved the linked document on an element, an **A** symbol displays in the bottom right corner of the element in its diagrams
- You can work on linked documents through the Automation Interface, using the methods **GetLinkedDocument ()**, **DeleteLinkedDocument ()**, **SaveLinkedDocument (string Filename)** and **LoadLinkedDocument (string Filename)**

### Learn more

- [Document Artifact](#) <sup>[1373]</sup>
- [Element Class](#) <sup>[2887]</sup>

## 5.6.9.1 Create Document Artifact

You can create a Document Artifact element to host a linked document for another model element. You can either create an appropriate connector between the Document Artifact and the documented element (to indicate how the document impacts the element), or drag the Document Artifact over the model element on the diagram or in the Project Browser so that it becomes a child of the model element.

### Create a linked document in a Document Artifact

Step	Action	See also
<b>1</b>	Open the diagram containing the element for which to create the linked document.	
<b>2</b>	Open the Common page of the Diagram Toolbox. Drag and drop the Document Artifact element icon from the Toolbox into your diagram.	<a href="#">Diagram Toolbox</a> <sup>[792]</sup> <a href="#">Common Page</a> <sup>[800]</sup> <a href="#">Document Artifact</a> <sup>[1373]</sup>
<b>3</b>	Double-click on the Document Artifact element. A default blank-page document opens in the Linked Document Editor, based on the <i>Normal.rtf</i> style template.	<a href="#">The Normal.rtf Style Template</a> <sup>[1058]</sup>

Step	Action	See also
4	Apply a template and/or continue to develop the linked document as described in <i>Create Linked Document On UML Element</i> , from step 2 onwards.	<a href="#">Create Linked Document on an Element</a> <sup>[1047]</sup>

### Notes

- There is a limit of one linked document per Document Artifact element, but you can provide more than one linked document for a **model** element by linking it to more than one Document Artifact
- This operation is available in all editions of Enterprise Architect; it is the sole method of creating a linked document in the Desktop and Professional editions
- When you have saved the document, an **A** symbol displays in the bottom right corner of the element

### Learn more

- [Linked Documents](#)<sup>[1044]</sup>

## 5.6.9.2 Create Linked Document on an Element

If you need to provide more extensive and structured documentation on an element, you can develop this as a formatted document directly linked to the element. You can develop the document from a blank page or using a system template or user-defined template, at several different points in the work area.

**Access** Click on an element in the **Project Browser**, **Diagram List**, **Package Browser**, **Specification Manager**, **Model Search** or diagram, and either:

**Right-click | (Create) Linked Document**


**Ctrl+Alt+D**



**Element | Linked Document**

**View | Notes:**  in toolbar, OR

**Right-click | Properties > General:**  in Notes field toolbar

### Create a linked document

Step	Action	See also
1	A default blank-page document opens in the Linked Document Editor, based on the <i>Normal.rtf</i> style template.  The New Linked Document from Template dialog might display, depending on whether you have previously selected the <b>Show this dialog on document creation</b> checkbox. In this case, go to step 3.	<a href="#">The Normal.rtf Style Template</a> <sup>[1058]</sup>
2	If you do <b>not</b> want to apply a system template or user-defined template, go to step 6.  If you <b>do</b> want to base the document on a selected template, click on the  (	

Step	Action	See also
	<p><b>New</b>) icon in the editor toolbar. A prompt displays for you to confirm that you are clearing the current document; click on the <b>OK</b> button.</p> <p>The New Linked Document from Template dialog displays.</p>	
3	<p>In the <b>Template</b> field, click on the drop-down arrow and select the template from which to create the document.</p> <p>This can be either a system-supplied template or a template that you have previously created yourself.</p>	<a href="#">Create Linked Document Templates</a>  <sup>[1098]</sup>
4	<p>If you want to display the New Linked Document from Template dialog by default whenever you start to create a linked document, select the <b>Show this dialog on document creation</b> checkbox.</p> <p>If you no longer want to display the dialog by default, deselect the checkbox.</p>	
5	<p>Click on the <b>OK</b> button.</p> <p>The new linked document opens in the Linked Document Editor. If you have based the document on a selected template, the document shows the content applied by the template.</p>	
6	<p>Enter and format the text of the document, using the facilities of the Linked Document Editor.</p>	<a href="#">Edit Linked Documents</a>  <sup>[1048]</sup>

### Notes

- You can create only one linked document on an element; if you need to create more, develop each document within its own **Document Artifact element** and link these to the model element
- This operation is available in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions
- When you have saved the document, an **A** symbol displays in the bottom right corner of the element

### Learn more

- [Create Document Artifact](#)  <sup>[1046]</sup>

### 5.6.9.3 Edit Linked Documents

To develop the contents of a linked document, you use the Linked Document Editor. This is based on the common Document Editor, providing a wide range of convenient features for formatting and editing the document, and for adding links from and references to the content of the document. The editor facilities are provided from a context menu that you display by right-clicking in the body of the document.

**Access**    **Right-click on element | Edit Linked Document (Ctrl+Alt+D)**



Option Descriptions

The Linked Document Editor option descriptions are grouped according to the actions you are performing.

Actions	Link
<b>Creating and importing documents</b>	<a href="#">File Control</a> <sup>[1050]</sup>
<b>Configuring the editor page display and formatting tools shown</b>	<a href="#">Editor Tool Display Options</a> <sup>[1052]</sup>
<b>Incorporating stylesheets, special texts and Tables of Contents</b> <b>Managing the base styles in the Normal.rtf style template file</b>	<a href="#">Styles, Special Texts &amp; Table of Contents</a> <sup>[1054]</sup> <a href="#">The Normal.rtf Style Template</a> <sup>[1058]</sup>
<b>Moving through, searching and selecting text</b>	<a href="#">Scroll, Search and Select Text</a> <sup>[1059]</sup>
<b>Formatting characters and text strings</b>	<a href="#">Format Text</a> <sup>[1062]</sup>
<b>Formatting paragraphs and text blocks</b>	<a href="#">Format Paragraphs</a> <sup>[1064]</sup>
<b>Inserting tab points</b>	<a href="#">Set Tabs</a> <sup>[1067]</sup>
<b>Inserting sections, columns and page breaks, and repaginating</b>	<a href="#">Define Document Sections</a> <sup>[1069]</sup>
<b>Inserting and editing page headers and footers, and footnotes and endnotes</b>	<a href="#">Insert Headers, Footers, Footnotes and Endnotes</a> <sup>[1071]</sup>
<b>Inserting tables</b>	<a href="#">Create Tables</a> <sup>[1075]</sup>
<b>Inserting hyperlinks and bookmarks</b>	<a href="#">Hyperlink From Linked Document</a> <sup>[1095]</sup> <a href="#">Create Element From Document</a> <sup>[1096]</sup> <a href="#">Insert Reference Links</a> <sup>[1083]</sup>
<b>Inserting images, OLE objects, frames and drawing objects</b>	<a href="#">Insert Images, Objects and Frames</a> <sup>[1085]</sup>

Actions	Link
Printer setup and printing documents	<a href="#">Print Report Documents</a> <sup>[1094]</sup>
Checking spelling and use of words	<a href="#">Checking Text</a> <sup>[1089]</sup>
Tracking, accepting and rejecting changes to text	<a href="#">Track Changes</a> <sup>[1091]</sup>
Protecting document text from accidental change	<a href="#">Protect Document Contents</a> <sup>[1093]</sup>
Create and refer to definitions of Glossary terms in the Project Glossary (using the <b>Create   Glossary Definition</b> menu option)	<a href="#">Project Glossary View</a> <sup>[534]</sup>

### Notes

- Throughout your document editing:
  - To undo one or more immediately previous edits, press (**Ctrl+Z**), or select the **Edit | Undo** menu option; you can still undo a change even after you have saved the change
  - To redo one or more immediately previous undone edits, press (**Ctrl+Y**), or select the **Edit | Redo** menu option

#### 5.6.9.3.1 File Control

The Document Editor provides a number of options for changing and saving a document report file contents as a complete file.

**Access** Select **File** from the menu bar at the top of the Document Editor window, **or** right-click and select **File** within either:

- The Content panel of the Document Template Designer or
- The text area of the document

### File control options

To	Action	See also
<b>Clear a file for redesign</b>	<p>Select the <b>File   New</b> menu option. This clears the existing contents of the document.</p> <p>If you have made any unsaved changes to the file, the editor prompts you to save them. If you want to protect those changes, click on the <b>Yes</b> button to display a browser screen and select a file path and file name to save the changes to.</p> <p>This is a method of clearing out an existing document for redesign. As you do not change the name, all references to it simply pick up the</p>	

To	Action	See also
	changed content.	
<b>Roll back changes to previous version</b>	Select the <b>File   Revert</b> menu option to roll back the file contents to the previously-saved version.	
<b>Save changes</b>	Select the <b>File   Save</b> menu option or press <b>Ctrl+S</b> , to save the file contents to the current file.	
<b>Save changes as new file</b>	<p><b>Applicable to the Document Template Designer and Linked Document template editor</b></p> <p>Select the <b>File   Save As</b> menu option or press <b>Ctrl+Shift+S</b>, to save a template contents under a new file name, which you are prompted to type in.</p>	
<b>Import an RTF file</b>	<p>Select the <b>File   Import</b> menu option. This imports the contents of an existing document into the current file.</p> <p>The <b>Import</b> option is useful when creating documents from existing documents external to Enterprise Architect, such as your own corporate standards templates and, in conjunction with the <b>New</b> option, of updating the document from the external file.</p>	<a href="#">Import a Document Template</a> <sup>[2707]</sup>
<b>Export a Document file</b>	<p>Select the <b>File   Export</b> menu option. This saves your document to an external document file.</p> <p>The <b>Export</b> option is useful for saving a specific template or document to be imported into another project.</p>	
<b>Update the standard styles used</b>	<p>Select the <b>File   Update Styles</b> menu option to either:</p> <ul style="list-style-type: none"> <li>Re-import the standard styles from Normal.rtf, into the current document, or</li> <li>Apply one of the system-supplied standard Stylesheets to the document</li> </ul> <p>A submenu displays from which you select the appropriate style source.</p> <p>This is useful for maintaining consistency of styles across all your documents, including such things as numbering formats.</p>	<a href="#">The Normal.rtf Style Template</a> <sup>[1058]</sup> <a href="#">Selecting a Stylesheet</a> <sup>[2653]</sup>
<b>Filter the content of a document report template</b>	Select the <b>File   Document Options</b> menu option. This displays the reporting Document Options dialog, which you use to set filters for object properties and for specific elements, connectors and element features.	<a href="#">Document Options</a> <sup>[2654]</sup>

To	Action	See also
<b>Close the file and editor</b>	Select the <b>File   Close</b> menu option to close the Document Editor; if you have any unsaved changes, a prompt displays to save them.	

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Reporting | RTF Templates | Filter Elements**
- (Alt+F1) | **Enterprise Architect | Reporting | RTF Templates | Filter Connectors**

#### **5.6.9.3.2 Editor Tool Display Options**





Within the Document Editor, to help you edit the document contents you can configure the page display and the formatting tools visible, using the **View** context menu options.

**Access** Select **View** from the menu bar at the top of the Document Editor window, **or** right-click and select **View** within either:

- The Content panel of the Document Template Designer or
- The text area of the document

#### Editing tools and options

Menu Option	Action	See also
<b>Page Layout Mode</b>	Toggles the text display between page layout and full window layout.	
<b>Horizontal Ruler</b>	Displays or hides the ruler at the top of the page.  The ruler shows tab stops and paragraph indentation marks; you can also use it to create or delete tab stops.	
<b>Vertical Ruler</b>	Displays or hides the ruler against the left edge of the page.  The ruler indicates the depth of the top and bottom margins of the page.	
<b>Tool Bar</b>	Displays or hides the two toolbars above the ruler.  The tool bars provide convenient icons for simple text formatting, copying and pasting, page layout, print management and file save functions.	
<b>Status Ribbon</b>	Displays or hides the status ribbon at the bottom of the editing panel.	

Menu Option	Action	See also
	The status ribbon displays the current page number, line number and column number; it also indicates the current insert/overtyping mode (at the left end of the ribbon).	
<b>Paragraph Marker</b>	<p>Displays or hides the various space markers in the text, such as:</p> <ul style="list-style-type: none"> <li>• The paragraph marker at the end of each paragraph</li> <li>• The Tab indicators between tabbed blocks of text, and</li> <li>• The blank character marker between words</li> </ul> <p>This option is useful when working with page-formatted text that is structured using both style definitions and key-strokes.</p>	
<b>Hidden Text</b>	<p>Shows or hides 'hidden' text.</p> <p>Text formatted with the <b>Font   Hidden</b> option is shown with a dotted underline; when the <b>Hidden Text</b> option is turned off, the hidden text is not visible and the normal text closes up to single character or line spacing.</p>	<a href="#">Format Text</a> 
<b>Field Names</b>	<p><b>Applicable to all document editing:</b></p> <p>If you have inserted any data fields and values, the <b>Field Names</b> option also toggles between displaying field names in the text and displaying the field values.</p> <p><b>Applicable to the Document Template Designer:</b></p> <p>As you develop your custom document template, you right-click on sections to insert field markers and select the <b>View   Insert Fields</b> option. The <b>Field Names</b> option makes the <b>Insert Fields</b> option available.</p>	<a href="#">Add Section Content</a>  <a href="#">Add Section Content</a> 
<b>Hyperlink Cursor</b>	<p><b>Applicable to the Document Template Designer and Linked Document Template Editor:</b></p> <p>Toggles the cursor icon when you 'mouse over' a hyperlink, between your default image (usually a line or arrow) and a 'pointing hand'.</p>	
<b>Page Header/Footer</b>	<p>Displays or hides the text of page headers and footers.</p> <p>To edit the header or footer text, select the <b>Edit   Edit Page Header/Footer</b> menu option.</p>	<a href="#">Insert Headers, Footers, Footnotes and Endnotes</a> 
<b>Zoom</b>	<p>Displays the Zoom Parameter dialog, on which you select the appropriate percentage of normal size by which to shrink or enlarge the display of the document content.</p> <p>The range of possible zoom percentages is from 10 to 200.</p>	

Menu Option	Action	See also

### 5.6.9.3.3 Styles, Special Text & Table of Contents

Using the Document Editor, you can define character-based styles and paragraph-based styles.

- A character style constitutes a set of character formatting attributes and is applied to a character string
- A paragraph style constitutes both a set of character formatting attributes and a set of paragraph formatting attributes, and is applied to one or more paragraphs

You can also include special, structured text in the document, such as **page number**, **date and time**, **table of contents** and **text input** fields.

Every custom template or document you create is automatically based on the **Normal.rtf** template file. This provides default styles, numbering and other base formats, which you can change using the Document Editor facilities and save either as a specific document, template or document report Stylesheet.


**Access** Select options from the menu bar at the top of the Document Editor window, **or** right-click within either:

- The Content panel of the Document Template Designer or
- The text area of the document


### Editing Styles

To	Action	See also
<b>Use double-byte characters</b>	<p>Select the <b>Edit   Inline Ime</b> menu option.</p> <p>Double-byte characters are, broadly, complex characters used in languages such as Japanese, Chinese, and Cyrillic.</p> <p>With this menu option, you can use both single-byte characters and double-byte characters in your text, without having to switch to an external Input Method Editor (IME) application.</p>	
<b>Create and edit styles</b>	<p>Select the <b>Edit   Edit Style</b> menu option. The Edit Stylesheet dialog displays.</p> <p>Select the appropriate radio button to define a paragraph style or a character style.</p> <p>If you want to update an existing style, select that style from the list box. If you want to create a new style, select the <b>Create a new style</b> checkbox, and type a name in the <b>Style Name</b> field. Short style names are best, but you can type in long names if you wish to.</p> <p>To begin defining the style properties, click on the <b>OK</b> button.</p> <p>You can use the ruler, toolbar or context menu selections to modify the style properties; for an existing stylesheet item, these also reflect the currently-selected properties.</p> <p>After you have defined the required style, you set it by either selecting the <b>Edit   Edit Style</b> menu option again or clicking</p>	

To	Action	See also
	<p>anywhere in the document:</p> <ul style="list-style-type: none"> <li>If you modified an existing stylesheet item, the document automatically reflects the updated style</li> <li>If you created a new stylesheet item, you can apply the style to highlighted text by selecting the <b>Font   Style</b> or <b>Paragraph   Style</b> menu options.</li> </ul>	
<b>Edit the hyperlink style</b>	<p>This option operates specifically on the style of any hyperlinks you want to create.</p> <p>Select the <b>Font   Hyperlink</b> menu option, and select the font, font size and emphasis (bold, italic) you want to apply. You cannot remove the hyperlink underline or change the color from blue.</p> <p>Select the <b>Font   Hyperlink</b> menu option again to set the style.</p> <p>When you create a hyperlink from this point onwards, the link text will be in the style you have set. Links created before you changed the style remain in the earlier style.</p>	<a href="#">Insert Reference Links</a> <sup>[1083]</sup>
<b>Apply character styles</b>	<p>Highlight the text to be styled and select the <b>Font   Style</b> menu option; the Select a Style dialog displays, from which you select the character style to apply.</p>	
<b>Apply paragraph styles</b>	<p>Highlight the text to be styled and select the <b>Paragraph   Style</b> menu option; the Select a Style dialog displays, from which you select the paragraph style to apply.</p>	
<b>Insert a table of contents</b>	<p>Edit the Table of Contents level styles (<b>TOC 1 to TOC 9</b>) using the <b>Edit   Edit Style</b> menu option, as you wish. Add at least one <i>Heading n</i> style to the document.</p> <p>Move the cursor to the point at which to insert the table of contents and select the <b>Insert   Table of Contents</b> menu option. The heading '<u>Table of Contents</u>' displays to mark the presence of a table of contents; this heading is not replicated in any document generated from a template.</p> <p>To create a Table of Contents heading that displays in the generated document, type the appropriate text at least one line above the inserted list and define the style as you wish - use a Heading style to include the heading itself in the Table of Contents, otherwise use a non-heading style.</p> <p>The Table of Contents is automatically generated and updated whenever generation or repagination occurs.</p>	<a href="#">Define Document Sections</a> <sup>[1069]</sup>
<b>Insert date and time fields</b>	<p>At the point in the text at which to insert the current date and/or time field, select the <b>Insert   Date and Time</b> menu option.</p> <p>The Insert Current Date/Time dialog displays, from which you can select the required date and time format. You can insert the date or time fields separately, or as a combined field.</p>	

To	Action	See also
	The date and time are automatically updated whenever the page text is refreshed (that is, clicked on).	
<b>Insert your own data fields</b>	<p>Move the cursor to the point at which to insert the data field and select the <b>Insert   Data Field</b> menu option.</p> <p>The Data Field Parameters dialog displays, in which you enter the field name and the data value. Wherever you insert the data field, the values are automatically updated with the most recent data value you enter.</p>	
<b>Insert your own text entry field</b>	<p>You insert text entry fields so that anyone using the document can type in their own text in response to either a lead-in sentence or the field name itself.</p> <p>Move the cursor to the point at which to insert the text entry field and type in any lead-in text you want to use. Then select the <b>Insert   Text Input Field</b> menu option.</p> <p>The Input Field Parameters dialog displays, in which you enter:</p> <ul style="list-style-type: none"> <li>• The field name</li> <li>• The initial value to display as a default</li> <li>• The maximum length of the field</li> <li>• The text font in which to display the text</li> </ul> <p>You can also specify whether or not the field has a border.</p> <p>For example: <i>Hello</i> <input type="text" value="Type your name here"/></p> <p>If you insert a text entry field in a template, any document generated from the template shows the typed entry as normal text, not boxed.</p>	
<b>Insert a Project Constant</b>	<p>(For Document Report templates.)</p> <p>You can create <b>Project Constants</b> to insert a specific text string wherever you place the flag for that string. You can therefore design a report containing many instances of that string, but create and edit the actual value in one place - on the Project Constants tab of the Generate Report dialog.</p> <p>Right click at the point at which you want to insert the Constant (flag), and select the <b>Project Constants   &lt;flag name&gt;</b> context menu option. The Constant displays at that point in the template, and when the report is generated the flag is replaced by its defined value.</p> <p>You can insert a Project Constant anywhere in the template - in the text body, section headings, headers, footers, endnotes or footnotes. Project Constants are available to any template in the model.</p>	<a href="#">Project Constants</a> 



To	Action	See also
<b>Insert a Report Constant</b>	<p>(For Document Report templates.)</p> <p>The Document Editor provides a set of data fields that represent report data such as the report name, title, generation date, version, author and status. When the report is generated, each field is replaced by the current value of the data item.</p> <p>Right-click on the point in the template to insert the data field and select the <b>Report Constants   &lt;field name&gt;</b> context menu option.</p> <p>You can insert a report constant anywhere in your template - in text, in headers or footers, or in cover page material. They are particularly suited for cover pages.</p>	
<b>Insert a selectable checkbox</b>	<p>You insert checkboxes in a document so that the users of the document can mark the item represented by the box. The checkbox can be selected and deselected in a document, a template and a document generated from a template.</p> <p>Move the cursor to the point at which to insert the checkbox and select the <b>Insert   Checkbox Field</b> menu option.</p> <p>The Checkbox Field Parameters dialog displays, in which you enter:</p> <ul style="list-style-type: none"> <li>• The field name (which does not display in the text)</li> <li>• Whether the checkbox defaults to selected</li> <li>• The size of the checkbox</li> </ul> <p>Click on the <b>OK</b> button.</p>	
<b>Insert a selection (drop-down) field</b>	<p>You insert selection fields in a document so that the users of the document can select one of the items presented in a drop-down list.</p> <p>Move the cursor to the point at which to insert the selection field and select the <b>Insert   Selection Field</b> option.</p> <p>The Selection Field Parameters dialog displays, in which you enter:</p> <ul style="list-style-type: none"> <li>• The field name</li> <li>• The options to present in the drop-down list (separated by the   character)</li> <li>• The value from this list to display as a default</li> </ul> <p>Click on the <b>OK</b> button.</p>	
<b>Define level numbering in generated document</b>	<p>(Used in the Document Template Editor and Linked Document Template Editor.)</p> <p>Select the <b>Edit   List and Overrides</b> menu option.</p> <p>Set up the numbering list and the list overrides.</p> <p>Apply the numbering list to the headings set for packages and</p>	<a href="#">Apply User-Defined Section Numbering</a> 

To	Action	See also
	elements, using paragraph numbering.	





#### 5.6.9.3.4 The Normal.rtf Style Template

Every custom template or document you create is automatically based on the **Normal.rtf** template file. This is an external file that provides default styles, numbering and other base formats, such as the default list numbering, **MasterList**. You can edit the Normal.rtf file to apply your own default styles to the documents and templates you create. You also have the options of:

- Editing selected styles within a template, so that reports or documents created from that template always have those different styles
- Using the standard Stylesheets or creating custom Stylesheets, to optionally apply to any document report you generate independent of the template you are using

**Access** The Normal.rtf template is stored in %APPDATA%\Sparx Systems\EA\DocTemplates\Document Templates

#### Edit Normal.rtf

Step	Action	See also
1	Open the Document Template Designer and create a new template called, for example, <i>Normal</i> .	<a href="#">Design Custom Document Templates</a> 
2	Select the <b>File   Import</b> menu option to import the <i>Normal.rtf</i> file into the <i>Normal</i> template.	<a href="#">Import Template</a> 
3	Modify the <i>Normal</i> template style definitions as you want, and remove any text before you save the template.	<a href="#">Styles, Special Text &amp; Table of Contents</a> 
4	<b>Export</b> the modified template ( <b>File   Export</b> ) back into the Normal.rtf file in the <i>RTF Templates</i> folder. The modified styles will be applied to all documents and templates you create from now on.	<a href="#">File Control</a> 

#### Notes

- You must not edit the **SSbookmark** style; this defines the section styles and must be preserved
- A related feature in the Document Editor is the **File | Update Styles** menu option, which you select to update an **existing** template to reflect any changes to Normal.rtf, or to apply a standard Stylesheet to the template

**Learn more**

- [Update Styles](#)<sup>[1051]</sup>
- [Apply User-Defined Section Numbering](#)<sup>[1081]</sup>
- [Selecting a Stylesheet](#)<sup>[2653]</sup>
- [Notes on Creating Stylesheets](#)<sup>[2705]</sup>

**Learning Center topics**

- (Alt+F1) | **Enterprise Architect | Reporting | RTF Templates | Altering the Default Template**

**5.6.9.3.5 Scroll, Search and Select Text**

In the Document Editor, you can move the cursor through the template or document using either the keyboard or the mouse. Alternatively, you could move the cursor to a position in the file by searching for a text string.

Having positioned the cursor on the required text, you can then highlight and manipulate it with various options, in addition to the standard 'mouse-click-and-drag' and the **Ctrl+C**, **Ctrl+X** and **Ctrl+P** keys.

**Access** In text, or select options from the menu bar at the top of the Document Editor window, **or** right-click within either:

- The Content panel of the Document Template Designer or
- The text area of the document

**Cursor Positioning**

Scroll Using	Options
<b>Keyboard Keys</b>	Press: <ul style="list-style-type: none"> <li>• <b>Ctrl+←</b> to move to the previous word</li> <li>• <b>Ctrl+→</b> to move to the next word</li> <li>• <b>Ctrl+↑</b> to move to the first column of the current line (if not already on the first column) or the first column of the previous line</li> <li>• <b>Ctrl+↓</b> to move to the first column of the next line</li> <li>• <b>F10</b> (or select the <b>Other   Jump</b> menu option), type in a line number to jump to, and click on the <b>OK</b> button</li> </ul>
<b>Mouse</b>	To perform various scrolling functions, click on the vertical or horizontal scroll bars. These functions are available when the scroll bars have been enabled by the startup parameters.  <b>Vertical Scroll Bar:</b> To scroll the screen up or down by one line, click on the arrows on either end.  To scroll the screen up by one page, click above the elevator (the block within the scroll bar); similarly, to scroll the screen down by one page, click below the elevator.  You can also scroll the screen up or down by dragging the elevator in the appropriate direction.

Scroll Using	Options
	<p><b>Horizontal Scroll Bar:</b> To scroll the screen left or right by one column, click on the arrow on the appropriate end of the bar; to scroll the screen left or right by half a screen, click on the bar on the appropriate side of the elevator.</p> <p>You can also scroll the screen left or right by dragging the elevator along the bar in the appropriate direction.</p>

### Text Search

When you **search for** a text string, the dialog for specifying the search details does not persist on the screen, so you can simply use keyboard keys or a menu option to continue the search without obscuring text behind a dialog.

To	Action	See also
<b>Find first instance of a text string</b>	<p>Select the <b>Other   Search</b> menu option, or press <b>F5</b>.</p> <p>The Search String Parameters dialog displays. Specify:</p> <ul style="list-style-type: none"> <li>• The term to search for</li> <li>• Whether to search from the start of the file or forwards or backwards from the current cursor position, and</li> <li>• Whether the search should exactly match the case of the search term</li> </ul> <p>The editor searches for the <b>first</b> instance of the specified character string as defined by the parameters.</p>	
<b>Find next instance of a text string</b>	<p>Select the <b>Other   Search Forward</b> menu option, or press <b>Ctrl+F</b>.</p> <p>The editor searches forwards for the next instance of the specified text string in the file, and highlights it.</p>	
<b>Find previous instance of a text string</b>	<p>Select the <b>Other   Search Backward</b> menu option, or press <b>Ctrl+Shift+F</b>.</p> <p>The editor searches backwards for the previous instance of the text string in the file, and highlights it.</p>	
<b>Replace a text string</b>	<p>Select the <b>Other   Replace</b> menu option, or press <b>F6</b>.</p> <p>The Replace String Parameters dialog displays, in which you specify:</p> <ul style="list-style-type: none"> <li>• The text string to locate</li> <li>• The text string to replace it with</li> <li>• Whether to search the whole document or just a highlighted block of text</li> <li>• Whether to confirm each replacement before making the change</li> </ul> <p>If you want to confirm replacement, a prompt displays for each</p>	

To	Action	See also
	occurrence of the search term to confirm or skip replacing that occurrence.	

### Text Selection

To	Action	See also
<b>Highlight a word</b>	Double-click on the word.	
<b>Highlight a line</b>	Move the cursor onto the line and press <b>F8</b> .	
<b>Delete a line</b>	Press <b>Shift+F9</b> to delete the current line; the remaining lines close up.	
<b>Paste special objects</b>	<p>Select the <b>Edit   Paste Special</b> menu option; the Paste Special dialog displays, listing the appropriate data type formats for inserting the copied object, as listed below:</p> <p>Click on the <b>Paste</b> button to embed the data into your application, or click on the <b>Paste Link</b> button to create a link to the original file.</p> <p><b><u>Native Object Format</u></b></p> <p>If available, this is the first format in the list box; you can edit data in this format using the original application, by double-clicking the object.</p> <p><b><u>Formatted Text</u></b></p> <p>A text format; this option offers the most suitable format if the data is pasted from another text output application, as the font and formatting attributes are reproduced accurately.</p> <p><b><u>Unformatted Text</u></b></p> <p>Another text format; this option pastes the text without retaining the formatting information.</p> <p><b><u>Picture Format</u></b></p> <p>The data is available in Picture format; you can later edit the object, by double-clicking on it and invoking the Microsoft MS Draw application.</p> <p>This format is preferable to the Bitmap and the Device Independent Bitmap formats.</p> <p><b><u>Device Independent Bitmap and Regular Bitmap formats</u></b></p> <p>The data is available in bitmap formats; you can later edit the object, by double-clicking on it and invoking the Microsoft MS Draw application.</p>	

To	Action	See also
	The editor converts these formats into the Picture format before calling the drawing application.	

#### 5.6.9.3.6 Format Text

When you create a document template or document, the default character or text styles are already defined, as they are derived directly from the Normal.rtf style template file. **Within** the document you can edit any of these character styles or apply individual formatting to text strings as you wish.

When you change the format of existing text, any new characters you type immediately after that text automatically assume the formatting characteristics you have applied.

**Access** Select options from the menu bar at the top of the Document Editor window, **or** right-click within either:

- The Content panel of the Document Template Designer or
- The text area of the document

#### Text formatting options

To	Action	See also
<b>Apply character format</b>	<p>Highlight the text to which to apply the format, and use one or more of these menu options or key combinations:</p> <ul style="list-style-type: none"> <li>• <b>Font   Normal</b>, or press <b>Alt+0</b></li> <li>• <b>Font   Bold</b>, or press <b>Ctrl+B</b></li> <li>• <b>Font   Underline</b>, or press <b>Ctrl+U</b></li> <li>• <b>Font   Double Underline</b>, or press <b>Ctrl+D</b></li> <li>• <b>Font   Italic</b>, or press <b>Ctrl+I</b></li> <li>• <b>Font   Superscript</b>, or press <b>Alt+4</b></li> <li>• <b>Font   Subscript</b>, or press <b>Alt+5</b></li> <li>• <b>Font   Strike</b>, or press <b>Alt+6</b> (puts a line through the text)</li> <li>• <b>Font   Double Strike</b></li> <li>• <b>Font   All Caps</b></li> <li>• <b>Font   Small Caps</b></li> </ul> <p>To <b>reset</b> any character format, highlight the text and select the <b>Font   Normal</b> menu option, or press <b>Alt+0</b>.</p>	
<b>Change font typeface and point size</b>	<p>Highlight the text to change and select the <b>Font   Fonts</b> menu option, or press <b>Alt+F10</b>.</p> <p>The Font Selection dialog displays, from which you select the</p>	

To	Action	See also
	<p>required typeface and point size.</p> <p>Click on the <b>OK</b> button.</p>	
<b>Change character style</b>	<p>Highlight the text to change and select the <b>Font   Style</b> menu option.</p> <p>The Select a Style dialog displays, listing the currently-defined character styles in the template stylesheet.</p> <p>Select the required style and click on the <b>OK</b> button.</p>	<a href="#">Styles. Special Text &amp; Table of Contents</a> <sup>[1054]</sup>
<b>Change the color of text, background (permanent highlight) or underline</b>	<p>Highlight the text to change and select one or more of the following options, as required:</p> <ul style="list-style-type: none"> <li>• <b>Font   Text Color</b></li> <li>• <b>Font   Background Color</b></li> <li>• <b>Font   Underline Color</b></li> </ul> <p>In each case, the Color dialog displays, through which you can select or define the required color.</p> <p>When you have selected a color, click on the <b>OK</b> button.</p>	
<b>Change character spacing</b>	<p>Normal character spacing (the space between characters) is 20 twips; if you want to change this (or return to it), highlight the text to adjust, and select the <b>Font   Spacing</b> menu option.</p> <p>The Character Spacing dialog displays.</p> <p>Select the radio button to expand or compress spacing, or to return to normal spacing; if you are changing <b>from</b> normal spacing, enter the number of twips to set the spacing to.</p> <p>Click on the <b>OK</b> button.</p>	
<b>Hide text</b>	<p>Hidden text is not displayed on the screen or printer, but remains in the document and is not deleted.</p> <p>Highlight the text to hide and select the <b>Font   Hidden</b> menu option, or press <b>Ctrl+H</b>.</p> <p>The highlighted text is not displayed and the rest of the text closes up.</p> <p>To view hidden text, select the <b>View   Hidden Text</b> menu option; you can then make the text normal again by highlighting it and deselecting the <b>Font   Hidden</b> menu option.</p>	<a href="#">Editor Tool Display Options</a> <sup>[1052]</sup>
<b>Create a box around text</b>	<p>Highlight the text to box and select the <b>Font   Boxed</b> menu option; this creates a broken-line border around the selected text.</p>	

To	Action	See also
<b>Insert a non-breaking space</b>	Move the cursor to the point at which to insert the non-breaking space and select the <b>Insert   Non-breaking Space</b> menu option.	
<b>Insert a non-breaking dash</b>	Move the cursor to the point at which to insert the non-breaking dash and select the <b>Insert   Non-breaking Dash</b> menu option.	
<b>Insert an optional hyphen</b>	Move the cursor to the point at which to insert the hyphen and select the <b>Insert   Optional Hyphen</b> menu option.	

#### 5.6.9.3.7 Format Paragraphs

When you create a document template or document the default paragraph styles are already defined, as they are derived directly from the Normal.rtf style template file. **Within** the document you can edit any of these paragraph styles or apply individual formatting to text blocks as you wish.

Each formatting option operates on the current paragraph, or on a **highlighted** block of text.

**Access** Select options from the menu bar at the top of the Document Editor window, **or** right-click within either:

- The Content panel of the Document Template Designer or
- The text area of the document

#### Paragraph formatting options

To	Action	See also
<b>Clear all paragraph formatting</b>	Select the <b>Paragraph   Normal</b> menu option.  All paragraph formatting applied whilst you have been <b>within</b> the file is removed.	
<b>Set text flow in document</b>	To set the text flow for: <ul style="list-style-type: none"> <li>• A highlighted block of text, select the <b>Paragraph   Text Flow</b> menu option; the Paragraph Text Flow dialog displays</li> <li>• The entire document, select the <b>Edit   Document Text Flow</b> menu option; the Document Text Flow dialog displays</li> </ul> <p>In either case, select the required text flow direction and click on the <b>OK</b> button. The options are:</p> <ul style="list-style-type: none"> <li>• <b>Left to right</b></li> <li>• <b>Right to left</b></li> <li>• <b>Default</b></li> </ul>	



To	Action	See also
	The <b>Default</b> action depends on the language setting of the system; for example, if the system language is Hebrew or Chinese, the default text flow is right to left.	
<b>Center text</b>	Select the <b>Paragraph   Center</b> menu option, or press <b>Alt+8</b> .	
<b>Right-justify text</b>	Select the <b>Paragraph   Right Justify</b> menu option, or press <b>Alt+9</b> .	
<b>Justify both sides of text</b>	Select the <b>Paragraph   Justify Both</b> menu option.	
<b>Indent paragraph left</b>	Select the <b>Paragraph   Indent Left</b> menu option, or press <b>Alt+L</b> . Select the option again to increase the indent. To reduce or cancel the indent, click on the <b>Indent Left icon</b> on the lower toolbar above the text area.	
<b>Indent paragraph right</b>	Select the <b>Paragraph   Indent Right</b> menu option or press <b>Alt+R</b> . Select the option again to increase the indent.	
<b>Create hanging indent</b>	Select the <b>Paragraph   Hanging Indent</b> menu option or press <b>Alt+T</b> . Select the option again to increase the indent of all lines below the first.	
<b>Keep paragraph lines together</b>	Select the <b>Paragraph   Keep Together</b> menu option. The editor attempts to keep all lines within the paragraph on the same page.	
<b>Keep paragraphs together</b>	Select the <b>Paragraph   Keep with Next</b> menu option. The editor attempts to keep the last line of the current paragraph and the first line of the next paragraph on the same page.	
<b>Prevent 'widow' and 'orphan' lines</b>	Select the <b>Paragraph   Widow/Orphan Control</b> menu option. The editor attempts to avoid having: <ul style="list-style-type: none"> <li>the first line of the paragraph on the previous page ('widow' line)</li> <li>the last line of the paragraph on the next page ('orphan' line)</li> </ul>	

To	Action	See also
<b>Start text on new page</b>	Move the cursor to the point at which to start the new page, and select the <b>Paragraph   Page Break Before</b> menu option.	
<b>Insert border and shading for text block</b>	<p>Select the <b>Paragraph   Border and Shading</b> menu option.</p> <p>The Paragraph Box Parameters dialog displays, on which you specify:</p> <ul style="list-style-type: none"> <li>• which sides of the box to display (including a line between text lines)</li> <li>• whether the lines are thick or doubled</li> <li>• the degree of gray shading behind the text</li> <li>• the color of the lines</li> </ul>	
<b>Define line spacing</b>	<p>Select the <b>Paragraph   Paragraph Spacing</b> menu option.</p> <p>The Paragraph Spacing Parameters dialog displays, on which you specify the line spacing and the point spacing before and after lines.</p> <p>Alternatively, to just set double line spacing, select the <b>Paragraph   Double Space</b> menu option.</p>	
<b>Set a background color for text space</b>	<p>Select the <b>Paragraph   Background Color</b> menu option.</p> <p>The Color dialog displays, on which you select the background color.</p> <p>The editor highlights the full width of the page in that color, for the selected lines.</p>	
<b>Create a bulleted list</b>	<p>Select the <b>Paragraph   Bullet</b> menu option.</p> <p>The editor formats the lines into a simple bullet list.</p> <p>If you want to indent the list (or a part of it) further, select the <b>Paragraph   Increase Level</b> menu option. If you have increased the indent but now want to decrease it, select the <b>Paragraph   Decrease Level</b> option.</p>	
<b>Create a numbered list</b>	<p>Highlight the required lines of text and select the <b>Paragraph   Numbering</b> menu option.</p> <p>The editor formats the lines into a simple numbered list.</p> <p>To create multiple-level numbered lists, see the entry below.</p>	
<b>Apply numbering to paragraphs</b>	<p>(Used in the Document Template Editor and Linked Document Template Editor.)</p> <ol style="list-style-type: none"> <li>1. Set up a numbering list and overrides (<b>Edit   List and Overrides</b>).</li> <li>2. Apply the numbering levels to the template sections (<b>Paragraph   List Numbering</b>).</li> </ol>	<a href="#">Apply User-Defined Section Numbering</a> <sup>[1081]</sup>

To	Action	See also
<b>Set the list characters</b>	<p>When you have set up your bulleted or numbered list elements, you can toggle between the list character options using the drop-down arrow icon just to the <b>left</b> of the bullet list icon on the lower toolbar of the document template editor.</p> <p>When you click on this icon, you are prompted to select between a <b>single-level</b> list format and a <b>multiple-level</b> list format. When you select one of these options, you then make a further selection between:</p> <ul style="list-style-type: none"> <li>Single-level: <b>Arabic numerals</b>, <b>Roman numerals</b>, <b>Upper-case letters</b> of the alphabet and <b>lower-case letters</b></li> <li>Multiple-level: <b>Arabic numerals</b>, <b>Roman numerals</b> and <b>bullet symbols</b></li> </ul>	
<b>Apply a paragraph style from the template stylesheet</b>	<p>Select the <b>Paragraph   Style</b> menu option.</p> <p>The Select a Style dialog displays, listing the currently-defined paragraph styles in the template stylesheet.</p> <p>Select the required style and click on the <b>OK</b> button.</p>	<a href="#">Styles, Special Text &amp; Table of Contents</a> <sup>[1054]</sup>

#### Learn more

- [Scroll, Search and Select Text](#) <sup>[1053]</sup>

#### 5.6.9.3.8 Set Tabs

Using the Document Editor, you can define *left*, *right*, *center* and *decimal* tab points to define the positions of text strings and columns across the page. A paragraph or highlighted block of text can have as many as twenty tab points, which apply to every line of the selected text.

If you do not set tab points, the **Tab** key moves the cursor to default tab points at 0.5-inch intervals.

**Access** Select **Paragraph | Set Tab** from the menu bar at the top of the Document Editor window, **or** right-click and select **Paragraph | Set Tab** within either:

- The Content panel of the Document Template Designer or
- The text area of the document

Or click on the ruler at the top of the page.

#### Create and Clear Tab Points

Topic	Detail	See also
<b>Using the ruler</b>	<p>If the ruler is not visible, select the <b>View   Horizontal Ruler</b> context menu option.</p> <p>To create a:</p> <ul style="list-style-type: none"> <li><b>Left</b> tab, click on the required tab position on the ruler; the left tab is indicated on the ruler by an <b>L</b> shape</li> </ul>	

Topic	Detail	See also
	<ul style="list-style-type: none"> <li>• <b>Right</b> tab, right-click on the required tab position on the ruler; the right tab is indicated on the ruler by a reversed <b>L</b> shape</li> <li>• <b>Center</b> tab, press (<b>Shift</b>) and click on the required tab position on the ruler; the center tab is indicated on the ruler by an inverted <b>T</b> shape</li> <li>• <b>Decimal</b> tab, press (<b>Shift</b>) and right-click on the required tab position on the ruler; the decimal tab stop is indicated on the ruler by an inverted <b>T</b> shape with a dot on the right hand side</li> </ul> <p>This tab is for numbers with a <b>decimal point</b>; numbers scroll left from the tab until you type the point, then numbers scroll right.</p> <p>To move a tab position, click on the tab symbol on the ruler and drag it to the new position.</p>	
<b>Using the Set Tab menu option</b>	<p>The <b>Paragraph   Set Tab</b> menu option displays the Set a Tab Position dialog, on which you select from a set of radio buttons to specify the tab type.</p> <p>The dialog provides two advantages over the ruler; you can:</p> <ul style="list-style-type: none"> <li>• Set the tab position with more precision and with a clear value (in inches) that you can duplicate</li> <li>• Select from a set of radio buttons to add or remove a dotted, broken or solid line preceding the tab point</li> </ul> <p>Click on the <b>OK</b> button to save the settings.</p>	
<b>Clear a single tab point</b>	<p>Highlight the block of text and select the <b>Paragraph   Clear Tab</b> menu option.</p> <p>The Clear a Tab Position dialog displays.</p> <p>Select the tab to clear and click on the <b>OK</b> button.</p>	
<b>Clear all tab points for a text block</b>	<p>Select the block of text and select the <b>Paragraph   Clear All Tabs</b> menu option.</p>	

#### Notes

- The **Other | Snap To Grid** menu option affects the movement of the tabs (and the paragraph indentation markers) on the ruler; when you select this option, the movements of the tab markers are locked on to an invisible grid at intervals of 1/16 inch (half a ruler division)

### 5.6.9.3.9 Define Document Sections

Using the Document Editor, you can partition your text into separate sections, present the text in a section in columnar format, and force page and column breaks to position text as you want. A multiple section document is useful to:


- Change the presentation and format of the text for different areas of information
- Change between portrait and landscape orientation for different parts of the document
- Vary the page margins from one page to another

As you change the section structure, you can insert page counts and insert and reset page numbers, and repaginate the document to update the page number, page count and Table of Contents.


**Access** Select options from the menu bar at the top of the Document Editor window, **or** right-click within either:

- The Content panel of the Document Template Designer **or**
- The text area of the document

#### Manage sections

To	Action	See also
<b>Create a new section</b>	<p>Select the <b>Insert   Insert Break   Section Break</b> menu option. This creates a new section on a new page.</p> <p>The section break is represented by a dotted line; if this is not visible, click on the  icon in the toolbar above the text panel.</p> <p>This menu option is not effective when <b>Edit   Edit Page Header/Footer</b> is active, because editing control passes to the header and footer bands away from the body of the document.</p>	
<b>Set the section properties</b>	<p>Section properties apply to all the document content following the section break, up to the next section break line.</p> <p>Select the <b>Edit   Edit Section</b> menu option; the Section Parameters dialog displays.</p> <p>Complete each panel to set up the section as you prefer.</p>	
	<p>The <b>Number of Columns</b> field defaults to <b>1</b>, to present the text as a normal page of text (effectively a single column).</p> <p>If you want to structure the text in two or more columns on a page:</p> <ul style="list-style-type: none"> <li>• In the <b>Number of Columns</b> field, type the number of columns you want to divide the text into across the page</li> <li>• In the <b>Space Between the Columns</b> field, type the</li> </ul>	

To	Action	See also
	<p>separation of the columns, in inches</p> <ul style="list-style-type: none"> <li>If you want to separate the columns with a vertical line, select the <b>Line Between Columns</b> checkbox</li> </ul> <p>When you close the dialog, the text is displayed in the selected number of columns of equal width, across the page. The text wraps at the edge of the column and, when it reaches the foot of the page, continues at the top of the next column.</p>	
	<p>If you want to insert a page break before the new section, select the <b>Start Section on New Page</b> checkbox.</p> <p>If you also want to adjust the page numbering so that the new section starts at a specific page, select the <b>Restart Page Number at</b> checkbox and type the page number.</p> <p>The section start page will often be 1, but this facility is also useful for inserting another document of a constant number of pages between, for example, an introduction and a technical section.</p>	
	Set the direction of the <b>text flow</b> within the section by selecting the appropriate radio button.	<a href="#">Format Paragraphs</a> 1064
	Set the <b>page orientation</b> within the section by selecting the appropriate radio button.	
	<p>Select a standard <b>paper size</b> for printing the text of the section.</p> <p>Alternatively, you can scroll down the list and select the <b>User Defined</b> option, and then define your own page width and height, in inches.</p> <p>You can also define special page margins by selecting the <b>File   Page Layout</b> menu option.</p>	<a href="#">Print Documents</a> 1094
	<p>The <b>Paper Source</b> selections both default to <b>Auto Select</b>, so that the pages of the section print from whatever paper trays have paper available.</p> <p>If you are printing on paper from a specific tray, or are manually feeding paper to the printer for this section, select the appropriate option. You can select different options for the first page of the section and subsequent pages of the section.</p>	
<b>Force a page break</b>	Select the <b>Insert   Insert Break   Page Break</b> menu option, or press <b>Ctrl+Enter</b> .	

To	Action	See also
	The forced page break is indicated by a labeled solid line; if this is not visible, click on the  icon in the Toolbar above the text panel.	
<b>Create a column break</b>	<p>Move the cursor to the appropriate point in the column text and select the <b>Insert   Insert Break   Column Break</b> menu option. The break is indicated by a labeled 'dot and dash' line.</p> <p>Normally in a multiple column section, the text flows from the end of one column to the top of the next column; a column break forces the text to the next column before the current column is completely filled.</p>	
<b>Delete a break</b>	<p>A section break, page break or column break is indicated by a labeled line. To delete the break, move the cursor onto this line and press <b>Delete</b>.</p> <p>If you delete a section break, the section properties of the preceding section now apply to the document content following the deleted line.</p>	
<b>Insert the page number</b>	<p>Position the cursor at the point at which to display the page number, and select the <b>Insert   Page Number</b> menu option.</p> <p>The page number is displayed in gray.</p>	
<b>Insert the page count</b>	<p>Position the cursor at the point at which to display the total number of pages in the document, and select the <b>Insert   Page Count</b> menu option.</p> <p>The page count is displayed in gray.</p>	
<b>Repaginate the document</b>	<p>Select the <b>Edit   Repaginate</b> menu option.</p> <p>The <b>Page Number</b> and <b>Page Count</b> fields and the table of contents are all updated.</p>	

#### 5.6.9.3.10 Insert Headers, Footers, Footnotes and Endnotes

When creating a document report or template, you can include page headers, footers, endnotes and footnotes. You switch options on to create and edit these items, and then switch these options off again to protect the entries. While you are creating headers and footers, you can also continue to edit the body of the document. You can create headers and footers:


- To apply throughout the document
- Separately for each section that you might create in your document
- Separately for the first page of the document and for the subsequent pages (for example, to leave no header and footer text on the title page of the document)

- Separately for left-hand and right-hand pages

**Access** Select options from the menu bar at the top of the Document Editor window, **or** right-click within either:

- The Content panel of the Document Template Designer or
- The text area of the document

#### Header and footer options

To	Action	See also
<b>Create the page header and footer</b>	<p>Select the <b>Edit   Edit Page Header/Footer</b> menu option.</p> <p>A dotted line marks off the header area at the top of the page and the footer area at the bottom of the page. This dotted line is at a position set using the Page Parameters dialog (<b>File   Page Layout</b>).</p> <p>Within each of these two areas, a paragraph marker displays to indicate where the text will sit relative to the top and bottom edges of the paper. This position is also defined on the Page Parameters dialog.</p> <p>You can type in any kind of free-form or formatted text and insert icons as you wish. You would normally work well within the defined areas for the header and footer, but if the text exceeds the area, the dotted lines simply move to accommodate the text.</p>	<a href="#">Print Documents</a> 
<b>Edit or delete the page header and footer text</b>	<p>Select the <b>Edit   Edit Page Header/Footer</b> menu option, move the cursor to the text and make any changes required. If you need to, delete the text altogether.</p>	
<b>Create separate header and footer for the initial page of the document</b>	<p>Select the <b>Edit   Edit Page Header/Footer</b> menu option to access the header and footer text spaces.</p> <p>Select the <b>Edit   First Page Header/Footer</b> option and either:</p> <ul style="list-style-type: none"> <li>• The <b>Create First Page Header</b> option, or</li> <li>• The <b>Create First Page Footer</b> option</li> </ul> <p>A paragraph marker displays at the top or bottom of the first page of the document. If you want specific text to display on the first page of the document, type it in and format it.</p> <p>Alternatively, leave the spaces blank so that the first page has no header or footer text.</p>	
<b>Display the common document header and footer on the initial page</b>	<p>If you have created a separate initial page header and/or footer and now want to display the same header and footer text on the initial page as is shown on subsequent pages of the document, you specifically delete the 'first page header/footer' assignment.</p> <p>Select the <b>Edit   Edit Page Header/Footer</b> menu option to access the header and footer text spaces.</p>	



To	Action	See also
	<p>Select the <b>Edit   First Page Header/Footer</b> menu option and either:</p> <ul style="list-style-type: none"> <li>• The <b>Delete First Page Header</b> option or</li> <li>• The <b>Delete First Page Footer</b> option</li> </ul> <p>The 'first page' assignment and text are removed from the first page header and/or footer spaces.</p>	
<b>Create separate header and footer for left-hand pages</b>	<p>Select the <b>Edit   Edit Page Header/Footer</b> menu option to access the header and footer text spaces.</p> <p>Select the <b>Edit   Left Page Header/Footer</b> option and either:</p> <ul style="list-style-type: none"> <li>• The <b>Create Left Page Header</b> option, or</li> <li>• The <b>Create Left Page Footer</b> option</li> </ul> <p>Scroll to the nearest left-hand page and in the header or footer space type in any kind of free-form or formatted text and insert icons as you wish.</p> <p>This text displays on every left-hand page in the section or, if there are no sections, in the document.</p> <p>Format your document to put left-hand page information on pages with left-hand headers and footers. For example, in English language documents, even-numbered pages are usually left-hand pages.</p>	
<b>Create separate header and footer for right-hand pages</b>	<p>Select the <b>Edit   Edit Page Header/Footer</b> menu option to access the header and footer text spaces.</p> <p>Select the <b>Edit   Right Page Header/Footer</b> option and either:</p> <ul style="list-style-type: none"> <li>• The <b>Create Right Page Header</b> option, or</li> <li>• The <b>Create Right Page Footer</b> option</li> </ul> <p>Scroll to the nearest right-hand page and in the header or footer space type in any kind of free-form or formatted text and insert icons as you wish.</p> <p>This text displays on every right-hand page in the section or, if there are no sections, in the document.</p> <p>Format your document to put right-hand page information on pages with right-hand headers and footers. For example, in English language documents, odd-numbered pages are usually right-hand pages, and sections such as Title, Introduction and Table of Contents start on right-hand pages.</p>	
<b>Delete left or right page headers and footers</b>	<p>Select the <b>Edit   Edit Page Header/Footer</b> menu option to access the header and footer text spaces.</p> <p>Select the <b>Edit   &lt;side&gt; Page Header/Footer</b> option (where <b>&lt;side&gt;</b> is</p>	

To	Action	See also
	<p>either <b>Left</b> or <b>Right</b>) and either:</p> <ul style="list-style-type: none"> <li>• The <b>Delete &lt;side&gt; Page Header</b> option, or</li> <li>• The <b>Delete &lt;side&gt; Page Footer</b> option</li> </ul> <p>You are prompted to confirm the deletion; click on the <b>Yes</b> button.</p> <p>Left and Right page headers or footers balance each other. If you create one, the system responds as if the other exists but is blank. If you delete one, the system responds as if the other is the 'common' header/footer for the section, and all page headers now contain that remaining text.</p>	
<b>Close and protect headers and footers</b>	Select the <b>Edit   Edit Page Header/Footer</b> menu option again, and check that you cannot move the cursor into the header and footer spaces.	
<b>Create a footnote</b>	<p>Move the cursor to the position at which to insert the footnote marker, and select the <b>Insert   Footnote/Endnote   Footnote</b> menu option.</p> <p>The Footnote Parameters dialog displays. In the:</p> <ul style="list-style-type: none"> <li>• <b>Footnote Marker</b> field type the marker you want to use, such as a number, asterisk, hash or letter of the alphabet, or any text string of up to 10 characters</li> <li>• <b>Footnote Text</b> field, type the text of the footnote</li> <li>• <b>Superscript Footnote Marker</b> field, leave the checkbox selected to display the marker as a superscript, or clear the checkbox to display the marker in the same format as the preceding text</li> </ul> <p>Click on the <b>OK</b> button; the editor inserts the footnote marker at the current cursor location and displays the marker and text at the bottom of the page, above the page footer space.</p>	
<b>Edit footnote text</b>	<p>Select the <b>Edit   Edit Footnote/Endnote   Edit Footnote Text</b> menu option.</p> <p>The marker and text of each footnote displays in the <b>document text</b> where the marker was originally inserted, highlighted by a broken underline.</p> <p>Locate the text and make the required changes. These changes are immediately reflected in the footnote text at the bottom of the page.</p> <p>When you have finished editing footnote text, click again on the <b>Edit   Edit Footnote/Endnote   Edit Footnote Text</b> menu option; the footnote text is no longer shown in the document text.</p>	
<b>Create an endnote</b>	Move the cursor to the position at which to insert the endnote marker, and select the <b>Insert   Footnote/Endnote   Endnote</b> menu option.	

To	Action	See also
	<p>The Endnote Parameters dialog displays. In the:</p> <ul style="list-style-type: none"> <li>• <b>Endnote Marker</b> field type the marker you want to use, such as a number or letter of the alphabet, or any text string of up to 10 characters</li> <li>• <b>Endnote Text</b> field, type the text of the endnote</li> <li>• <b>Superscript Endnote Marker</b> field, leave the checkbox selected to display the marker as a superscript, or clear the checkbox to display the marker in the same format as the preceding text</li> </ul> <p>Click on the <b>OK</b> button; the editor inserts the endnote marker at the current cursor location and displays the marker and text at the end of the section or, if there are no sections, at the end of the document.</p>	
<b>Edit endnote text</b>	<p>Select the <b>Edit   Edit Footnote/Endnote   Edit Endnote Text</b> menu option.</p> <p>The marker and text of each endnote displays in the <b>document text</b> where the marker was originally inserted, highlighted by a broken underline.</p> <p>Locate the text and make the required changes; these changes are immediately reflected in the endnotes at the end of the section or document.</p> <p>When you have finished editing endnote text, click again on the <b>Edit   Edit Footnote/Endnote   Edit Endnote Text</b> menu option; the endnote text is no longer shown in the document text.</p>	

### Notes

- You can create and edit headers and footers whether the **View | Page Layout Mode** option is switched on or switched off; this option makes no difference.

### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Reporting | RTF Templates | Add Header and Footer**
- (Alt+F1) | **Enterprise Architect | Reporting | RTF Templates | Define First Page Header**

#### 5.6.9.3.11 Create Tables

Within any document report, you can present information using simple tables with any number of rows and columns. In a report template, it is also possible to render a **model section** as a table, defined with any number of columns but with only *two* rows. You can either add your own text to the tables, or populate the table cells with fields that extract information from your model.

**Access** Select **Table** from the menu bar at the top of the Document Editor window, **or** right-click and select the **Table** option within either:

- The Content panel of the Document Template Designer or
- The text area of the document

Create and format a table

To	Action	See also
<b>Insert a table in the document</b>	<p>Position the cursor on the point in the text to create the table, and select the <b>Table   Insert Table</b> menu option.</p> <p>The New Table Parameters dialog displays, in which you specify the number of table rows and columns. For a model section table in a report template you can accept the default values of two rows and three columns. Click on the <b>OK</b> button.</p> <p>Your table might be invisible; if so, select the <b>Table   Show Gridlines</b> menu option to reveal the table and cell borders in dotted lines. These lines are for guidance in creating the document, and do not display on the printed document. You can add printable borders if you wish, using other context menu options (below).</p> <p>The editor initially creates a table with cells of equal width across the page; you can change the cell width by dragging the cell borders using the mouse, or using other context menu options (below).</p>	<a href="#">Create Sections as Tables</a> <small>[2700]</small>
<b>Add a header row</b>	<p>Select the top row of the table, and then select the <b>Table   Header Row</b> menu option. Apply any heading text, settings and formatting to the highlighted row.</p> <p>In the document or compiled report, the heading row is repeated at the top of the columns each time the table flows on to a new page. In a report, if the heading row is populated with the values from field names, the heading rows on subsequent pages reflect the values from that first row.</p> <p>This option operates on the single row at the top of the table, and not on multiple rows. If you edit the table and add a row to the top of the table to act as a new header, selecting the menu option on the new row clears the previous setting.</p>	
<b>Insert a new row above the current row</b>	Click on a row and then select the <b>Table   Insert Row</b> menu option.	
<b>Insert a new column to the left of the current column</b>	Click on a column and then select the <b>Table   Insert Column</b> menu option.	
<b>Merge cells</b>	<p>Highlight the cells to merge and select the <b>Table   Merge Cells</b> menu option.</p> <p>The width and/or height of the resulting cell is the sum of the widths and/or the sum of the heights of the merged cells.</p> <p>You can merge cells across a row, down a column, and in a block spanning both rows and columns.</p>	

To	Action	See also
<b>Select a column</b>	<p>Click on a cell and select the <b>Table   Select Current Column</b> menu option.</p> <p>The whole column is highlighted and selected for further formatting.</p>	
<b>Split a cell</b>	<p>Select the cell to split and select the:</p> <ul style="list-style-type: none"> <li>• <b>Table   Split Cell Horizontally</b> menu option to split the selected cell into two cells of equal width within the column</li> <li>• <b>Table   Split Cell Vertically</b> menu option to split the cell into two cells of equal height, doubling the height of the whole row</li> </ul> <p>Any text in the original cell is assigned to the first cell, and the second cell is created empty.</p>	
<b>Delete cells</b>	<p>Select the cells to delete and select the <b>Table   Delete Cells</b> menu option.</p> <p>The Delete Table Cells dialog displays, on which you specify whether to delete:</p> <ul style="list-style-type: none"> <li>• <b>Cells</b> - deletes the highlighted cells; realign the column and/or row after doing this</li> <li>• <b>Columns</b> - deletes the column or columns containing the highlighted cells, or</li> <li>• <b>Rows</b> - deletes the row or rows containing the highlighted cells</li> </ul> <p>If you delete all cells in a table, the table itself is automatically deleted.</p>	
<b>Position the table on the page</b>	<p>Click on any part of the table and select the <b>Table   Row Position</b> menu option.</p> <p>The Table Row Alignment dialog displays, on which you select a radio button to left-justify, center or right-justify the table on the page.</p> <p>Leave the <b>All Rows in the Table</b> checkbox selected.</p> <p>This menu option has greater effect if the table is narrower than the page or text column.</p>	
<b>Set the height of a row, or all rows</b>	<p>Highlight the rows to adjust and select the <b>Table   Row Height</b> menu option.</p> <p>The Row Height Parameters dialog displays; you can select to:</p> <ul style="list-style-type: none"> <li>• Allow the row height to be determined automatically</li> <li>• Set a minimum row height, or</li> <li>• Set an exact row height</li> </ul> <p>If you are setting the height, type the value in twips.</p> <p>You can apply the setting to the selected rows only, or select the <b>Apply to all rows in the current table</b> checkbox to set the height of all rows in</p>	

To	Action	See also
	the table.	
<b>Keep row text together if it continues over a page</b>	<p>Select the rows to protect (preferably all rows in the table) and select the <b>Table   Keep Row Together</b> menu option.</p> <p>If the row continues over the end of the page, the whole row is moved to the top of the next page.</p>	
<b>Set the text flow in the table rows</b>	<p>Select the rows and select the <b>Table   Row Text Flow</b> menu option.</p> <p>The Table Text Flow dialog displays, on which you select a radio button to set the text flow in the rows to:</p> <ul style="list-style-type: none"> <li>• <b>Left to right</b></li> <li>• <b>Right to left</b></li> <li>• <b>Default</b></li> </ul> <p>You can apply the text flow to just the selected rows, or select the <b>Apply to all rows in the current table</b> checkbox to set the text flow for all rows in the table.</p> <p>This option also moves the selected rows over to the appropriate side of the page or column.</p>	<a href="#">Format Paragraphs</a> <small>[1064]</small>
<b>Set the width of selected cells</b>	<p>Select the cells to act on and select the <b>Table   Cell Width</b> menu option.</p> <p>The Set Cell Width dialog displays, on which you set the width of the cells and the margin between the text and the left and right sides of each cell, both in twips. You can apply the settings to:</p> <ul style="list-style-type: none"> <li>• All cells in the table</li> <li>• The selected cells only</li> <li>• The complete column or columns containing the selected cells, or</li> <li>• The complete row or rows containing the selected cells</li> </ul>	
<b>Define the width of the cell borders</b>	<p>Select the cells to act on and select the <b>Table   Cell Border Width</b> menu option.</p> <p>The Set Cell Border dialog displays, on which you set the width of the lines at any or all of the top, bottom, left and right of the selected cells. If you want the border to enclose:</p> <ul style="list-style-type: none"> <li>• The selected cells as a block, select the <b>Draw outline border around the selected cells</b> checkbox</li> <li>• Each selected cell separately, leave the checkbox unselected</li> </ul> <p>You can also set the margin between the text in each cell and the left and right borders, in twips, and apply all the settings to:</p> <ul style="list-style-type: none"> <li>• All cells in the table</li> </ul>	

To	Action	See also
	<ul style="list-style-type: none"> <li>The selected cells only</li> <li>The complete column or columns containing the selected cells, or</li> <li>The complete row or rows containing the selected cells</li> </ul>	
<b>Define the color of the cell borders</b>	<p>Select the cells to act on and select the <b>Table   Cell Border Color</b> menu option.</p> <p>The Set Cell Border Color dialog displays, on which you set the color of each of the lines at the top, bottom, left and right of a cell, or all of the cell borders at once.</p> <p>You can apply the settings to:</p> <ul style="list-style-type: none"> <li>All cells in the table</li> <li>The selected cells only</li> <li>The complete column or columns containing the selected cells, or</li> <li>The complete row or rows containing the selected cells</li> </ul>	
<b>Define the degree of shading on the cell color</b>	<p>Select the cells to act on and select the <b>Table   Cell Shading</b> menu option.</p> <p>The Cell Shading Parameters dialog displays, in which you type the shading percentage.</p> <p>Shading is the intensity of the cell background color, expressed as a percentage, and is uniform across the selected cells. If the cells have no colored background, a shading of <b>0%</b> is white. For colored backgrounds, <b>0%</b> is the color initially set. In all cases, a shading of <b>100%</b> is black.</p> <p>You can apply the setting to:</p> <ul style="list-style-type: none"> <li>All cells in the table</li> <li>The selected cells only</li> <li>The complete column or columns containing the selected cells, or</li> <li>The complete row or rows containing the selected cells</li> </ul>	
<b>Define the background color of the cell</b>	<p>Select the cells to act on and select the <b>Table   Cell Color</b> menu option.</p> <p>The Cell Color Parameters dialog displays. Click on the <b>Set Color</b> button to display the Color dialog, on which you select a basic color or define a more specific color.</p> <p>When you have selected the color, click on the <b>OK</b> button.</p> <p>You can apply the setting to:</p> <ul style="list-style-type: none"> <li>All cells in the table</li> <li>The selected cells only</li> <li>The complete column or columns containing the selected cells, or</li> </ul>	

To	Action	See also
	<ul style="list-style-type: none"> <li>The complete row or rows containing the selected cells</li> </ul>	
<b>Vertically align text in cells</b>	<p>Select the cells to act on and select the <b>Table   Cell Vertical Align</b> menu option.</p> <p>The Cell Vertical Alignment dialog displays, on which you select to align the text in the selected cells by the top, center or bottom of the cell or the baseline of the lowest top line of text.</p> <p>Text in a table defaults to being aligned with the top of the cell and, as you type it in, it scrolls downwards. You can change this alignment to the center or bottom of the cell so that text scrolls out from the center or up from the bottom of the cell.</p> <p>If you have special formatting - such as 'before-paragraph' spacing - you can align the text to the base of the lowest first line (the <b>baseline</b>), so that text in a row of cells is aligned with the special formatted text rather than with the top or bottom of the cell. If the special formatting is changed or removed, the text remains aligned.</p> <p>You can apply the setting to:</p> <ul style="list-style-type: none"> <li>All cells in the table</li> <li>The selected cells only</li> <li>The complete column or columns containing the selected cells, or</li> <li>The complete row or rows containing the selected cells</li> </ul>	<a href="#">Format Paragraphs</a> <small>110641</small>
<b>Rotate the text of a cell to display vertically</b>	<p>Select the cells to act on and select the <b>Table   Cell Rotate Text</b> menu option.</p> <p>The Cell Text Rotation dialog displays, on which you select to display text vertically up the cell, vertically down the cell or, if the text is already vertical, horizontally across the cell,</p> <p>Text in a table defaults to running horizontally across the cell from left to right, and scrolling down the cell from top to bottom. You can change the text position so that the text is rotated through 90 degrees to display:</p> <ul style="list-style-type: none"> <li>From top to bottom, scrolling across the cell from right to left, or</li> <li>From bottom to top, scrolling across the cell from left to right</li> </ul> <p>If the text is running horizontally, you can type as much as you need - the row height increases to accommodate the text. If the text is running vertically, you adjust the row height and column width manually to accommodate your text.</p> <p>You can apply the setting to:</p> <ul style="list-style-type: none"> <li>All cells in the table</li> <li>The selected cells only</li> <li>The complete column or columns containing the selected cells, or</li> <li>The complete row or rows containing the selected cells</li> </ul>	



To	Action	See also

### Notes

- Under some circumstances, a table in a generated report might repeat the header row rather than the output row; if this occurs, create another row in the table between the header row and the output row, and leave this blank

#### 5.6.9.3.12 Apply User-Defined Section Numbering

In your report template, you might want to number the section levels, with a numbering format of your own design. For example:

```

1. Package Level 1
  1.1 Package Level 2 (child package)
    1.1.1 Element Level 1
      1.1.1.1 Element (child element)

```

To define the numbering format you:

- First create a numbering list, then
- Create a set of list overrides for this list

In the overrides, you change the initial 1.0.0 setting to 1.1.1. You can then apply the numbering list to the headings set for packages and elements, using paragraph numbering.

**Access** Right-click in the Content panel of the Document Template Designer and select **Edit | List and Overrides**

### Define the numbering format

Step	Action	See also
1	<p>To create the numbering list, select the <b>Create List Item</b> menu option.</p> <p>The List Properties dialog displays:</p> <ul style="list-style-type: none"> <li>In the <b>List Name</b> field, type a name for the list</li> <li>Leave the <b>Multi-level list</b> checkbox selected</li> <li>If you have added sections to your document and want to restart numbering for each new section, select the <b>Restart at section break</b> checkbox</li> <li>Click on the <b>OK</b> button to close the dialog</li> </ul>	
2	<p>To create the list override, select the <b>Create List Override</b> menu option.</p> <p>The List Override Properties dialog displays:</p> <ul style="list-style-type: none"> <li>In the <b>List to Override</b> field, type or select the name of the list you have just created</li> <li>Leave the <b>Override Levels</b> checkbox selected</li> </ul>	

Step	Action	See also
	<ul style="list-style-type: none"> <li>Click on the <b>OK</b> button to close the dialog</li> </ul>	
<b>3</b>	<p>To set up the list level properties for a level, select the <b>Edit List Level</b> menu option. The List Level properties dialog displays.</p> <ul style="list-style-type: none"> <li>Select the <b>List item</b> radio button and type or select the list item you have just created</li> <li>Set the <b>List Level</b> field to <b>1</b> (for the <i>Package Section</i>) and the <b>Number text</b> field to <b>~1~</b></li> <li>Click on the <b>OK</b> button to save the values and close the dialog</li> </ul>	
<b>4</b>	<p>Select the <b>Edit List Level</b> menu option again to re-open the dialog:</p> <ul style="list-style-type: none"> <li>Select the <b>List item</b> radio button and type or select the list item you have just created</li> <li>Set <b>List Level</b> to <b>2</b> (for the <i>Element Section</i> or <i>Child Package Section</i>, for example)</li> <li>Set <b>Start at</b> to <b>1</b> (to ensure that numbering at this level begins at 1.1 rather than 1.0)</li> <li>Click on the <b>OK</b> button to close the dialog and save the changes</li> </ul>	
<b>5</b>	Repeat steps 3 and 4 for each additional list level to need, incrementing the <b>List Level</b> number and resetting <b>Start at</b> to <b>1</b> each time.	

#### Apply the numbering levels defined above

Step	Action	See also
<b>1</b>	<p>In the Content panel of the Document Editor, select the first item of text to be numbered (for example, <i>Package</i>).</p> <pre> package&gt;&gt;   Package: {Pkg.Name}    element&gt;&gt;   Element: {Element.Name}   child-elements&gt;&gt;   &lt;-child-elements   &lt;-element   child-packages&gt;&gt;   &lt;-child-packages   &lt;-package   </pre>	
<b>2</b>	Set the text style to one of the Heading styles ( <b>Heading 1</b> to <b>Heading 9</b> ), using the style drop-down field in the Document Editor toolbar.	

Step	Action	See also
	<p><b>package-&gt;</b>  <b>Package:-{Pkg.Name}</b> ¶</p>	
3	<p>Right-click on the text and select the <b>Paragraph   List Numbering</b> menu option.  The Apply paragraph numbering using Lists dialog displays.</p> <ul style="list-style-type: none"> <li>In the <b>List</b> panel, select the required List and Override combination</li> <li>Set the <b>Level</b> field to the required level (1, for the top level)</li> <li>Click on the <b>OK</b> button to close the dialog, and check that the required level has been applied to the selected text</li> </ul> <p><b>package-&gt;</b>  <b>1. Package:-{Pkg.Name}</b> ¶</p>	
4	<p>Repeat step 3 for the next level (Element), but change the <b>Level</b> field to 2.</p> <p><b>element-&gt;</b>  <b>1.1 Element:-{Element.Name}</b> ¶</p>	
5	<p>Continue applying the overrides for each lower section level as necessary, then generate your document.</p> <p>The output will be numbered and formatted, as illustrated by this example:</p> <p>1. <b>Package: Formal Requirements</b>  1.1 <b><i>Package: Manage Users</i></b>  1.1.1 <b>Element: REQ011 - Manage User Accounts</b>  1.1.2 <b>Element: REQ016 - Add Users</b>  1.2 <b><i>Package: Manage Inventory</i></b>  1.2.1 <b>Element: REQ019 - Manage Inventory</b>  1.2.2 <b>Element: REQ020 - Receive Books</b>  1.2.3 <b>Element: REQ021 - List Stock Levels</b></p>	

#### 5.6.9.3.13 Insert Reference Links

Within a document, you can insert two types of reference link:


- A **bookmark**, to locate material inside the document, either from within the same document or from a separate Word document
- A **hyperlink**, to display material held outside the document, namely an external document, Help topic or web page

**Access** Select **Insert** from the menu bar at the top of the Document Editor window, **or** right-click and select **Insert** within either:

- The Content panel of the Document Template Designer or
- The text area of the document

### Insert Bookmarks and Hyperlinks

To	Detail	See also
<b>Manage bookmarks</b>	<p>As your <b>report template</b> develops, the editor automatically assigns a unique bookmark to the start and end of each model section you select from the Sections panel.</p> <p>You can also create your own bookmarks in <b>any</b> document wherever you want to reference a specific point, and move these to other points in the text.</p> <p>(In the report template you can move the automatic bookmarks as well, but if you want to use these as reference points for a Word document to access, it is better to leave them as they are.)</p> <p>To insert a bookmark:</p> <ul style="list-style-type: none"> <li>• Move the cursor to the point in the text to mark, and select the <b>Insert   Bookmark</b> menu option; the Bookmark dialog displays</li> <li>• Either select an existing bookmark from the list (to switch it to the current location) or type a new bookmark name in the top field</li> <li>• Click on the <b>Insert</b> button</li> </ul> <p>To locate bookmarked text in the document:</p> <ul style="list-style-type: none"> <li>• Select the <b>Insert   Bookmark</b> menu option and, on the Bookmark dialog, click on the bookmark and on the <b>Go to</b> button; the editor displays the point at which the bookmark was set, in the top left corner of the display</li> </ul> <p>To delete an existing bookmark:</p> <ul style="list-style-type: none"> <li>• Select the <b>Insert   Bookmark</b> menu option and, on the Bookmark dialog, click on the bookmark and on the <b>Delete</b> button</li> </ul> <p>To insert a reference to a bookmark:</p> <ul style="list-style-type: none"> <li>• Move the cursor to the point at which to insert the reference and, if you want to, type some lead-in text such as "Go to page " or "See page "</li> <li>• Select the <b>Insert   Bookmark</b> menu option and, on the Bookmark dialog, click on the bookmark and on the <b>Set Page Reference</b> button</li> </ul> <p>The editor inserts the number of the page containing the bookmark, at the cursor position in the text; for a document, this acts as a hyperlink to the bookmark position</p>	<p><a href="#">Setting Sections for Reporting</a> [2688]</p> <p><a href="#">Document Bookmarks</a> [2730]</p>

To	Detail	See also
	If you set the reference in a template, the link is created in the document you generate from the template	
<b>Insert hyperlinks</b>	<p>To insert a hyperlink, right-click on the point in the text at which to create it and select the <b>Insert   Hyperlink</b> context menu option; the Insert Hyperlink dialog displays.</p> <p>In the <b>Link Text</b> field, type the text to be hyperlinked and, in the <b>Link Code</b> field, type or paste the:</p> <ul style="list-style-type: none"> <li>• Web page URL</li> <li>• Help topic file name (add the prefix <b>\$Help:/</b>) or</li> <li>• External file path and name</li> </ul> <p>To capture the help topic file name, right-click on the displayed topic, select the <b>Properties</b> context menu option, and copy the file name.</p> <p>Click on the <b>OK</b> button; the hyperlinked text displays in the document. Edit the sentence to accommodate the link text.</p> <p>To display the web page, help page or external document:</p> <ul style="list-style-type: none"> <li>• Linked in a document, double-click on the link</li> <li>• Linked in a report template, compile the report and click on the hyperlink in the report</li> </ul> <p>In a Team Review document, you can also insert hyperlinks to related objects in the model.</p>	<a href="#">Add Object Links</a> 

#### 5.6.9.3.14 Insert Images, Objects and Frames

In creating your document or template, you might want to embed or display images and other objects in the text, to insert external information and/or structure the text and graphics on a page. You could insert, for example:

- An image in the text or as background to the text
- A link to an image in the text
- An Object Linking and Embedding (OLE) object, which you create and edit in the document using an external application
- A frame - a rectangular border within the text that can contain separate text and pictures, and that the existing text flows **around**
- A drawing object - a rectangle, text box or line that can sit **over** the existing text
- Another document file

Having inserted an object, you can subsequently move, edit or delete it.

**Access** Select options from the menu bar at the top of the Document Editor window, **or** right-click within either:

- The Content panel of the Document Template Designer or
- The text area of the document

Insert items

To	Action	See also
<b>Insert a picture in the document</b>	<p>You can insert a picture bitmap or Windows metafile in your document as either an embedded picture or a link to the external picture file.</p> <p>Position the cursor at the point at which to insert the picture, and select either:</p> <ul style="list-style-type: none"> <li>• <b>Insert   Embed Picture</b> ( F8 ) or</li> <li>• <b>Insert   Link Picture</b></li> </ul> <p>A browser dialog displays, through which you select the picture to insert in the document; the picture is shown at the current cursor location.</p> <p>The <b>embedded</b> picture is saved within the document; only the filename of the <b>linked</b> picture is saved with the document, which is a better option if you intend to edit the picture over time.</p>	
<b>Embed an OLE object in the text</b>	<p>Position the cursor at the point at which to embed the object, and select the <b>Insert   OLE Object</b> menu option.</p> <p>The Insert Object dialog displays, listing the applications that are available to create the object.</p> <p>When you select an application, the editor launches it and you create the object; when you save the application, the editor inserts an icon or view that represents the inserted object.</p> <p>You can also use the <b>Edit   Paste Special</b> menu option to import OLE objects, provided that the object is available in the clipboard.</p> <p>You can later edit the object using the application, by double-clicking on the object or by selecting the <b>Edit   Edit OLE Object</b> menu option.</p>	<a href="#">Scroll, Search and Select Text</a> <sup>[1059]</sup>
<b>Change the size and alignment of an embedded picture</b>	<p>Click on the picture and select the <b>Edit   Edit Picture</b> menu option; the Edit Current Picture Parameters dialog displays, through which you can change the height and width of the picture, in inches.</p> <p>You can also align the top, middle, or bottom of the picture with the baseline of any text on the same line as the picture, or offset the bottom of the picture from that baseline by any number of twips.</p>	
<b>Insert a background or watermark picture for the document pages (document report templates)</b>	<p>Select the <b>Other   Watermark Picture</b> menu option; a browser dialog displays, through which you select the graphic file to insert as a watermark. The picture occupies the entire text area, and is displayed as a washed-out image. If you choose an image that is already washed-out, no watermark is displayed at all.</p> <p>To remove the watermark image, deselect the <b>Watermark Picture</b> menu option.</p>	

To	Action	See also
<b>Insert a document File</b>	Position the cursor at the point at which to insert the file, and select the <b>Insert   Insert RTF File</b> menu option.	
<b>Embed a frame at the cursor position</b>	<p>Select the <b>Insert   Frame</b> menu option.</p> <p>The border of the frame displays, with the cursor at the top left corner. You can change the size of the frame by clicking on the border and dragging the small black 'handles'.</p>	
<b>Embed a drawing object at the cursor position</b>	<p>Select the <b>Insert   Drawing Object</b> menu option.</p> <p>The Select Drawing Object dialog displays, on which you select the radio button for either a:</p> <ul style="list-style-type: none"> <li>• Text Box</li> <li>• Rectangle or</li> <li>• Line</li> </ul> <p>Click on the <b>OK</b> button; the selected object displays, overlaid on the existing document content. If you wish, you can change the size of the object by clicking on the border and dragging the small black 'handles'.</p> <p>You can also change the appearance and the relationship of the object to the document text (see below).</p>	
<b>Move the frame or drawing object</b>	<p>Click inside the outline and then move the cursor just outside the outline so that the cursor changes to a plus-shape.</p> <p>Drag the plus shape (and hence the outline) to the new position.</p>	
<b>Edit the border, background and impact on document text of a drawing object</b>	<p>Select the <b>Edit   Edit Frame/Drawing Object   Edit Drawing Object</b> menu option.</p> <p>The Line and Fill Attributes dialog displays.</p> <p>Select the options to define:</p> <ul style="list-style-type: none"> <li>• Whether to display a border line and, if so, your preferred border thickness, type and color</li> <li>• Whether the object is transparent or filled and, if filled, the fill color</li> <li>• The wrapping behavior of the text outside the object: <ul style="list-style-type: none"> <li>• <b>Wrap Through</b> causes the document text to flow through the diagram object so that they overlap each other</li> <li>• <b>No Wrap</b> causes the document text to skip over the object, so that the text resumes below the object</li> <li>• <b>Wrap Around the Object</b> causes the document text to flow around the object, into a column on one side or the other of the object, or on both sides; if you move the object across the page, the editor tends to leave a wider</li> </ul> </li> </ul>	

To	Action	See also
	<p>margin before it starts to wrap text to the left of the object</p> <ul style="list-style-type: none"> <li>The Z Order of the object, so that if there is an overlap the object either lies in front of the document text (<b>Z-order = 1</b>) or behind the document text (<b>Z Order = &gt;1</b>)</li> </ul>	
<b>Insert text into the frame, text box or rectangle</b>	<p>Click inside the outline and type the text at the cursor position. The text wraps at the right hand side of the box and, for a:</p> <ul style="list-style-type: none"> <li>Frame, extends the bottom border of the box</li> <li>Text box and rectangle, overflows the bottom of the box but is still associated with the box and not with the 'normal' text of the document; to enclose all lines adjust the height manually to enclose the text</li> </ul> <p>If you change the <b>width</b> of the outline, any text inside the outline is automatically wrapped to adjust to the new width.</p>	
<b>Rotate text to display it vertically in the frame or text box</b>	<p>Select the <b>Edit   Edit Frame/Drawing Object   Rotate Text</b> menu option. The Text rotation dialog displays; select the radio button for the text direction you want to apply:</p> <ul style="list-style-type: none"> <li><b>Horizontal</b> (default) - the text flows from the top left corner of the box across the width of the box, wrapping at the right border and scrolling downwards</li> <li><b>Top to Bottom</b> - the characters are rotated through 90 degrees to flow from the top right corner of the box downward, wrapping at the bottom edge of the box and scrolling right to left</li> <li><b>Bottom to Top</b> - the characters are rotated through 90 degrees to flow from the bottom left corner of the box upward, wrapping at the top edge of the box and scrolling left to right</li> </ul> <p>If the text is running horizontally, you can type as much as you need - the box height increases to accommodate the text. If the text is running vertically, and it flows outside the box, you adjust the box height and width manually to accommodate your text.</p> <p>Click on the <b>OK</b> button to apply the change.</p>	
<b>Insert a picture into a frame or drawing object text box</b>	<p>Click inside the outline and either <b>Insert</b> the picture (see above) or copy and paste it at the cursor position.</p> <p>If the frame or text box contains only the picture, the picture size is automatically adjusted to as nearly fill the outline as possible without distorting the image.</p> <p>If you change the width or height of the box, the picture is adjusted to completely fill the outline; you continue manually adjusting the outline to reduce the distortion of the image.</p>	



To	Action	See also
<b>Lock a frame or drawing object to a point on the page</b>	<p>You can fix the position of the outline to a specific point so that as you enter and format the text the outline maintains its position relative to that point.</p> <p>The default position is relative to the <b>first character of the paragraph</b>, in which case the outline moves on the page with any change that affects the position of the paragraph.</p> <p>You can also fix the position relative to either the:</p> <ul style="list-style-type: none"> <li>• <b>Top edge of the page</b> or</li> <li>• <b>Top margin of the page</b></li> </ul> <p>In these cases, if the document text on the page is adjusted it flows <b>past</b> the outline.</p> <p>Click on the outline and select the <b>Edit   Edit Frame/Drawing Object   Vertical Base Position</b> menu option.</p> <p>The Object Position parameters dialog displays. Select the radio button for the locking point you prefer:</p> <ul style="list-style-type: none"> <li>• <b>Top of the Page</b></li> <li>• <b>Top Margin</b></li> <li>• <b>Current Paragraph</b></li> </ul> <p>Click on the <b>OK</b> button to apply the lock.</p>	
<b>Delete a frame or drawing object</b>	<p>Click on the outline and press ( <b>Delete</b> ).</p> <p>The editor prompts you to confirm the deletion; click on the <b>Yes</b> button.</p> <p>This deletion is <b>reversible</b> - press ( <b>Ctrl+Z</b> ) or select the <b>Edit   Undo</b> menu option.</p>	

#### 5.6.9.3.15 Checking Text

In developing a document or template, you would probably want to spell-check the text that you have entered; for example, to make sure that 'boiler-plate' text is correct, as it will be used many times in many reports. In the Document Editor you can both spot-correct a word from a context-sensitive list, or run a spell checker over the whole document. You can also check the use of a word against an online thesaurus, to see if there is a more precise term or to check the opposite meaning of the word.

**Access** Select options from the menu bar at the top of the Document Editor window, **or** right-click within either:

- The Content panel of the Document Template Designer or
- The text area of the document

#### Check Spelling

To	Action	See also
<b>Spot-correct a word</b>	<p>Select the <b>Other   Auto Spell</b> option.</p> <p>When the <b>Auto Spell</b> option is on, the Document Editor automatically checks the spelling of all <b>typed</b> or <b>pasted</b> text in the document (but not field names or generated text). Words that are apparently spelt incorrectly are underlined in red.</p> <p>Right-click on an underlined word; a short list displays, containing the correct spelling of the assumed word and a selection of other words that this word might have been intended to be. Click on:</p> <ul style="list-style-type: none"> <li>• One of the words, to use it in place of the mis-spelt word</li> <li>• The <b>Ignore All</b> option to use the word as it is in this context</li> <li>• The <b>Add</b> option to add the word to the suggested list of spellings/other words</li> </ul> <p>You might be using a language for which there is no spelling dictionary installed, or text that contains a high proportion of new terms, in which case you might not want to have the auto spell function underlining all the text. You can turn it off by selecting the <b>Other   Auto Spell</b> option again.</p>	
<b>Run the spell checker over the whole document</b>	<p>Select the <b>Other   Spell Check</b> menu option.</p> <p>The Check Spelling dialog displays, identifying the first apparent spelling mistake in the document.</p> <p>You might want to modify the things that the Spell Checker identifies as errors; click on the <b>Options</b> button to tailor the operation of the checker.</p> <p>Having tailored the spell checker, respond to each of the possible errors that it highlights in your document.</p> <p>When you have acted on all the words presented by the Spell Checker (or clicked on the <b>Cancel</b> button) a status message displays identifying the number of words picked up and the number of those words you have corrected. Click on the <b>OK</b> button.</p>	<p><a href="#">Select Spell Checker Options</a> <sup>[550]</sup></p> <p><a href="#">Correcting Words</a> <sup>[554]</sup></p>

### Run Thesaurus

To	Action	See also
<b>Access the Thesaurus</b>	<p>Right-click on a word in the text and select the <b>Other   Thesaurus</b> menu option.</p> <p>The Thesaurus dialog displays, showing the:</p> <ul style="list-style-type: none"> <li>• Selected word</li> <li>• Word-categories the word belongs to, in the Categories panel</li> <li>• List of <b>synonyms</b> for the word within the first category (words that have the same meaning as the selected word) in the Synonyms panel</li> </ul>	

To	Action	See also
	If there are several categories, click on each one to display the synonyms within each category.	
<b>Display antonyms</b>	<p>If you want to display the <b>antonyms</b> for the word (the opposite meanings of the word), select the <b>Antonyms</b> checkbox.</p> <p>The <i>Antonyms</i> title displays in the Categories panel, with the word-categories for the opposite of the selected word listed underneath.</p> <p>Click on a category name to list the antonyms in that category, in the Synonyms panel.</p>	
<b>Check for homonyms</b>	<p>A homonym is a word that sounds like another and can be confused with it. For example, <i>Sympathy</i> might be confused with <i>Empathy</i> or <i>Symphony</i>.</p> <p>If you want to check that you have not chosen a sound-alike word by mistake, click on the <b>Suggest</b> button.</p> <p>The Suggestions dialog displays, listing possible sound-alike words. If you want to use one of those words, or check for another with the same meaning, click on the <b>Look up</b> button.</p> <p>The Thesaurus dialog refreshes with the new word you have selected.</p>	
<b>Search for new terms</b>	<p>If you want to examine a totally different word, in the <b>Term</b> field type the new word and click on the <b>Look up</b> button.</p> <p>The display changes to show the synonyms (and antonyms) of the new word.</p>	
<b>Change to a synonym</b>	<p>Click on a synonym (or an antonym, if you wish) in the list, and click on the <b>OK</b> button.</p> <p>The word in the text changes to the selection.</p>	

#### Learn more

- [Design Custom Document Templates](#) <sup>[2684]</sup>

#### 5.6.9.3.16 Track Changes

In a team environment, two or more people might work on developing documents and/or templates. You might, therefore, need to review each other's work, or just be able to see what changes another person has made to the file. In the Document editor, you can turn on change tracking to highlight the changes you make to the text.

**Access** Select **Edit | Track Changes** from the menu bar at the top of the Document Editor window, **or** right-click and select **Edit | Track Changes** within either:

- The Content panel of the Document Template Designer or
- The text area of the document

### Tracking options

To	Action	See also
<b>Turn tracking on</b>	<p>Select the <b>Track Changes   Enable Tracking</b> menu option.</p> <p>Any changes made after Tracking is turned on are immediately highlighted:</p> <ul style="list-style-type: none"> <li>• Inserted text - printed in a different color</li> <li>• Deleted text - struck through with a single colored line</li> <li>• Formatting changes - indicated by a colored dotted underline</li> </ul> <p>A change bar is also inserted in the page or column margin to indicate the line on the page on which the change was made.</p> <p>If you mouse over a highlighted change, a tooltip displays the type of change, the name of the person who made the change, and the date and time the change was made.</p>	
<b>Turn tracking off</b>	<p>Select the <b>Track Changes   Enable Tracking</b> menu option again.</p> <p>Any changes made after Tracking was turned on remain highlighted, but changes made after tracking was turned off are not highlighted.</p>	
<b>Hide changes</b>	<p>If you want to review the document as it is, without the highlighted changes, but you do not want to lose the changes or highlighting, select <b>Track Changes   Hide Changes</b> menu option.</p> <p>The highlighted additions and deletions are hidden, and the remaining text closes up. Format changes are not hidden.</p> <p>To expose the hidden changes, click on the <b>Track Changes   Hide Changes</b> menu option again.</p>	
<b>Locate changes</b>	<p>Select the <b>Track Changes   Find Next Change</b> menu option or the <b>Track Changes   Find Previous Change</b> menu option to search the text and place the cursor on the next or previous change made, respectively.</p>	
<b>Respond to change</b>	<p>If you want to assess each change and act on your decision, select the:</p> <ul style="list-style-type: none"> <li>• <b>Track Changes   Accept Change</b> menu option to confirm the change, or</li> <li>• <b>Track Changes   Reject Change</b> menu option to reverse the change to the original text or format</li> </ul> <p>In either case, the change tracking markings are removed. You can</p>	

To	Action	See also
	make further changes if you wish.	
<b>Accept all changes</b>	<p>If you are not assessing each change, for example if you just want to see what a colleague has done before making your own changes, you can accept all changes across the document in one action.</p> <p>Select the <b>Track Changes   Accept All Changes</b> menu option.</p>	
<b>Reject all changes</b>	<p>If you are not assessing each change, but just want to reverse all changes across the document in one action, select the <b>Track Changes   Reject All Changes</b> menu option.</p> <p>A prompt displays for you to confirm the action; click on the <b>Yes</b> button.</p> <p>You might do this if, for example, you or a colleague have edited the wrong version of a document or acted on a technical change that was itself reversed.</p>	

#### 5.6.9.3.17 Protect Document Contents

When you have created a document, or some important text within a document, you might want to protect it from being edited or deleted so that it is safe from accidental change. The Document Editor defaults to locking **against** protection, so you first turn off the lock and then protect either the whole document or a selected block of text. Once protection is set, neither you nor any other user can change the text without specifically removing the protection.

**Access** Select options from the menu bar at the top of the Document Editor window, **or** right-click within either:

- The Content panel of the Document Template Designer or
- The text area of the document

#### Protect text

To	Action	See also
<b>Enable protection</b>	<p><b>De</b>-select the <b>Other   Protection Lock</b> context menu option.</p> <p>A prompt displays for you to confirm the removal of the lock. Click on the <b>Yes</b> button.</p> <p>You can now set protection on your text.</p> <p>To reset the protection lock and block protection of <b>further</b> text, select the <b>Other   Protection Lock</b> context menu option again. The text that you have <b>already</b> protected, however, remains protected and cannot be edited until you or another user specifically removes the protection. You cannot remove protection from this text until you turn off the protection lock again.</p>	
<b>Protect text</b>	Highlight the line, block or section of text to be protected, and select	

To	Action	See also
	<p>the <b>Font   Protect</b> context menu option.</p> <p>The selected text is displayed in a grey font, and cannot be edited.</p> <p>If the menu option is unavailable, or the text is not faded, it means that you have not removed the protection lock (above).</p> <p>To release the text for editing again, highlight it and reselect the <b>Font   Protect</b> option. You can select and release a portion of the protected text; the remainder stays protected.</p>	
<b>Protect a document</b>	<p>Within the document or template, select the <b>Other   Protect Form</b> context menu option.</p> <p>The entire document is protected.</p> <p>An exception is protecting a template in the Document Template Designer. As you select checkboxes in the Sections panel, section markers are still added to the Content panel. You cannot, however, add fields or text between the markers.</p> <p>To release the document for editing again, reselect the <b>Other   Protect Form</b> context menu option.</p>	<a href="#">Setting Sections for Reporting</a> <span>[2688]</span>

#### 5.6.9.3.18 Print Report Documents

When you are editing a document or template, you can manage print jobs within the Document Editor. You can select from options to define the page layout, text to be printed and number of copies, and to check the appearance of the printing before actually submitting the print job.

**Access** Select **File** from the menu bar at the top of the Document Editor window, **or** right-click and select **File** within either:

- The Content panel of the Document Template Designer or
- The text area of the document

#### Print options

To	Action	See also
<b>Set the page margins for viewing and printing the document</b>	<p>Select the <b>File   Page Layout</b> menu option.</p> <p>The Page Parameters dialog displays, on which you can define the left, right, top and bottom page margins in inches.</p> <p>The <b>Top</b> and <b>Bottom</b> margins define how much space is set aside to contain header and footer text.</p> <p>In the <b>Header distance from the Page Top</b> and <b>Footer distance from the Page Bottom</b> fields, type the separation of the header and footer text from the edge of the paper.</p>	

To	Action	See also
<b>Set the parameters for the print run</b>	<p>Select the <b>File   Printer Setup</b> menu option, or press ( <b>Ctrl+Shift+P</b> ).</p> <p>The Print Setup dialog displays for the default printer, which you would have selected on the <b>Windows Control</b> panel.</p> <p>You can simply confirm or reselect the core parameters such as page size, paper source and page orientation, or click on the <b>Properties</b> button to define more advanced settings such as use of a watermark, duplex printing and print resolution.</p>	
<b>Preview document before printing</b>	<p>Select the <b>File   Print Preview</b> menu option to display the document as it would appear when printed.</p> <p>The editor displays either one page or two - toggle the display by clicking on the <b>One Page</b> or <b>Two Pages</b> buttons. Scroll to further pages using the scroll bar.</p> <p>By default the preview rectangle is sized to fit the current window; you can use the <b>Fitted</b> field to enlarge or shrink the size of the displayed pages as required.</p> <p>Click on the <b>Print</b> button to display and use the Print dialog, as described below.</p> <p>To return to editing mode click on the <b>Edit</b> button, or on the <b>File   Print Preview</b> menu option again.</p>	
<b>Print your document</b>	<p>Select the <b>File   Print</b> menu option, or press ( <b>Ctrl+P</b> ).</p> <p>The Print dialog displays, on which you can select the pages to print and the number of copies, and initiate the print task.</p> <p>You can also choose to print only a selected part of the file. To do this, highlight the required text before selecting the <b>Print</b> option. You can print a highlighted block of:</p> <ul style="list-style-type: none"> <li>• Lines, or</li> <li>• Characters</li> </ul>	

#### 5.6.9.3.19 Hyperlink From Linked Document

Within the text of your linked document, you can create hyperlinks to other elements in the model, and to web pages, Help files, Model Searches and Team Review Forums. When you next open the linked document, you can double-click on the hyperlink to open the linked files or to locate and highlight the object in the Project Browser. You can perform all normal operations on the object, including opening any linked document **that** element might have.

#### Create hyperlinks

Task	Action	See also
<b>Create Hyperlinks to elements</b>	<p>Click on the element in the Project Browser and drag it into the linked document to the position at which to create the hyperlink. The Linked Document Editor automatically creates the hyperlink, using the object name as the hyperlink text. You can edit this text if you prefer.</p>	

Task	Action	See also
	<p><b>Alternatively:</b></p> <p>Highlight the link text in the linked document, right-click on it and select the <b>Create   Link to Existing Element</b> context menu option. The Select Classifier dialog displays, through which you browse for and select the element to link to.</p>	<a href="#">Select &lt;Item&gt; Dialog</a> <sup>[994]</sup>
<b>Create hyperlinks to other documents</b>	<p>Highlight the appropriate text and select the <b>Create   Hyperlink</b> context menu option.</p> <p>The Hyperlink Details dialog displays, through which you specify the type of document to link to, and its location. When specifying the object location, you can use full paths or local (path substitution) paths.</p> <p>In some cases you might not want to have normal text as the hyperlink, but the text of the link itself. In this case, do not highlight any text or, in Hyperlink Details dialog, type any link text in. The system then inserts the <b>Address</b> path as the link text.</p> <p>In this context you do not need to provide an Alias or Notes.</p>	<a href="#">Hyperlinks</a> <sup>[2002]</sup>

#### Learn more

- [Edit Linked Documents](#)<sup>[1048]</sup>
- [Insert Reference Links](#)<sup>[1083]</sup> (for an alternative method of creating a hyperlink to an external document, Help file or web page)
- [Model Searches](#)<sup>[709]</sup>
- [Team Review](#)<sup>[343]</sup>
- [The Web Browser](#)<sup>[170]</sup>

#### **5.6.9.3.20 Create Element From Document**

Whilst you can set up hyperlinks from the linked document to **existing** elements and diagrams in the Project Browser, you can also **create new** elements to illustrate and support information written in the linked document. You can create and link to any type of element or diagram, but the facility has specific options for the following element types:

- Class
- Requirement
- Issue

The element or diagram is created in the same package as the element for which the linked document was created. A hyperlink is automatically created in the document to the new object. When you click on the hyperlink, the element or diagram is highlighted in the Project Browser.

#### Create element or diagram



Step	Action	See also
1	Open the linked document in the model element or Document Artifact element.	<a href="#">Document Artifacts</a> <sup>[1046]</sup> <a href="#">Create Linked Document on an Element</a> <sup>[1047]</sup>
2	Enter or select the appropriate text to act as the hyperlink text (such as the element or diagram name).  Highlight this text and right-click on it; the Editor context menu displays.	
3	Select the <b>Create   New</b> menu option, and one of the submenu options. If you select the: <ul style="list-style-type: none"> <li>• <b>Class, Requirement or Issue</b> option, the corresponding element is immediately created in the Project Browser</li> <li>• <b>Other</b> option, the New Element dialog displays; specify the element type and - if appropriate - stereotype, and click on the <b>Create</b> button</li> <li>• <b>Diagram</b> option, the New Diagram dialog displays; specify the diagram type and click on the <b>OK</b> button</li> </ul>	<a href="#">New Elements</a> <sup>[903]</sup> <a href="#">New Diagrams</a> <sup>[822]</sup>
4	The highlighted text is now a hyperlink to the new object in the Project Browser. Click on the link to highlight the object.  You can now edit or expand the element or diagram as required.	

#### Learn more

- [Class](#) <sup>[1363]</sup>
- [Requirement](#) <sup>[1763]</sup>
- [Issues \(Defects\)](#) <sup>[2631]</sup>
- [Hyperlink From Linked Document](#) <sup>[1095]</sup>

#### 5.6.9.4 Replace or Delete Documents

If a linked document is out of date you can edit the text, or you can:

- Delete the content and replace it from another file, or
- Delete the document altogether

#### Replace the document content

Step	Action	See also
1	Either:	

Step	Action	See also
	<ul style="list-style-type: none"> <li>Right-click on the element and select the <b>Edit Linked Document</b> menu option, or</li> <li>Click on the element and press <b>Ctrl+Alt+D</b></li> </ul> <p>The linked document opens in the Linked Document Editor.</p>	
2	Click in the body of the document and press <b>Ctrl+A</b> to select all the document text.	
3	Press <b>Delete</b> .	
4	<p>Right-click and select the <b>File   Import</b> context menu option.</p> <p>The Windows Open dialog displays, in which you can browse for the file to import into the document.</p>	
5	Click on the <b>Save</b> icon in the Linked Document Editor toolbar.	

#### Delete the linked document

Step	Action	See also
1	Right-click on the element in the Project Browser or diagram, and select the <b>Delete Linked Document</b> context menu option.	
2	A prompt displays for you to confirm the deletion; click on the <b>Yes</b> button.	

#### Learn more

- [Edit Linked Documents](#) <sup>[1048]</sup>

### 5.6.9.5 Create Linked Document Templates


To create a linked document, you can open an empty file and add the text and contents as you want. You can **also** base your linked document on a linked document **template** that you either:

- Select from a drop-down list of (initially) system-provided templates in the New Linked Document from Template dialog or
- Create in the *Document Generation > Linked Document Templates* folder of the Project Resources window

Linked document templates that you create are added to the drop-down list on the New Linked Document from Template dialog.

**Access** **Project | Resources** (Alt+6)

### Create a template

Step	Detail	See also
1	In the Resources window: 1. Expand the <i>Document Generation</i> folder. 2. Right-click on the <i>Linked Document Templates</i> folder. 3. Select the <b>Create Template</b> context menu option. The New Document Template dialog displays.	
2	In the <b>New template</b> field, type a name for your template.	
3	In the <b>Copy template</b> field, if you want to base the new template on an existing template, click on the drop-down arrow and select the name of the source template.  Otherwise leave the field set to <b>None</b> .	
4	Click on the <b>OK</b> button.  The template opens in the Linked Document Template Editor in the Diagram View. Create and edit the template contents as you need.	<a href="#">Edit Linked Document Templates</a> 
5	Save the template ( <b>File   Save</b> ) and close the editor ( <b>File   Close</b> ). The template is immediately available for use.	
6	If you want to assign your template to a group, right-click on the template name in the Resources window and select the <b>Assign Template to Group</b> context menu option.  The Enter the Model Document Template Group Name dialog displays. In the <b>Enter Value</b> field type a category name and click on the <b>OK</b> button.  The template is immediately transferred to the named group.	

### Notes

- You can transport customized linked document templates between models, using the **Export Reference Data** and **Import Reference Data** options

### Learn more

- [Resources](#) 
- [Export Reference Data](#) 

- [Import Reference Data](#)<sup>[380]</sup>

### 5.6.9.6 Edit Linked Document Templates







To develop the contents of a linked document template, held in the Resources window, you use the Linked Document Template Editor. This is based on the Document Editor, providing a wide range of convenient features for formatting and editing the template, and for adding links from and references to the content of the document. The editor facilities are provided from a menu displayed at the top of the editor.

**Access** **Project | Resources > Document Generation > Linked Document Templates ( > template group) > template name | Modify Template**

#### Option Descriptions

The Document Template Editor menu option descriptions are grouped according to the actions you are performing.



Actions	Link
Creating and importing documents	<a href="#">File Control</a> <sup>[1050]</sup>
Configuring the editor page display and formatting tools shown	<a href="#">Editor Tool Display Options</a> <sup>[1052]</sup>
Incorporating stylesheets, special texts and Tables of Contents Managing the base styles in the Normal.rtf style template file	<a href="#">Styles, Special Texts &amp; Table of Contents</a> <sup>[1054]</sup> <a href="#">The Normal.rtf Style Template</a> <sup>[1058]</sup>
Moving through, searching and selecting text	<a href="#">Scroll, Search and Select Text</a> <sup>[1059]</sup>
Formatting characters and text strings	<a href="#">Format Text</a> <sup>[1062]</sup>
Formatting paragraphs and text blocks	<a href="#">Format Paragraphs</a> <sup>[1064]</sup>
Inserting tab points	<a href="#">Set Tabs</a> <sup>[1067]</sup>
Inserting sections, columns and page breaks, and repaginating	<a href="#">Define Document Sections</a> <sup>[1069]</sup>
Inserting and editing page headers and footers, and footnotes and endnotes	<a href="#">Insert Headers, Footers, Footnotes and Endnotes</a> <sup>[1071]</sup>
Inserting tables	<a href="#">Create Tables</a> <sup>[1075]</sup>

Actions	Link
Inserting hyperlinks and bookmarks	<a href="#">Insert Reference Links</a> 
Checking spelling and use of terms	<a href="#">Checking Text</a> 
Tracking, accepting and rejecting changes to text	<a href="#">Track Changes</a> 
Protecting template text from accidental change	<a href="#">Protect Document Contents</a> 
Inserting images, OLE objects, frames and drawing objects	<a href="#">Insert Images, Objects and Frames</a> 
Printer setup and printing documents	<a href="#">Print Report Documents</a> 

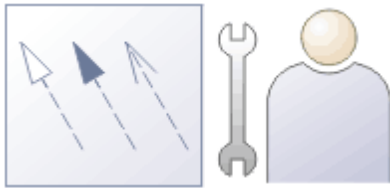
### Notes

- Throughout your template editing:
  - To undo one or more immediately previous edits, press **(Ctrl+Z)**, or select the **Edit | Undo** menu option; you can still undo a change even after you have saved the change
  - To redo one or more immediately previous undone edits, press **(Ctrl+Y)**, or select the **Edit | Redo** menu option

### Learn more

- [Create Linked Document Templates](#) 
- [Resources](#) 

## 5.7 Connectors



UML connectors, along with elements, form the basis of a UML model. Connectors link elements together to denote some kind of logical or functional relationship between them. UML supports a wide range of connectors, each having its own purpose, meaning and notation, and each used in specific kinds of UML diagrams.

### Learn more

- [Connector Management Options](#) <sup>[1102]</sup>
- [Connector Tasks](#) <sup>[1108]</sup>
- [Connector Properties](#) <sup>[1126]</sup>
- [BPMN](#) <sup>[1862]</sup>
- [Connector Style Options](#) <sup>[1106]</sup>

### 5.7.1 Connector Management Options

If you right-click on a connector in a diagram, the connector context menu displays. This provides quick access to some important functions.

### Topics

Topic	Detail	See also
<b>Context Menu</b>	<p>The menu is split into up to seven distinct sections:</p> <ul style="list-style-type: none"> <li>• <b>Extensions</b> - displays in the first section only if you have Add-Ins installed and registered, such as Eclipse</li> <li>• <b>Properties</b></li> <li>• <b>Type Specific</b></li> <li>• <b>Advanced</b></li> <li>• <b>Style</b></li> <li>• <b>Appearance</b></li> <li>• <b>UML Help</b> - Displays the Enterprise Architect Help topic for this connector type</li> <li>• <b>Delete</b> - delete the connector with this option</li> </ul>	<p><a href="#">Manage Connector Properties</a> <sup>[1103]</sup></p> <p><a href="#">Connector Type-Specific Options</a> <sup>[1103]</sup></p> <p><a href="#">Advanced Connector Management Options</a> <sup>[1105]</sup></p> <p><a href="#">Connector Style Options</a> <sup>[1106]</sup></p> <p><a href="#">Connector Appearance Options</a> <sup>[1107]</sup></p>
<b>Connector Role Context Menu</b>	<p>For connectors with Roles, right-clicking a connector within up to 60 pixels of an end point displays a role-specific context menu.</p> <p>The Role context menu has three additional menu options:</p> <ul style="list-style-type: none"> <li>• A <b>Source/Target Role...</b> menu option that opens the</li> </ul>	<p><a href="#">Source Role</a> <sup>[1130]</sup></p> <p><a href="#">Connect To Element Feature</a> <sup>[1110]</sup></p>

Topic	Detail	See also
	<p>connector specification dialog with the respective role page selected</p> <ul style="list-style-type: none"> <li>• A <i>Multiplicity</i> submenu that enables you to set the multiplicity for the role</li> <li>• A <b>Link to Element Feature</b> menu option that displays a dialog through which you can attach the end of the connector to a specific attribute or operation</li> </ul>	

#### Notes

- Context menus vary slightly between connector types, and not all menu options are present on all connector context menus; the type-specific menu options are not always included

### 5.7.1.1 Manage Connector Properties

The *Properties* section of the connector context menu contains the following options:

Menu Option	Action	Shortcut	See also
<b>&lt;Connector type&gt; Properties</b>	Open the Properties window for the selected connector.		<a href="#">Properties Window</a> <sup>[1126]</sup>
<b>Advanced</b>	Display the <b>Advanced</b> menu.		<a href="#">Advanced Connector Management Options</a> <sup>[1105]</sup>
<b>Attach Note or Constraint</b>	Attach a note or constraint to the connector.		<a href="#">Attach a Note or Constraint</a> <sup>[1111]</sup>

#### Notes

- Context menus vary slightly between connector types, and not all menu options are present on all connector context menus; the type-specific menu options are not always included

### 5.7.1.2 Connector Type-Specific Options

The *Type-Specific* section of the connector context menu is specific to the object, and only appears for a few different connectors. Some examples are shown below:

Connector	Menu Option	Action	See also
Transition	<b>Locate in State Table</b>	Switch a State Machine diagram to a State Table, with the selected Transition cell highlighted on the table.	
Transition	<b>Message</b>	Set the value of the Message.	

Connector	Menu Option	Action	See also
Transition	<b>Find Triggers Associated</b>	Identify hidden triggers, and locate a trigger in the Project Browser; if there: <ul style="list-style-type: none"> <li>Is one trigger, it is immediately highlighted in the Project Browser</li> <li>Are two or more triggers the Element Usage dialog displays, listing the triggers; double-click on the required trigger to highlight it in the Project Browser</li> </ul>	
Transition	<b>Queue Triggers in Simulation</b>	(Available when a simulation is being executed.)  Adds the Trigger to the <b>end</b> of the list in the Simulation Events tab of the Breakpoints & Events window, without signalling it.	<a href="#">Simulation Events</a> <sup>[2504]</sup>
Transition	<b>Signal Triggers in Simulation</b>	(Available when a simulation is being executed.)  Adds the Trigger to the Simulation Events tab of the Breakpoints & Events window and signals it at the <b>current step</b> of the simulation, leaving it with the status <i>signalled</i> .	<a href="#">Waiting Triggers</a> <sup>[2508]</sup>
Association	<b>Find Association Class</b>	Locate and highlight the Association Class element in the Project Browser.	
Aggregation	<b>Set Aggregation to Composite</b>	Change the Aggregation to Composite.	
Aggregation	<b>Set Aggregation to Shared</b>	Set the Aggregation to shared; this appears after <b>Set Aggregation to Composite</b> has been selected.	
Information Flow	<b>Find Items Conveyed</b>	(Listed only if there are Information Items on the connector.)  Locate the Information Item in the Project Browser; if there: <ul style="list-style-type: none"> <li>Is one item, it is immediately highlighted in the Project Browser</li> <li>Are two or more items the Element Usage dialog displays, listing the items; double-click on the required item to highlight it in the Project Browser</li> </ul>	









**Notes**

- Context menus vary slightly between connector types, and not all menu options are present on all connector context menus; the type-specific menu options are not always included

**5.7.1.3 Advanced Connector Management Options**

The *Advanced* section of the connector context menu contains the following options:

Menu Option	Action	Shortcut	See also
<b>Set Source and Target</b>	Change the source and/or target of the connector.		<a href="#">Change the source and/or target</a>  <sup>[1113]</sup>
<b>Change Type</b>	Change the connector type.		<a href="#">Change the connector type</a>  <sup>[1113]</sup>
<b>Change Direction</b>	Keep the <b>ends</b> of the connector the same but change the <b>direction</b> setting. There are four options: <ul style="list-style-type: none"> <li><b>Unspecified</b></li> <li><b>Source -&gt; Destination</b></li> <li><b>Destination -&gt; Source</b></li> <li><b>Bi-Directional</b></li> </ul>		
<b>Reverse Direction</b>	Reverse the <b>ends</b> of the connector; for example, if the connector is an arrow, the arrowhead swaps to the other end.  This command maintains the <b>direction</b> of the connector (such as <i>Source -&gt; Destination</i> ).		
<b>Specialize Associations</b>	Specify how the properties of this Association specialize the properties of other Associations.		
<b>Information Items Conveyed</b>	Add information items to an Information Flow connector.		<a href="#">Information Item Conveyed</a>  <sup>[1413]</sup>  <a href="#">Information Flow</a>  <sup>[1410]</sup>
<b>Information Flows Realized</b>	Realize any information items conveyed on an Information Flow connector between two elements.		<a href="#">Information Item Realized</a>  <sup>[1414]</sup>
<b>Dependency Stereotypes</b>	Select a stereotype for the Dependency (or Trace, Role Binding, Occurrence or Represents connector).		



Menu Option	Action	Shortcut	See also
<b>Extension Points</b>	<p>On an Extend connector between two Use Cases, provides options to:</p> <ul style="list-style-type: none"> <li>Specify which Extension Point (if any) in the target Use Case the connector and its source Use Case relate to</li> <li>Show or hide the Note element that identifies the defined Extension Point</li> </ul>		<a href="#">Extend</a> 


### Notes

- Context menus vary slightly between connector types, and not all menu options are present on all connector context menus; the type-specific menu options are not always included

#### 5.7.1.4 Connector Style Options

The *Style* section of the connector context menu provides the following options:

Action	Usage	Shortcut	See also
<b>Line Style</b>	<p>Set the connector line style; the options are:</p> <ul style="list-style-type: none"> <li>Direct</li> <li>Auto Routing</li> <li>Custom</li> <li>Bezier</li> <li>Tree (Horizontal, Vertical, Lateral Horizontal or Lateral Vertical)</li> <li>Orthogonal (with Square or Rounded corners)</li> </ul>		<a href="#">Set the connector line style</a> 
<b>Pin End(s)</b>	<p>Pin the connector start and/or end to the current position on the target element; a sub-menu displays to offer the options of pinning the start point only, the end point only, or both.</p> <p>Once one or both ends are pinned, a fourth option is available to unpin both ends.</p>		
<b>Bend Line at Cursor</b>	Insert an anchor point on the line at the point of the cursor so you can change the shape of the line.	<b>Ctrl+Q</b>	<a href="#">Insert an anchor point</a> 
<b>Suppress Line Segment</b>	<p>Hide a segment of a connector so that you can view a part of the diagram that it crosses.</p> <p>To reverse the change, right-click on the connector and select the <b>Show All Line Segments</b> context menu option.</p>		


Action	Usage	Shortcut	See also
<b>Straighten Line at Cursor</b>	Remove an anchor point on the line at the point of the cursor.  (This is the exact opposite of <b>Bend Line at Cursor</b> , and ( <b>Ctrl+Q</b> ) toggles the connector point between the options).	<b>Ctrl+Q</b>	<a href="#">Connector Styles</a> 

#### Notes



- Context menus vary slightly between connector types, and not all menu options are present on all connector context menus; the type-specific menu options are not always included
- If two connectors cross, the later connector has a 'hoop' at the intersection, indicating that the connectors are crossing

### 5.7.1.5 Connector Appearance Options

The *Appearance* section of the connector context menu provides the following options:

Menu Option	Action	Shortcut	See also
<b>Appearance</b>	Set the line color and line thickness of the connector.		
<b>Visibility</b>	Set connector visibility; see the table below for sub-menu options.		
<b>Tidy Line Angles</b>	Tidy the line angles of a custom connector.		<a href="#">Tidy the line angles</a> 

#### Visibility Sub-Menu:

Menu Option	Action	Shortcut	See also
<b>Hide Connector</b>	Hide the connector.  To show the connector again, follow the steps in the <i>Hide/Show Connectors</i> topic.		<a href="#">Hide/Show Connectors</a> 
<b>Hide Connector in Other Diagrams</b>	Hide or show the connector in other diagrams.		<a href="#">Hide/Show Connectors</a> 
<b>Hide All Labels</b>	Hide or show all labels attached to the connector.		

Menu Option	Action	Shortcut	See also
<b>Set Label Visibility</b>	Hide or show labels attached to the connector, individually.		<a href="#">Hide or show labels</a> <sup>[1122]</sup>

### Notes

- Context menus vary slightly between connector types, and not all menu options are present on all connector context menus; the type-specific menu options are not always included

## 5.7.2 Connector Tasks

This topic details some of the tasks associated with managing model connectors, such as:

- [Connect Elements](#) <sup>[1109]</sup>
- [Connect to an Element Feature](#) <sup>[1110]</sup>
- [Change Connector Styles](#) <sup>[1114]</sup>
- [Arrange Connectors](#) <sup>[1112]</sup>
- [Change Connector Type](#) <sup>[1113]</sup>
- [Create Connector in Project Browser](#) <sup>[1118]</sup>
- [Reverse Connector](#) <sup>[1123]</sup>
- [Delete Connectors](#) <sup>[1119]</sup>
- [Hide/Show Connectors](#) <sup>[1127]</sup>
- [Hide/Show Labels](#) <sup>[1122]</sup>
- [Create Generalization Set](#) <sup>[1120]</sup>
- [Change the Source or Target Element](#) <sup>[1113]</sup>
- [Set Relation Visibility](#) <sup>[1119]</sup>
- [Add a Note to a Connector](#) <sup>[1111]</sup>
- [Use Tree Style Hierarchy](#) <sup>[1125]</sup>
- [Create Connector in Project Browser](#) <sup>[1118]</sup>
- [Show Uses Arrow Head](#) <sup>[1125]</sup>
- [Set Association Specializations](#) <sup>[1123]</sup>
- [Change Sequence Message Scope](#) <sup>[1124]</sup>

### Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions, if security is enabled, you must have **Update Element** permission to update or delete a connector

### Learn more

- [List of Available Permissions](#) <sup>[329]</sup>

### 5.7.2.1 Connect Elements

#### Topics

Topic	Detail	See also
<b>Connect Elements on a Diagram</b>	<p>The fastest and simplest ways to create connectors are using the Quick Linker and using the Toolbox.</p> <p>To create another connector of the same type as the last connector you used, click on the appropriate source element and press ( <b>F3</b> ).</p>	<a href="#">Create Connectors In Place Using the Quick Linker</a> <sup>[898]</sup> <a href="#">Create Connectors Using the Toolbox</a> <sup>[792]</sup> <a href="#">Design Patterns</a> <sup>[1464]</sup> <a href="#">Using UML Profiles</a> <sup>[1472]</sup>
<b>Selecting Connectors</b>	<p>To select a connector, simply click on it; drag handles display, indicating that the connector is selected.</p> <p>This gives the connector focus for keyboard commands such as ( <b>Delete</b> ), and displays connector properties in docked windows such as the Tagged Values window.</p> <p>If there is more than one connector on a diagram, you can cycle through them using the arrow keys.</p>	
<b>Drag Connectors</b>	<p>You can drag a connector to position it; click on the connector and drag it to where it is to appear - note that there are some limitations on how far or to where you can drag a connector.</p> <p>You can also reattach the end of a connector to a different source or target element.</p>	<a href="#">Change the Source or Target Element</a> <sup>[1113]</sup> <a href="#">The Element Browser</a> <sup>[989]</sup>
<b>Connector Properties and Commands</b>	<p>You can double-click on a connector to change properties, or right-click to display the context menu containing commands to change connector type and direction.</p> <p>You can also highlight the connectors on a specific element; select the element and press and hold ( <b>L</b> ) - all the connectors issuing from or terminating at that element are highlighted in separate colors. This is a useful feature to apply in conjunction with the context filtering facility on an element on a diagram.</p>	<a href="#">Connector Properties</a> <sup>[1126]</sup> <a href="#">Change Connector Type</a> <sup>[1113]</sup> <a href="#">Reverse Connector</a> <sup>[1123]</sup> <a href="#">Context Filter a Diagram</a> <sup>[789]</sup>
<b>Create Connectors Without a Diagram</b>	<p>Sometimes it is useful to create relationships between elements without a diagrammatic representation; you can do this using the Project Browser and the Relationship Matrix.</p>	<a href="#">Add Connectors With the Project Browser</a> <sup>[1118]</sup> <a href="#">Add Connectors With the Relationship Matrix</a> <sup>[737]</sup>

#### Notes

- You can reposition a connector by selecting and dragging the connectors as required
- If a connector has source and target roles, you can attach either end of the connector to a specific attribute or operation in the source or target element

#### Learn more

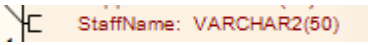
- [Connect to Element Feature](#) 

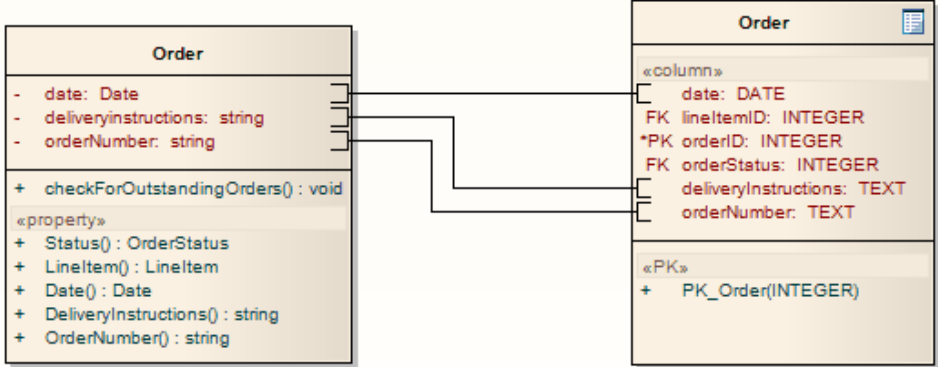
### 5.7.2.2 Connect to Element Feature

If a connector has source and target roles, you can connect either end of the connector to a specific operation or attribute in the source or target element. This is entirely a visual aid, to indicate which features are significant in the relationship. In code generation or transformation, the link is interpreted as a normal source-element to target-element relationship.

#### How to

To connect the end of the connector to a specific operation or attribute

Step	Action	See also
1	Right-click on the end of the connector joined to the element containing the required feature.	
2	Select the <b>Link to Element Feature</b> context menu option. The Link to Element Feature dialog displays.	
3	In the <b>Feature Type</b> field, click on the drop-down arrow and select the required feature type - <b>Attribute</b> or <b>Operation</b> . The attributes or operations from the element are listed in the <b>Feature</b> field.	
4	Click on the required attribute or operation, and click on the <b>OK</b> button. The end of the connector changes to a bracket next to the selected feature.  You might create a number of feature-to-feature relationships between two elements (such as a Class and a Table that represents the Class data) to produce a diagram similar to the following:	

Step	Action	See also
	 <p>You can change the feature to which the connector is attached by following the above procedure and selecting the new feature.</p> <p>You can break the link to the selected feature in the following ways:</p> <ul style="list-style-type: none"> <li>• Follow the above procedure, setting the <b>Feature Type</b> to <b>None</b></li> <li>• Delete the attribute or operation from the element</li> <li>• Change the connector type to a type that does not have source and target roles</li> <li>• Change the connector to a different source or target element that does not contain the feature</li> </ul> <p>Note that reversing the direction of the connector does not break the connector's attachment to the feature.</p>	

### 5.7.2.3 Add a Note to a Connector

This topic describes how you can connect notes and constraints to graphical relationships.

Notes enable you to provide explanations and further detail for one or more connectors on a diagram, with a visible Note element.

#### How to

To add a note or constraint to a connector

Step	Action	See also
1	Right-click on one of the connectors to attach a note to. The context menu displays.	
2	Select the <b>Attach Note or Constraint</b> menu option. The Link Relations dialog displays.	

Step	Action	See also
3	Select the appropriate radio button in the Link Type panel.	
4	Check all the connectors that participate in the set.	
5	Click on the <b>OK</b> button to complete the note or constraint creation.	
6	Use the normal Note dialog to enter the appropriate text for the note or constraint.	

### Notes

- The constraint note is drawn slightly differently to a regular note, and has { and } automatically added to visually indicate the constraint form
- As an alternative to the procedure above, you can drag the Quick Linker arrow on an existing Note or Constraint, and drag it to the required connector; a Notelink is automatically created to attach the Note or Constraint to the connector


### Learn more

- [Quick Linker](#) 

## 5.7.2.4 Arrange Connectors

Connectors between two elements can be moved around the element borders to create a good layout. There is a limit to how much a connector can be moved around, but generally it is very easy to find an acceptable layout. For the best layouts, use the *custom* line style; this enables you to add as many line points and bends as you require to create a clean and readable diagram.

### Move and arrange connectors on a diagram

Step	Action	See also
1	Click once on the connector to select it.	
2	Holding the mouse button down, move the connector in the required direction.	
3	To refine the movement, click and hold very near to one end of the connector; this enables a slightly different movement range.	
4	To further refine the movement and range, select either a routed, direct or custom line style; each behaves slightly differently.	<a href="#">Connector Styles</a> 



### 5.7.2.5 Change Connector Type

#### How to

To change a connector type

Step	Action	See also
1	In the Diagram View, right-click on the connector to change. The context menu displays.	
2	Select the <b>Connection Detail   Change Type</b> menu option.	
3	In the <b>Connector Type</b> field, click on the drop-down arrow and select the required connector type.	
4	Click on the <b>OK</b> button to apply changes.	

### 5.7.2.6 Change the Source or Target Element

After you have created a connector between two elements, you might later want to change either the source or target. Instead of deleting and re-creating the connector, Enterprise Architect enables you to change the source or target. There are two ways of doing this: using the Set Source and Target dialog or using the mouse.

#### How to

To change the source or target element of a connector using the Set Source and Target dialog

Step	Action	See also
1	Right-click on the connector to open the context menu.	
2	Select the <b>Advanced   Set Source and Target</b> menu option. The Set Source and Target dialog displays.	
3	Click on the drop-down arrows on the <b>From Element</b> and <b>To Element</b> fields, and select the source and target elements.	
4	Click on the <b>OK</b> button to apply changes.	

To change the source or target element of a connector using the mouse

Step	Action	See also
1	Click on the connector and position the cursor over the 'handle' at one end.	
2	When the cursor changes, click the mouse and drag the handle to the new element.	

#### Notes

- The connector does not actually move until you release the mouse button over the new source or target element; however:
  - A dotted line shows where the connector would be during the move
  - The solid outline of the nearest element or extension changes to a hatched outline as you move the cursor onto it; this helps you identify where the connector should connect to, if there are many closely-arranged elements, Parts, Ports and other extensions

#### 5.7.2.7 Connector Styles

Connectors come in seven different routing styles. Additionally, if one solid-line connector crosses another, the second one 'jumps' over the first (unless the **Enable Connector line jumps** option is deselected on the Options dialog).

#### Reference

Style	Description	See also
<b>Direct</b>	A straight line from element A to element B.  You can move the line (back and forward, up and down) to a limited degree.	
<b>Auto Routing</b>	A vertical and horizontal route from A to B with 90-degree bends.  You can move the line to improve the route, but the location and number of bends are not configurable.	
<b>Bezier</b>	A smooth curved line from A to B.  Bezier style is directly available for Data Flow diagram connectors, Mind Mapping connectors, State Flows, State Transitions, Object Flows, and Control Flows. <ul style="list-style-type: none"> <li>You can convert other types of relationship to Bezier style by assigning the Tagged Value <code>_Bezier</code>, with an integer value other than <b>0</b>; however, some relationship types (such as Aggregate) do not accommodate this style very well</li> <li>This Tagged Value over-rides the value of the <b>Style</b> field in the connector Properties dialog</li> </ul>	<a href="#">Connector Properties</a> <small>[1128]</small>

Style	Description	See also
<b>Custom Line</b>	The most flexible option; you can add one or more line points and bend and push the line into virtually any shape, using the <b>Toggle Line Point at Cursor</b> option.	
<b>Tree Style - Vertical</b> <b>Tree Style - Horizontal</b>	A line from element A to B with two right-angle bends, and the end points fixed to selected locations on the elements (Vertical or Horizontal). <ul style="list-style-type: none"> <li>You can convert relationships to Tree Style by assigning the Tagged Value <code>_TreeStyle</code>, with a value of <b>H</b> (Horizontal) or <b>V</b> (Vertical)</li> <li>This Tagged Value over-rides the value of the <b>Style</b> field in the connector Properties dialog</li> </ul>	
<b>Lateral - Vertical</b> <b>Lateral - Horizontal</b>	A line from element A to B with a single right-angle bend, and the end points fixed to selected locations on the elements (Vertical or Horizontal). <ul style="list-style-type: none"> <li>You can convert relationships to Lateral style by assigning the Tagged Value <code>_TreeStyle</code>, with a value of <b>LH</b> (lateral-horizontal) or <b>LV</b> (lateral vertical)</li> <li>This Tagged Value over-rides the value of the <b>Style</b> field in the connector Properties dialog</li> </ul>	
<b>Orthogonal - Square</b> <b>Orthogonal - Rounded</b>	You can add one or more line points and bend and push the line into a variety of shapes, using the <b>Toggle Line Point at Cursor</b> option.  All line segments thus created are either vertical or horizontal.	

### How to

To set the connector style

Step	Action	See also
<b>1</b>	Right-click on the connector to change. The context menu displays.	
<b>2</b>	Select the <b>Line Style</b> option.	
<b>3</b>	From the submenu, select the required style: <ul style="list-style-type: none"> <li><b>Direct</b></li> <li><b>Auto Routing</b></li> <li><b>Custom</b></li> <li><b>Tree</b></li> </ul>	

Step	Action	See also
	<ul style="list-style-type: none"> <li>• <b>Lateral</b></li> <li>• <b>Orthogonal</b></li> <li>• (or <b>Bezier</b>, where appropriate)</li> </ul>	

*Alternatively:*

Step	Action	See also
<b>1</b>	Select the connector to change.	
<b>2</b>	Press the following keys to change the style: <ul style="list-style-type: none"> <li>• ( <b>Ctrl+Shift+D</b> ) for Direct</li> <li>• ( <b>Ctrl+Shift+A</b> ) for Auto Routing</li> <li>• ( <b>Ctrl+Shift+C</b> ) for Custom</li> <li>• ( <b>Ctrl+Shift+Z</b> ) for Bezier (where appropriate)</li> </ul>	

To bend a connector to quickly and easily route connectors in the required layout

Step	Action	See also
<b>1</b>	Right-click on the connector. The context menu displays.	
<b>2</b>	Set the line style to Custom Line ( <b>Ctrl+Shift+C</b> ); this enables the <b>Bend Line at Cursor</b> option in the context menu.	
<b>3</b>	Click on the <b>Bend Line at Cursor</b> option to add a line point. Right-clicking a line point displays the <b>Straighten Line at Cursor</b> context menu option, which you can use to remove the line point.	
<b>4</b>	Using the mouse, drag the line point to the required position.	

*Alternatively:*

Step	Action	See also
1	Hold down ( <b>Ctrl</b> ) or ( <b>Shift</b> ) and click on a point on the connector to create a line point.  ( <b>Ctrl</b> )+click also removes a line point.	
2	Using the mouse, drag the line point to the required position.	

To tidy line angles (custom connector)

Step	Action	See also
1	Right-click on the connector.  The context menu displays.	
2	Click on the <b>Tidy Line Angles</b> menu option; this nudges the custom line in horizontal and vertical increments, saving you the time of trying to get a good layout manually.  You can set the <b>Tidy Line Angles</b> option to operate by default; click on the <b>Tools   Options</b> menu option to display the Options dialog, and select the Diagram Behavior page.	

To suppress individual line segments

Step	Action	See also
1	Right-click on the connector.  The context menu displays.	
2	Set the line style to Custom Line ( <b>Ctrl+Shift+C</b> ).  This enables the <b>Suppress Line Segment</b> option in the context menu.	
3	Click on the <b>Suppress Line Segment</b> option to suppress a line between two bend points.  The segment you right-clicked on is suppressed.	
4	To show the segment again, right-click on the line and click on the <b>Show All Line Segments</b> context menu option.	

Step	Action	See also
	<p>One application for this is to represent the continuation of flow when your diagram crosses the page boundary marker in the Diagram View.</p> <p>When you suppress the line segment that crosses the boundary, the link name (connector properties) displays at both ends of the hidden segment; when you print the diagram on multiple pages, the link name identifies the connection apparently broken by the page boundary.</p>	

### 5.7.2.8 Create Connector in Project Browser

You can create a connector from one element to another directly in the Project Browser.

#### How to

To connect elements from the Project Browser

Step	Action	See also
1	<p>In the Project Browser, either:</p> <ul style="list-style-type: none"> <li>Right-click on the element to create a connector for, and select the <b>Add   Create Link</b> context menu option, or</li> <li>Select the element, press ( <b>Insert</b> ) and select the <b>Create Link</b> context menu option</li> </ul> <p>The Create Link dialog displays.</p>	
2	In the <b>Direction</b> field, click on the drop-down arrow and select the direction of the new connector ( <b>Outgoing</b> means this element is the source).	
3	In the <b>Link Type</b> field, click on the drop-down arrow and select the type of connector.	
4	<p>In the <b>Choose target(s)</b> list, click on the name of the target.</p> <p>(If necessary, in the <b>Select Target Type</b> field click on the drop-down arrow and select a feature to list only elements having that feature).</p>	
5	Click on the <b>OK</b> button to create the connector.	

#### Notes

- You can also reproduce an existing connector between two elements when you paste those elements from the Project Browser into a diagram as instances; an option enables you to copy just the elements, or the relationship as well

Learn more

- [Pasting From the Tree](#)<sup>[833]</sup>

**5.7.2.9 Relationship Visibility**

You can change the visibility of individual connectors or relationships, diagram by diagram.

Access **Diagram | Advanced | Visible Relations ( Ctrl+Shift+I )**

How to

To set relationship visibility

Step	Action	See also
1	Open the diagram to change.	
2	Select the <b>Visible Relations</b> menu option. The Set Visible Relations dialog displays.	
3	Select the checkbox against each list item to show, and clear the checkbox against each item to hide. If you want to display the information in a more readable layout, you can resize the dialog.	
4	Click on the <b>OK</b> button to apply the changes.	

**5.7.2.10 Delete Connectors**How to

To delete a connector

Step	Action	See also
1	Right-click on the connector and select the <b>Delete Connector</b> context menu option. The Remove Connector dialog displays; this dialog provides the options to hide the connector so that it remains functional, or remove the connector completely.	
2	Click on the appropriate radio button and click on the <b>OK</b> button. If you select the <b>Hide</b> option, it has the same effect as hiding the connector on the Links tab of the source element Properties dialog, or using the <b>Visibility   Hide Connector</b> context menu option.	<a href="#">Hiding the Connector</a> <sup>[1121]</sup> <a href="#">Relationships</a> <sup>[742]</sup>

Step	Action	See also
	It also hides the connector on the Relationships window.	

#### Notes

- The dialog does not display if:
  - You have previously selected the **Don't ask again** checkbox or
  - On the Links page of the Options dialog (**Tools | Options | Links**) the **Prompt on connector deletes** checkbox is not selected
- Selecting the **Don't ask again** checkbox also deselects the **Prompt on connector deletes** checkbox
- Selecting the **Prompt on connector deletes** checkbox restores the dialog if you have used the **Don't ask again** checkbox
- If you hide the dialog, the **Delete Connector** context menu option defaults to the setting you last used on the dialog; make sure that you have selected the right option to default to

#### 5.7.2.11 Generalization Sets

A Generalization Set enables you to specify the relationship of a group of Generalizations. Each Generalization is a binary relationship that relates a specific Classifier to a more general Classifier (e.g. from a class to its superclasses). Each Generalization Set defines a particular set of Generalization relationships that describe the way in which a general Classifier (or superclass) can be divided using specific subtypes.

**Access** **connector context menu | Advanced | Generalization Set | New**

#### How to

To create a generalization set

Step	Action	See also
1	Select the <b>New</b> menu option. The Generalization Set dialog displays.	
2	In the <b>Name</b> field, type the name of the Generalization set; for example, <b>Gender</b> .	
3	In the <b>Power Type</b> field, either: <ul style="list-style-type: none"> <li>Type a new power type, or</li> <li>Click on the drop-down arrow or browser button ( ... ) and select an existing one from the Select &lt;Item&gt; dialog</li> </ul>	<a href="#">Select &lt;Item&gt; Dialog</a> <sup>[994]</sup>
4	Check the <b>IsMember</b> column for the child subtypes that are part of this Generalization set.	



--	--	--

### 5.7.2.12 Hide/Show Connectors

Connectors/relations that appear in multiple diagrams can be selectively shown or hidden. This makes it easier to read diagrams where elements might have many connectors, but not all are relevant in the context of the current diagram.

#### Hide or show a connector in the current diagram

Step	Action	See also
1	Double-click on the required diagram element in the Diagram view. The element Properties dialog displays.	
2	Select the Links tab. This lists the connectors linked to the element, whether or not they are hidden on the diagram.	
3	Right-click on the connector to hide or show. The context menu displays.	
4	Select the <b>Show Relation</b> option to show the hidden connector on the diagram, or the <b>Hide Relation</b> option to hide the visible connector.  Alternatively, hide a connector by right-clicking on it on the diagram and selecting the <b>Visibility   Hide Connector</b> context menu option; however, you must use the Links tab of the element Properties dialog to show the relationship again.	

#### Hide or show a connector in other diagrams

Step	Action	See also
1	Right-click on the connector in the diagram. The context menu displays.	
2	Select the <b>Visibility   Hide Connector in Other Diagrams</b> menu option. The Set Connector Visibility dialog displays.	
3	If the two connected elements have been included in other diagrams, these diagrams are	

Step	Action	See also
	<p>listed here.</p> <p>In the list, all diagrams for which the checkbox is selected show the connector; deselect the checkbox for any diagrams in which to hide the connector.</p> <p>If you want to display the information in a more readable layout, you can resize the dialog.</p> <p>To hide the connector in all of the diagrams listed, click on the <b>Suppress All</b> button.</p>	
4	Click on the <b>OK</b> button to save the changes.	

### Notes

- Certain elements, such as Requirements, do not have a Links tab in the Properties dialog

In these cases, open the Relationships window (**Element | Relationships**) for the element, right-click on the relationship in the list and select the context menu option to hide or show that relationship in the diagram

Be aware that, in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions with security on, locks on the diagram and elements can make the required option unavailable

### Learn more

- [Relationships](#) <sup>742</sup>

## 5.7.2.13 Hide/Show Labels

### How to

To hide or display one or more labels on a connector

Step	Action	See also
1	<p>Right-click on the connector.</p> <p>The context menu displays.</p>	
2	<p>Select the <b>Visibility   Set Label Visibility</b> menu option.</p> <p>The Label Visibility dialog displays; if you have several long labels, you can resize this dialog for greater clarity.</p>	
3	Select the checkbox against each label to display, and clear the checkbox against each label to hide.	

Step	Action	See also
4	Click on the <b>OK</b> button.	

#### 5.7.2.14 Connector In-place Editing Options

You can edit many of the Enterprise Architect connector labels directly on the diagram. Each label can be bound to a single connector field.

Topic	Detail	See also
<b>Putting a label in Edit Mode</b>	<p>To put a label into Edit mode, either:</p> <ul style="list-style-type: none"><li>• Select the <b>Edit Label</b> option from the context menu, or</li><li>• Select a label and press ( <b>F2</b> )</li></ul> <p>To save the current text to the field, either press ( <b>Return</b> ) or deactivate the Edit window</p> <p>To cancel edit mode without saving any changes, press ( <b>Esc</b> ).</p>	

#### 5.7.2.15 Reverse Connector

You can reverse the direction of a connector without having to delete and re-create it. This is helpful if your design changes or you add the connector wrongly to begin with.

##### How to

Step	Action	See also
1	Right-click on the incorrect connector.	
2	Select the <b>Connection Detail   Reverse Direction</b> context menu option.	

#### 5.7.2.16 Set Association Specializations

UML enables specialization of properties defined by Associations. Enterprise Architect enables this through the **Specialize Associations** option in the advanced section of the context menu for an Association.

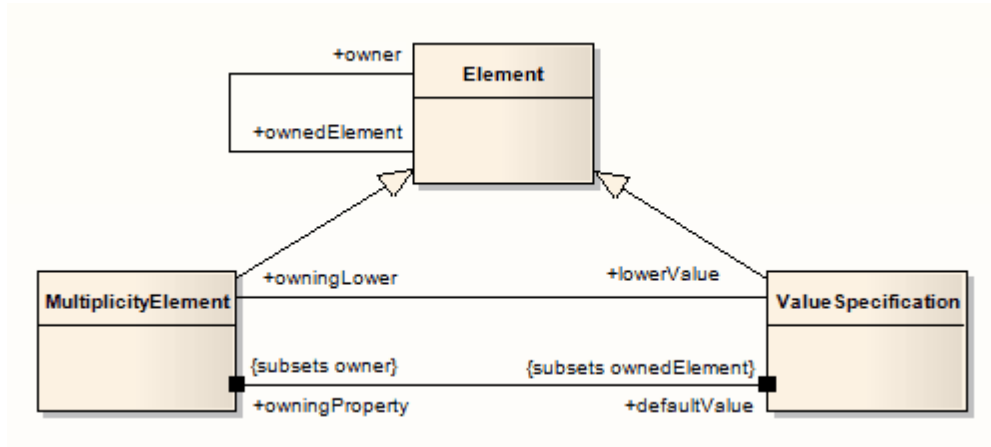
A dialog displays, showing all Associations between the two Classes connected by the current Association and their parents.

The left two columns define the source role of the current Association, while the right two define the target role. With this you are able to select the relationships of each end of the properties listed. When a relationship is set then this is drawn at the corresponding end of the connector on any diagram it appears on.

The dialog displays when you select the **Advanced | Specialize Associations** context menu option on the

lowest Association connector in the following diagram.

#### Example



#### 5.7.2.17 Change Sequence Message Scope

A message in a Sequence diagram represents a dynamic interaction from one element to another. Sometimes when you are designing your model you might have to change either the start or end point of a message as the responsibilities of elements change during design. For this reason, Enterprise Architect enables you to change the message scope by setting a new start or end element.

#### How to

Step	Action	See also
1	Select the message in the Sequence diagram.	
2	Right-click on the message to open the context menu.	
3	Select <b>Advanced   Set Source and Target</b> .	
4	In the pop up dialog, in the <b>From Element</b> and <b>To Element</b> fields, click on the drop-down arrows and select the required elements.	
5	Click on the <b>OK</b> button to save changes. The message is re-routed to meet your changed requirements.	

### 5.7.2.18 Show Uses Arrow Head

By default the *Use* connector in Use Cases has no arrow head.

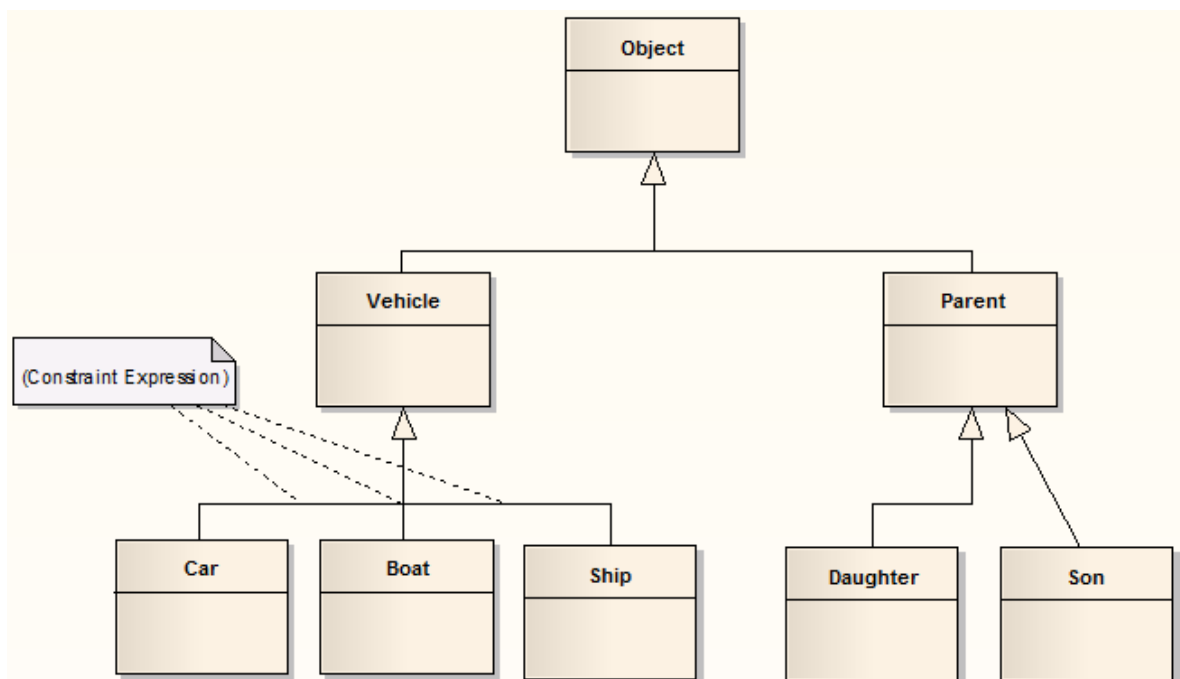
#### How to

To generate arrow heads on Use connectors

Step	Action	See also
1	Select the <b>Tools   Options   Links</b> menu option. The Links page of the Options dialog displays.	
2	In the General panel, select the <b>Show Uses arrowheads</b> checkbox.	
3	Click on the <b>Close</b> button. When you save the Use Case diagram, the Use connectors change to display arrowheads.	

### 5.7.2.19 Tree Style Hierarchy

In Enterprise Architect you can create a tree style inheritance diagram using a special form of the *Generalization* connector, as shown below.



**How to**

To create a tree style connector

Step	Action	See also
1	Create a normal Generalization between two elements.	
2	Right-click on the connector to open the context menu.	
3	Select the <b>Line Style   Tree Style - Vertical</b> or the <b>Line Style   Tree Style - Horizontal</b> menu option.	
4	Enterprise Architect automatically makes the Generalization layout conform to a specific shape.  By adding more Generalization connectors, and checking their Tree Style options, you can achieve the appearance of the diagram above.  You can slide the root and child Classes left and right to achieve the required result; Enterprise Architect maintains the conformity of the branch connectors.	

To set this style of connector as default

Step	Action	See also
1	Select the <b>Tools   Options   Links</b> menu option.  The Links page of the Options dialog displays.	
2	Select the <b>Generalization link style Default = Tree</b> checkbox to make this branching style the default style for inheritance connectors.	

**Notes**

- In the example diagram, the *Son ->Parent* connector has not yet been put in **Tree Style - Vertical** style
- This style of diagram provides a clearer layout for inheritance hierarchies and is easy to work with

**5.7.3 Connector Properties**

To access the connector Properties dialog, double-click on a connector in a diagram. You can change several characteristics of connectors from this dialog. Many of these characteristics generate text labels on or around the connector. You can change these labels using the **Label** context menu.

The connector Properties dialog has the following pages:




- General (see below)
- Constraints
- Binding
- Source Role
- Target Role
- Tagged Values
- Advanced properties.

**Access**    **Double-click on a connector in the diagram, or**  
**Right-click a connector in a diagram and, from the context menu, select Properties**

The General page enables you to configure the name of the connector, the direction, the line style, the stereotype (optional) and a comment.

#### General Page Options


Field/Option/ Button	Action	See also
<b>Source</b>	Indicates the name of the source element for the connector.	
<b>Target</b>	Indicates the name of the target element for the connector.	
<b>Name</b>	(Optional) Indicates a name for the connector; if entered, the name displays on the diagram.	
<b>Alias</b>	(Optional) Indicates an alternative name or alias for the connector.	
<b>Direction</b>	Select the appropriate direction details: from source to destination, destination to source, or bi-directional.  Some connectors have arrow heads that depend on this setting; some connectors are logically dependent on this (such as Inheritance).	
<b>Style</b>	Select the appropriate connection style; choose from: <ul style="list-style-type: none"> <li>• Direct</li> <li>• Auto-Routing</li> <li>• Bezier</li> <li>• Custom</li> <li>• Tree (Vertical, Horizontal, Lateral Vertical or Lateral Horizontal), or</li> <li>• Orthogonal (Square Corners or Rounded Corners)</li> </ul>	
<b>Stereotype</b>	(Optional) Indicate the name of a stereotype for the connector, or	<a href="#">Stereotype Selector</a>

Field/Option/ Button	Action	See also
	<p>click on the drop-down arrow and select one.</p> <p>Alternatively, click on the ( ... ) button and select the stereotype from the Stereotype Selector dialog.</p> <p>If entered, the stereotype is displayed on the diagram and overrides the connector type in the report documentation.</p>	
<b>Virtual Inheritance</b>	Indicates that inheritance is virtual; available only for <i>Generalization</i> connectors.	
<b>Scope</b>	Select the appropriate value for the scope (used for inheritance); available only for <i>Generalization</i> connectors where the child Class is C++.	
<b>Notes</b>	<p>(Optional) Specifies any notes on the connector; the notes are displayed in documentation, if required.</p> <p>As for the Notes window, you can format the text using the Notes toolbar at the top of the field.</p>	<a href="#">Notes Window</a>  <a href="#">Notes Toolbar</a> 

### Notes

- Generalize, Realize, Associate and Template Binding connectors also have a Binding page, which enables you to define binding expressions and parameter substitutions for the connector

### Learn more

- [Message Scope](#) 
- [Constraints](#) 
- [Binding](#) 
- [Source Role](#) 
- [Target Role](#) 
- [Connector Advanced](#) 
- [Tagged Values](#) 
- [Template Binding](#) 
- [Manage Object Labels](#) 

### 5.7.3.1 Connector Constraints

A UML connector can also have associated constraints placed on it. Constraints tell us something about the rules and conditions under which a relation operates. For example, it might be a pre-condition that a customer is of a certain type before an Association connector to an Account is allowed.

### How to



To set constraints on a connector

Step	Action	See also
1	Double-click on a connector to open the Connector Properties dialog.	
2	Select the Constraints tab.	
3	Fill in details of the constraint(s) that apply and click on the <b>Save</b> button.	

### Reference

Field	Usage	See also
<b>Constraint</b>	The name of the constraint.	
<b>Type</b>	Specifies the type of constraint (such as pre-condition).	
<b>Notes</b>	Indicates notes on the connector.	
<b>Defined Constraints</b>	Displays the list of constraints for this connector.	

### Notes

- Constraints about an Association (connector) can be added to further refine the model; constraints detail the business and operational rules for the model

### 5.7.3.2 Binding

Generalize, Realize, Associate and Template Binding connectors have a Binding tab on their Properties dialog. This enables you to define binding expressions and parameter substitutions for the connector.

### Learn more

- [Template Binding](#)<sup>[1444]</sup>
- [Parameter Substitution](#)<sup>[1445]</sup>
- [Parameterized Classes \(Templates\)](#)<sup>[1365]</sup>
- [Class Details](#)<sup>[959]</sup>

### 5.7.3.3 Source Role




A connector can have certain properties assigned to one end, and be associated with the particular role that element plays in the relationship. You can enter details about this role to further develop your model.

This description refers to the role of the *Source* element in a relationship, but applies equally to the role of the *Target* element.


#### Set the source role details

Step	Action	See also
1	Double-click on a connector to open the Connector Properties dialog.	
2	Select the Source Role page.	
3	Enter the required details and click on the <b>OK</b> button.	

#### Field Details

Field	Action	See also
<b>&lt;Type&gt; Role</b>	Enter the name of the role to be played.	
<b>Alias</b>	Enter an alias for the role, if required.	
<b>Role Notes</b>	Enter notes about the role.	
<b>Derived</b>	Indicate that the role value or values can be computed from other information.	
<b>Owned</b>	Indicate that the role is owned by the opposite Class rather than the Association.  Selecting this checkbox adds a 'dot' to the appropriate end of the connector (  ).	
<b>Derived Union</b>	Indicate that the role is derived from the properties that subset it.	
<b>Multiplicity</b>	Specify the role multiplicity (you can define the values of this field on the Cardinality tab of the UML Types dialog).  This is the range of instances of the role that can be active in the relationship - for example, <i>one</i> employee can be assigned to tasks; for the target role you define the range of instances (such as tasks) the	<a href="#">Cardinality</a>  <a href="#">Attribute properties</a> 

Field	Action	See also
	<p>employee could be assigned to.</p> <p>The values have the following formats:</p> <ul style="list-style-type: none"> <li>• *, or <b>0..*</b> - zero, one or many instances</li> <li>• <b>0..n</b> - zero or up to n instances, but no more than n</li> <li>• <b>n</b> - exactly n instances</li> <li>• <b>n..*</b> - n, or more than n instances</li> </ul> <p>You can also define source and target element multiplicity in the element Attribute properties.</p>	
<b>Ordered</b>	Indicate that the role is a list and the list is ordered.	
<b>Allow Duplicates</b>	<p>Indicate that the role can contain duplicate elements (relevant only if multiplicity is &gt; 1).</p> <p>This field maps to the UML property <i>isUnique</i> (selecting the checkbox maps to the <i>isUnique</i> value of <i>FALSE</i>).</p>	
<b>Containment</b>	Indicate the nature of the containment at the Destination (reference, value...).	
<b>Access</b>	Indicate the access level for the role.	
<b>Aggregation</b>	Indicate the type of aggregation that this role uses.	
<b>Target Scope</b>	Indicate the level at which this role applies (instance or classifier).	
<b>Navigability</b>	Indicate whether or not this role is navigable (non-navigable ends are shown depending on diagram properties).	
<b>Changeable</b>	Indicate whether this role is subject to change.	
<b>Constraint(s)</b>	Indicate a constraint on the role.	
<b>Qualifier(s)</b>	<p>Indicate any qualifiers or restrictions on the role; separate multiple qualifiers with a semi-colon.</p> <p>Alternatively, click on the ( ... ) button at the end of the field, and define a new qualifier on the Qualifiers dialog (qualifiers typed into the <b>Qualifier (s)</b> field are also automatically added to this dialog).</p>	<a href="#">Qualifiers Dialog</a> <sup>[1396]</sup>
<b>Stereotype</b>	(Optional) Indicate the name of a stereotype that applies to this end of	<a href="#">Stereotype</a>

Field	Action	See also
	the Association, or click on the ( ... ) button at the end of the field and select a stereotype from the Stereotype Selector dialog.	<a href="#">Selector</a> 
<b>Member Type</b>	Indicate a role type that can be used when generating collection Classes for multiplicity > 1.	

#### Notes


- Source role details are displayed at the start end of a connector; if you have drawn the connector the wrong way, you can always use the **Reverse Direction** menu option from the connector context menu

### 5.7.3.4 Target Role

A connector can have certain properties assigned to one end, and be associated with the particular role that element can play in the relationship.

You can enter details about this role to further develop your model.

#### Set the destination role details

Step	Action	See also
<b>1</b>	Double-click on a connector to open the Connector Properties dialog.	
<b>2</b>	Select the Target Role tab.	
<b>3</b>	The details and appearance of this tab are identical to the Source Role tab.	<a href="#">Source Role</a> 

#### Notes

- Destination role details are displayed at the terminating end of a connector on the diagram

### 5.7.3.5 Connector Tagged Values

The Tagged Values tab of the connector Properties dialog simply provides the Tagged Values window within the frame of the Properties dialog.

You can define Tagged Values for the connector and, on Association and Aggregation connector types, you can set additional Tagged Values for the source and/or target role.

#### How to

To set Tagged Values for the connector

Step	Action	See also
1	On the Properties dialog for the connector, click on the Tagged Values tab.	
2	Select the connector type, <b>Connector Source</b> or <b>Connector Target</b> as required.	
3	Either click on the <b>New Tags</b> button or press ( <b>Ctrl+N</b> ). The Tagged Value dialog displays.	
4	In the <b>Tag</b> field type the tag name and value, or click on the drop-down arrow and select a predefined Tagged Value type.	
5	Click on the <b>OK</b> button to save the changes.	

#### Learn more

- [Tagged Values](#)  f1134

### 5.7.3.6 Connector Advanced

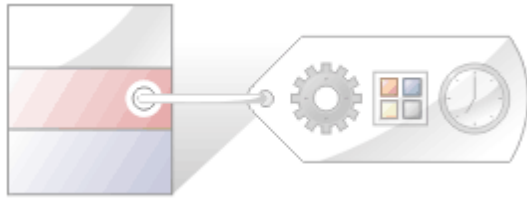
The Advanced page displays the advanced properties of the connector (where they exist) and enables you to reset the values of these properties.

#### How to

To set a value for an advanced (or custom) property


Step	Action	See also
1	Click on the data field to the right of the property name.	
2	Depending on the property, either: <ul style="list-style-type: none"> <li>• Type the value in free text</li> <li>• Click on the drop-down arrow and select the value from the list</li> <li>• Click on the ( ... ) Browse button and search for the required value</li> </ul>	
3	Click on the <b>OK</b> button.	

## 5.8 Tagged Values



### Topics

Topic	Detail	See also
<b>What is a Tagged Value</b>	<p>Tagged Values are a convenient way of adding additional information to an element, beyond what is directly supported by UML. The UML specification provides the Tagged Value element for just this purpose.</p> <p>Often Tagged Values are used during code generation or by other tools to pass information or operate on elements in particular ways. Tagged Values are the preferred method of extending the code generation capabilities of the modeling tool per element / per language.</p> <p>A Tagged Value, strictly, is the value of a property of a modeling item, the property being called a tag. For example, a Class element called <i>Person</i> might have a tag called <i>Age</i> with the Tagged Value of <b>42</b>. The combination of tag and value, however, is often referred to as a Tagged Value.</p> <p>Enterprise Architect defines a Tagged Value Type, which constrains the possible values of a tag and can specify how a value is assigned to the tag. For example, the tag <i>Age</i> might have a Tagged Value Type of <i>Integer</i>, so the user simply types in a numeric value. Alternatively, the type could be <i>Spin</i>, with lower and upper limits of, say, <b>20</b> and <b>120</b>, so the user sets a value by clicking on arrows in the field to increment or decrement the value within the limits of 20 and 120.</p> <p>To quickly add Tagged Values to one of more elements, see the <i>Quick Start - Add Tagged Value to Elements</i> topic.</p>	<p><a href="#">Tagged Value Type</a> <sup>[1150]</sup></p> <p><a href="#">Quick Start - Add Tagged Value to Elements</a> <sup>[1136]</sup></p>
<b>The Tagged Values Window</b>	<p>The Tagged Values window is used to view and modify Tagged Values for the currently selected modeling item, either in the current diagram or in the Project Browser.</p> <p>The Tagged Values window is a dockable window. You can use it to perform the following actions:</p> <ul style="list-style-type: none"> <li>• Assign a Tagged Value to an Item</li> <li>• Modify Tagged Values</li> <li>• Assign Notes to a Tagged Value</li> <li>• Perform advanced tag management</li> </ul> <p>A Technology Developer can also create new structured Tagged Values, reference data Tagged Values and custom Tagged Values from predefined Tagged Value Types.</p>	<p><a href="#">Assign a Tagged Value to an Item</a> <sup>[1137]</sup></p> <p><a href="#">Modify Tagged Values</a> <sup>[1138]</sup></p> <p><a href="#">Assign Notes to a Tagged Value</a> <sup>[1138]</sup></p> <p><a href="#">Perform Advanced Tag Management</a> <sup>[1140]</sup></p> <p><a href="#">Custom Tagged Values</a> <sup>[1626]</sup></p> <p><a href="#">Tagged Value Types</a> <sup>[1627]</sup></p>

Topic	Detail	See also														
<b>Model Elements and Features with Tagged Values</b>	The following model components can use the Tagged Values window as a convenient way to quickly view and modify Tagged Values:															
	<table><tr><th>Component</th><th>Description</th></tr><tr><td><b>Elements</b></td><td>Elements display their own Tagged Values and inherited values.</td></tr><tr><td><b>Object Instances</b></td><td>Object Instances display owned tags and those inherited from their classifier.</td></tr><tr><td><b>Ports and Parts</b></td><td>Ports and parts display information similar to Port/Part 'Type' instead of a classifier.  Tags are included for all parents and other super types.</td></tr><tr><td><b>Attributes</b></td><td>Include owned Tagged Values and those inherited from type classifiers, with the inclusion of any inherited values.</td></tr><tr><td><b>Operations</b></td><td>Owned properties only.</td></tr><tr><td><b>Connectors</b></td><td>Owned properties only.</td></tr></table>	Component	Description	<b>Elements</b>	Elements display their own Tagged Values and inherited values.	<b>Object Instances</b>	Object Instances display owned tags and those inherited from their classifier.	<b>Ports and Parts</b>	Ports and parts display information similar to Port/Part 'Type' instead of a classifier.  Tags are included for all parents and other super types.	<b>Attributes</b>	Include owned Tagged Values and those inherited from type classifiers, with the inclusion of any inherited values.	<b>Operations</b>	Owned properties only.	<b>Connectors</b>	Owned properties only.	
	Component	Description														
	<b>Elements</b>	Elements display their own Tagged Values and inherited values.														
	<b>Object Instances</b>	Object Instances display owned tags and those inherited from their classifier.														
	<b>Ports and Parts</b>	Ports and parts display information similar to Port/Part 'Type' instead of a classifier.  Tags are included for all parents and other super types.														
	<b>Attributes</b>	Include owned Tagged Values and those inherited from type classifiers, with the inclusion of any inherited values.														
	<b>Operations</b>	Owned properties only.														
<b>Connectors</b>	Owned properties only.															
When over-riding an inherited property, Enterprise Architect copies the tag from the parent down to the child element and sets the new value, leaving the original tag unchanged.																
<b>Tagged Values Toolbar Buttons</b>	The buttons in the Tagged Values toolbar enable you to add, edit, sort, delete and arrange the Tagged Values of model features															
	<div></div> <p>From left to right, the button functions are as follows:</p> <ul style="list-style-type: none"><li>The <b>Show in compartments</b> button displays the Tagged Values in element compartments on diagrams</li><li>The <b>Sort and Show Alphabetically</b> button sorts the current Tagged Values for the element alphabetically</li><li>The <b>New Tagged Value</b> button adds a new tag, to which you assign a value</li><li>The <b>Edit Tagged Value Notes</b> button enables you to create notes that explain the purpose of the Tagged Value</li></ul>															

Topic	Detail	See also
	<ul style="list-style-type: none"> <li>The <b>Delete Tagged Value</b> button removes the currently selected Tagged Value</li> <li>The <b>Default Tagged Value Types</b> button enables quick access to tag definitions created in the Configuration menu</li> <li>The <b>Tagged Value Options</b> button enables you to show or hide the fully qualified paths for the Tagged Values in the window, and to show duplicate Tagged Values</li> <li>The <b>Help</b> button displays help relating to use of the Tagged Values window</li> </ul>	

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect** | **Getting Started** | **Basic Tasks** | **Add A Tagged Value**

### 5.8.1 Quick Start - Add Tagged Value To Elements

You can add a Tagged Value to one or more elements with a special shortcut. You can also use the Current Element toolbar, on which the last button is a shortcut to the **Add Tagged Value** function.

To delete this property you must open the element Properties dialog, go to the Tagged Values tab and manually delete the item. There is currently no shortcut to delete tags from multiple elements simultaneously.

To add notes to the Tagged Value, go to the Tagged Values tab, click on the Tagged Value name, and click on the **Edit Notes** button in the tab toolbar. The Notes dialog displays.

Any Notes text you enter also displays in the **Info** section at the bottom of the Tagged Values window.

#### Access    **Add | Tagged Value**

#### Add a Tagged Value

Step	Action	See also
1	<p>From an element context menu (or the context menu of a multi-selection) choose the <b>Advanced   Tagged Value</b> menu option.</p> <p>Alternatively, in a diagram, select one or more elements and press <b>Ctrl+Shift+T</b>. The Tagged Values dialog displays, in which you enter a Name and Value for the tag.</p>	<a href="#">Tagged Values</a> <sup>[1134]</sup>
2	Click on the <b>OK</b> button to add your new Tagged Value to all the currently selected elements.	



Learn more


- [The Element Browser](#)<sup>[989]</sup>

**5.8.2 Assign a Tagged Value to an Item**

You can assign Tagged Values to a range of model objects, including elements, object instances, connectors, attributes and operations, using the Tagged Values window. You would probably have the window open already as a docked window, and any object you select becomes the focus of the window; you can then review and add or delete Tagged Values for that object.

**Access**    **View | Tagged Values** (Ctrl+Shift+6 )

Assign a Tagged Value to an item

Step	Action	See also
1	Select the model feature to which to assign a Tagged Value. This feature becomes the focus of the window.	
2	In the Tagged Value window, either click on the <b>New Tags</b> button  or press ( <b>Ctrl+N</b> ). The Tagged Value dialog displays.	
3	In the <b>Tag</b> field type the tag name, or click on the drop-down arrow and select the appropriate tag to assign to the item.	
4	If appropriate, type a specific value for the tag in the <b>Value</b> field.	
5	To confirm selection of the Tagged Value, click on the <b>OK</b> button.	

Notes

- You can also assign a Tagged Value directly to an element by clicking on the element and either selecting the **Element | Advanced | Add Tagged Value** menu option or pressing ( **Ctrl+Shift+T** ), and completing the Tagged Value dialog as described above

Learn more

- [Predefined Tagged Value Types](#)<sup>[1621]</sup>
- [Assign Notes to a Tagged Value](#)<sup>[1138]</sup>
- [Modify Tagged Values](#)<sup>[1138]</sup>
- [Model Elements and Features with Tagged Values](#)<sup>[1135]</sup>

### 5.8.3 Assign Notes to a Tagged Value

Once a Tagged Value has been assigned to a model feature, you can add information and **notes** describing the Tagged Value to the information property of the Tagged Value.

**Access** **View | Tagged Values (Ctrl+Shift+6)**

#### Assign a note to a Tagged Value

Step	Action	See also
1	Select the <b>Tagged Values</b> menu option. The Tagged Values window displays.	
2	Click on the model feature for which to edit the Tagged Values. Its Tagged Values display in the Tagged Values window.	
3	Click on the Tagged Value to add information to.	
4	Click on the <b>Edit Tagged Value Notes</b> button or press <b>Ctrl+E</b> . The Tagged Value Note dialog displays.	
5	In the <b>Note</b> field, type the information relating to the Tagged Value. Click on the <b>OK</b> button. The information is displayed in the lower portion of the Tagged Values window whenever the Tagged Value is selected.	

#### Learn more

- [Model Features](#) 

### 5.8.4 Modify Tagged Values

Once a Tagged Value has been assigned to a model feature (such as an element or connector), it is possible to edit the values in the Tagged Values window.

**Access** **View | Tagged Values ( Ctrl+Shift+6 )**

#### Edit Tagged Values

Step	Action	See also
1	<p>Click on the model feature for which to edit the Tagged Values.</p> <p>The Tagged Values window shows all of the tags for the selected feature, each with their current value.</p>	
2	<p>Edit the fields as appropriate; the information entered can only reflect the value types that have been defined by the tag's Tagged Value Type.</p> <p>There are six types of value field for a Tagged Value:</p> <ul style="list-style-type: none"> <li>• 'Open' fields, in which you can type any appropriate value</li> <li>• Structured Tags, where you do not directly edit the composite value but expand the tag to show the component tags and edit the component values in 'open' fields; the composite value updates as you click off the component field</li> <li>• 'Drop-down list' fields, where you click on the drop-down arrow to select from a discrete list of possible values such as <b>M</b> or <b>F</b>, or <b>Win</b>, <b>Lose</b> or <b>Draw</b></li> <li>• 'Spin' fields, where you click on up or down arrows in the field to increase or decrease the value within certain limits</li> <li>• Checklist fields, where you click on an ellipsis ( ... ) in the field to display the Check List dialog, which lists a set of items that can be selected by ticking the checkbox against each item; if any checkboxes are <b>not</b> selected the value of the tag is <b>Incomplete</b>, and when <b>all</b> checkboxes are selected the tag value is <b>Complete</b></li> <li>• 'Further detail' fields, where you click on an ellipsis ( ... ) in the field to display a dialog in which you enter information (such as notes) or indicate a source of further information (such as a classifier)</li> </ul>	<p><a href="#">Predefined Structured Types</a> <sup>[1622]</sup></p> <p><a href="#">Select &lt;Item&gt; Dialog</a> <sup>[994]</sup></p>

### Notes

- To override a Tagged Value defined in a parent element, edit the value in the *from <parentname>* compartment of the Tagged Values window

Once this has been done the tag is moved into the selected element's Tagged Values; this does not affect the Tagged Values defined in the parent element

### Learn more

- [Tagged Values](#) <sup>[1134]</sup>
- [Assign a Tagged Value to an Item](#) <sup>[1137]</sup>

## 5.8.5 Show Duplicate Tags

Tagged Values are by default set to hide duplicate values. This setting is used to facilitate inherited and overridden tag names.

### How to

To set the Tagged Values window to show duplicate values

Step	Action	See also
1	On the Tagged Values window toolbar, click on the <b>Options</b> icon.	
2	Select the <b>Show Duplicate Tags</b> context menu option.	

**Alternatively:**

Step	Action	See also
1	Select the <b>Tools   Options</b> menu option. The Options dialog displays.	
2	From the hierarchical tree, select the <b>Objects</b> item.	
3	Select the <b>Show Duplicate Tags</b> checkbox.	

**Notes**

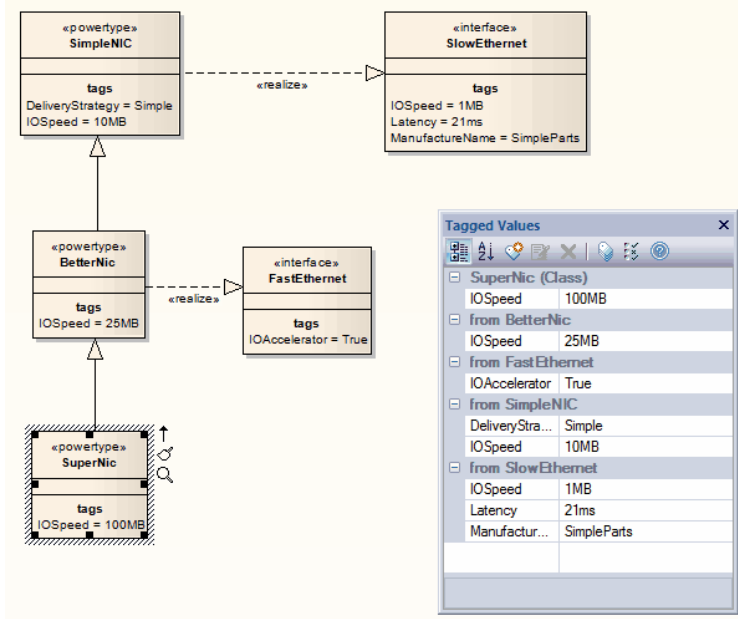
- In either procedure, to hide duplicate values again deselect the option or check box

**5.8.6 Advanced Tag Management**

Tagged Values can also be managed within a type hierarchy and with respect to element instances, using the Tagged Values window.

**Topics**

Topic	Detail	See also
<b>Usage</b>	Using the Tagged Values window it is possible to: <ul style="list-style-type: none"> <li>• View Tagged Values inherited from parent Classes or realized interfaces or applied stereotypes</li> <li>• Override Tagged Values derived from parents or applied stereotypes with a unique value for the current element</li> <li>• Delete Tagged Values from the current element (if a parent version of the Tagged Value exists, it re-appears in the list after the override is deleted)</li> </ul>	
<b>Tag Hierarchy</b>	The diagram below illustrates a complex tag hierarchy and the way Tagged Values can be either inherited or overridden in specialized Classes to create the final tagged property set for an element.  Note also that a similar concept applies to instances, in which case the full tag set is created from the directly owned tags, plus all of	

Topic	Detail	See also																											
	<p>those merged in from the classifier's type hierarchy, additional stereotypes and realized interfaces.</p>  <p>The diagram illustrates a class hierarchy for network interfaces. At the base is <b>SuperNic</b> (power type) with a tagged value <b>IOSpeed = 100MB</b>. It inherits from <b>BetterNic</b> (power type), which has <b>IOSpeed = 25MB</b> and realizes the <b>FastEthernet</b> interface (tagged value <b>IOAccelerator = True</b>). <b>BetterNic</b> inherits from <b>SimpleNIC</b> (power type), which has <b>DeliveryStrategy = Simple</b> and <b>IOSpeed = 10MB</b>, and realizes the <b>SlowEthernet</b> interface (tagged values <b>IOSpeed = 1MB</b>, <b>Latency = 21ms</b>, <b>ManufactureName = SimpleParts</b>). A 'Tagged Values' window on the right shows the inheritance of these values.</p> <table border="1" data-bbox="903 645 1206 1048"> <caption>Tagged Values</caption> <thead> <tr> <th>Class/Interface</th> <th>Property</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>SuperNic (Class)</td> <td>IOSpeed</td> <td>100MB</td> </tr> <tr> <td>from BetterNic</td> <td>IOSpeed</td> <td>25MB</td> </tr> <tr> <td>from FastEthernet</td> <td>IOAccelerator</td> <td>True</td> </tr> <tr> <td>from SimpleNIC</td> <td>DeliveryStra...</td> <td>Simple</td> </tr> <tr> <td>from SimpleNIC</td> <td>IOSpeed</td> <td>10MB</td> </tr> <tr> <td>from SlowEthernet</td> <td>IOSpeed</td> <td>1MB</td> </tr> <tr> <td>from SlowEthernet</td> <td>Latency</td> <td>21ms</td> </tr> <tr> <td>from SlowEthernet</td> <td>Manufactur...</td> <td>SimpleParts</td> </tr> </tbody> </table>	Class/Interface	Property	Value	SuperNic (Class)	IOSpeed	100MB	from BetterNic	IOSpeed	25MB	from FastEthernet	IOAccelerator	True	from SimpleNIC	DeliveryStra...	Simple	from SimpleNIC	IOSpeed	10MB	from SlowEthernet	IOSpeed	1MB	from SlowEthernet	Latency	21ms	from SlowEthernet	Manufactur...	SimpleParts	
Class/Interface	Property	Value																											
SuperNic (Class)	IOSpeed	100MB																											
from BetterNic	IOSpeed	25MB																											
from FastEthernet	IOAccelerator	True																											
from SimpleNIC	DeliveryStra...	Simple																											
from SimpleNIC	IOSpeed	10MB																											
from SlowEthernet	IOSpeed	1MB																											
from SlowEthernet	Latency	21ms																											
from SlowEthernet	Manufactur...	SimpleParts																											

### Learn more

- [Tagged Values](#) <sup>1134</sup>

## 5.9 Notes

Notes are the main documentation feature you use to describe an element, diagram, feature or connector; in the documentation that Enterprise Architect generates, notes feature prominently. You can record and view notes on a modeling object using the Notes window.

**Access** **View | Notes** (Ctrl+Shift+1)

### Topics

Topic	Detail	See also
<b>Usage</b>	<p>You use the Notes window to view and edit the documentation (notes) associated with elements, diagrams, attributes, operations and connectors, either from a diagram or from the Project Browser.</p> <p>When you select an element, the note displayed changes to reflect the current selection; if you make changes to notes in this window, they are saved.</p> <p>If you want to display the Notes information in a more readable layout, you can resize the window.</p> <p>You can also format the notes text using the Notes toolbar at the top of the Notes window.</p> <p>You can cut, copy, paste and delete text in the Notes window, or in any <b>Notes</b> or <b>Description</b> field that shows the Notes toolbar, using a right-click context menu; from the context menu, you can also select an option to spell-check a highlighted word.</p> <p>You can also edit notes by double-clicking on an element or connector in a diagram or in the Project Browser, to open the Properties dialog and edit the <b>Notes</b> field; any formatting changes made in one display are reflected in the other.</p> <p>On the Testing, Maintenance and Project Management windows, any descriptive, history, input or results text for a selected item is also displayed in the Notes window; you cannot edit this text in the Notes window.</p>	<p><a href="#">Notes Toolbar</a> <sup>1143</sup></p>
<b>Glossary Entries</b>	<p>The Notes window or field also enables you to create a Project Glossary entry from text you have highlighted in the window or field.</p> <p>To create the Glossary entry</p> <ul style="list-style-type: none"> <li>• Highlight the notes text to use as the Glossary definition (if suitable text is available), and press ( <b>Ctrl+C</b> ) to copy it</li> <li>• Highlight the text to use as the Glossary term, and right-click on it to display the context menu</li> <li>• Select the <b>Create   Glossary Definition</b> menu option; the Glossary Detail dialog displays, with the selected term in the <b>Term</b> field</li> <li>• If you have copied some definition text, paste it into the <b>Meaning</b> field; otherwise type a suitable definition of the term in this field</li> </ul>	<p><a href="#">Project Glossary</a> <sup>533</sup></p> <p><a href="#">Glossary Detail</a> <sup>534</sup></p>


Topic	Detail	See also
	<ul style="list-style-type: none"> <li>In the <b>Type</b> field, select the appropriate term type</li> <li>Click on the <b>Apply</b> button to save the new Glossary definition</li> </ul> <p>The term displays in the Notes text as a roll-over hyperlink which, when you move the cursor over it, displays the Glossary definition of the term.</p> <p>Having created a glossary definition anywhere else in the model, you can insert the glossary term in the text of the Notes window (or Notes panel of a dialog) as a rollover hyperlink to the definition.</p>	
<b>Inserting a Glossary term in a Notes Window</b>	<ul style="list-style-type: none"> <li>In the <b>Notes</b> window, move the cursor to the point in the text at which to insert the glossary term</li> <li>Press ( <b>Ctrl+Space</b> ); the glossary term selection list displays</li> <li>Double-click on the term to insert in the Notes text; the term is inserted as a rollover hyperlink to the definition</li> </ul>	
<b>Additional keyboard shortcuts</b>	<ul style="list-style-type: none"> <li>Insert Date and Time ( <b>F5</b> )</li> <li>Select line of text ( <b>F8</b> )</li> <li>Spell check Notes text only ( <b>F7</b> )</li> <li>Undo changes ( <b>Ctrl+Z</b> )</li> <li>Redo changes ( <b>Ctrl+Y</b> ) or ( <b>Ctrl+Shift+Z</b> )</li> <li>Copy ( <b>Ctrl+C</b> )</li> <li>Paste ( <b>Ctrl+V</b> )</li> <li>Cut ( <b>Ctrl+X</b> )</li> </ul> <p>Any Note text appearing in the element Note compartments in diagrams is not formatted.</p>	

### 5.9.1 Notes Toolbar

Although it is not an independent toolbar that you can pin to the screen top or sides, or float in your work area, the Notes toolbar appears in many places across Enterprise Architect in the **Notes**, **Description** and **History** fields.

#### Topics

Topic	Detail	See also
<b>Where the Toolbar Appears</b>	<p>Throughout Enterprise Architect on various tabs and pages of, for example, the:</p> <ul style="list-style-type: none"> <li>Element and Connector Properties dialogs</li> </ul>	

	<ul style="list-style-type: none"> <li>• Operations and Attributes Properties dialogs</li> <li>• Diagram Properties dialog</li> <li>• Hyperlink Details dialog</li> <li>• Testing window descriptions</li> <li>• Notes window</li> <li>• Glossary Detail dialog</li> <li>• Assigned Resources dialog</li> <li>• Project Management window</li> <li>• Scenarios &amp; Requirements window</li> </ul>	
Options	 <p>The options of this toolbar operate on selected text and any new text continuing from the formatting.</p> <p>The options (with some keyboard shortcuts) are, from left to right:</p> <ul style="list-style-type: none"> <li>• Make text bold ( <b>Ctrl+B</b> )</li> <li>• Make text italic ( <b>Ctrl+I</b> )</li> <li>• Underline text ( <b>Ctrl+U</b> )</li> <li>• Change the font color of the text</li> <li>• Insert list bullet points ( <b>Ctrl+.</b> ) (full stop)</li> <li>• Insert list numbering ( <b>Ctrl+1</b> )</li> <li>• Make text superscript</li> <li>• Make text subscript</li> <li>• Insert a hyperlink - this displays the Hyperlink Details dialog, on which you specify the type of hyperlink and type in or browse for the location of the target of the hyperlink; you can use full paths or local (path substitution) paths, and you can either select or type the link text or leave the text blank to use the link address as the text</li> <li>• Create a new linked document for the element, or edit an existing one</li> </ul>	<p><a href="#">Hyperlinks</a> [2002]</p> <p><a href="#">Create Linked Document on an Element</a> [1047]</p>

### Notes

- If the toolbar is displayed but grayed out, the text field is read-only and cannot be edited; other long text fields in Enterprise Architect might not have the toolbar, in which case the Notes facility is not available for those fields
- For any Notes text that is displayed on a diagram, you must select the **Render Formatted Notes** checkbox on the Feature Visibility dialog in order to reproduce the formatting
- You can create a Project Glossary term and definition from text in any field that has the Notes toolbar

### Learn more

- [Feature Visibility dialog](#) [845]



- [Project Glossary](#)<sup>[533]</sup>
- [Notes](#)<sup>[1142]</sup>

## 5.10 Reference Data



Reference data is used in many places to provide content for drop-down list boxes. Setting up a project often involves setting up the base set of reference types to use. Reference data options can be set up from the **Settings** menu, including:

- People
- General Types
- Maintenance
- Metrics and Estimation
- UML
- Data Types

Having set up the reference data in a project, you can also export and import it between projects.

### Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have Manage Reference Data - Update permission to update and delete reference items.

### Learn more

- [People](#) <sup>[1152]</sup>
- [General Types](#) <sup>[1158]</sup>
- [Maintenance](#) <sup>[1169]</sup>
- [Metrics and Estimation](#) <sup>[1169]</sup>
- [UML](#) <sup>[1146]</sup>
- [Data Types](#) <sup>[1171]</sup>
- [Export and Import](#) <sup>[374]</sup>
- [Manage Reference Data - Update](#) <sup>[329]</sup>
- [Applying Stereotypes](#) <sup>[1453]</sup>
- [Custom Stereotypes](#) <sup>[1461]</sup>

### 5.10.1 UML Types

The UML Types dialog enables you to configure stereotypes, Tagged Value types and the cardinality list for your project.

Access   **Settings | UML Types**

Use to

- Add, modify and delete stereotypes
- Customize stereotypes appearance
- Updating shape scripts and metafiles associated with Stereotypes
- Create/modify tagged values
- Define cardinality rules

#### Learn more

- [Stereotypes](#) <sup>[1147]</sup>
- [Tagged Value Types](#) <sup>[1150]</sup>
- [Cardinality](#) <sup>[1151]</sup>

### 5.10.1.1 Stereotype Settings

Enterprise Architect has an extensive set of Standard Element Stereotypes that you can apply to any UML construct. Using the Stereotypes tab of the UML Types dialog, a Technical Developer can also customize the stereotypes for your project by adding, modifying and deleting them. For information on customizing stereotypes, see *Custom Stereotypes*.

Stereotypes can be modified to make use of metafiles (image files) or customized colors, or to make use of the Enterprise Architect Shape Scripts to make new element shapes to determine the shape and dimensions of the element.

**Access**    **Settings | UML Types > Stereotypes**

#### Use to

- Add, modify and delete stereotypes
- Customize stereotypes' appearance
- Update shape scripts and metafiles associated with stereotypes

#### Reference

Field	Usage	See also
<b>Stereotype</b>	Specify the name of the stereotype.	
<b>Group name</b>	Enable grouping of stereotype features by a plural name, for attributes and operations, which is shown on diagrams in the attribute and operations compartments.	
<b>Base Class</b>	Enable the stereotyped element to inherit the base characteristics from a pre-existing element type.	
<b>Notes</b>	Indicates any notes concerning the stereotype (not the elements to which the stereotype is to be applied).	

Field	Usage	See also
<b>None</b>	Switch to the default element appearance.	
<b>Metafile</b>	Enable an image file to be used for the appearance of the stereotype.	
<b>Shape Script</b>	Specify custom shapes for the stereotype using the Enterprise Architect Shape Scripting language.	<a href="#">Shape Scripts</a> [1582]
<b>Assign</b>	Add the associated metafile or Shape Script from the stereotyped element.	
<b>Remove</b>	Remove the associated metafile or Shape Script from the stereotyped element.	
<b>Fill</b>	<p>Set the default background color of the element.</p> <p>This color will be applied to all occurrences of any element to which the stereotype has been applied; if the color is subsequently changed, the change is immediately applied to all occurrences of any element to which the stereotype was applied (as for changes to any other property of the stereotype).</p> <p>However, on elements created with the stereotype, the default color might be overridden by other color definitions of a higher priority that have been applied to the element, such as the Format Toolbar fill definition or the Alternative Appearance dialog (<b>F4</b>) definition.</p>	<a href="#">Format Toolbar</a> [785] <a href="#">Set an Element's Default Appearance</a> [927]
<b>Border</b>	Control the border color.	
<b>Font</b>	Control the color of the stereotype font.	
<b>Reset</b>	Reset the appearance of the element to the default element appearance.	
<b>Preview</b>	Provides a visual representation of the currently active Stereotype.	
<b>New</b>	Enables a new stereotype to be created.	
<b>Save</b>	Saves the Stereotype details and adds the Stereotype to the Defined Stereotypes list.	

Field	Usage	See also
<b>Delete</b>	Removes a Stereotype from the Defined Stereotypes list.	

#### Learn more

- [Standard Element Stereotypes](#) <sup>[1457]</sup>
- [Applying Stereotypes](#) <sup>[1453]</sup>
- [Custom Stereotypes](#) <sup>[1461]</sup>

#### 5.10.1.1.1 Shape Editor

The Shape Editor enables a Technology Developer to specify custom shapes via a scripting language; that is, to create *Shape Scripts*. These custom shapes are drawn instead of the standard UML notation. Each script is associated with a particular Stereotype, and is drawn for every element of that stereotype.

**Access**    **Settings | UML Types > Stereotypes : Assign**

#### Reference

Field	Usage	See also
<b>Format</b>	The format in which the script is written.	
<b>Import</b>	Opens a Windows browser dialog, allowing for a script file to be loaded.	
<b>Export</b>	Opens a Windows browser dialog, allowing for a script file to be saved.	
<b>Edit window</b>	The area in which a script can be written.	<a href="#">Writing Scripts</a> <sup>[1585]</sup>
<b>Preview of main</b>	A visual representation of the compiled script. Nothing displays until the <b>Refresh</b> button is clicked.	
<b>Next Shape</b>	If a composite shape is defined within the Edit Window, clicking on the <b>Next Shape</b> button pages through the components of the shape.	
<b>Refresh</b>	Parses your script and produces a visual example of the shape in the Preview of main window.	
<b>OK</b>	Exit from the Shape Editor; don't forget to save your scripts from the	

	Stereotype tab.	
--	-----------------	--

### Notes

- Shape Scripts adopt the same color gradient settings as normal elements, as defined in the Standard Colors page of the Options dialog
- If an element's appearance is modified by a Shape Script, many of the **Advanced** context menu options for that element are disabled
- Once you have finishing writing your Shape Script, click on the **OK** button; to save the Shape Script you must click on the Stereotypes tab

### Learn more

- [Getting Started With Shape Scripts](#) <sup>[1583]</sup>
- [Advanced Settings](#) <sup>[944]</sup>

## 5.10.1.2 Tagged Value Types

Tagged Values are used in a variety of places within Enterprise Architect to specify additional information about an element or connector. The Tagged Value Types tab of the UML Types dialog enables a Technology Developer to rapidly create Tagged Values, using a range of predefined structured Tagged Values to create structured tags that adhere to a specific format. For example, for model features that use the predefined tag Boolean you can use the Tagged Values window to assign a value of *True* or *False* and no other value.

You can also add default Tagged Value names and create predefined reference data Tagged Value types and custom masked Tagged Value types.

Any Tagged Value names created display in the drop-down lists of Tagged Value names in the Tagged Value dialogs for elements, operations and attributes.

**Access**    **Settings | UML Types > Tagged Value Types**

### Reference

Field	Usage	See also
<b>Tag Name</b>	Type the new name of the Tagged Value.	
<b>Description</b>	Type a description of the Tagged Value.	
<b>Detail</b>	Type any additional information necessary.	
<b>New</b>	Click to begin creating a new Tagged Value.	
<b>Save</b>	Click to save the details of a Tagged Value.	

Field	Usage	See also
<b>Delete</b>	Click to delete a Tagged Value from the <b>Defined Tag Types</b> list.	
<b>Defined Tag Types</b>	A list of previously defined Tagged Values.	

### Notes

- You can transport these Tagged Value Type definitions between models, using the Export Reference Data and Import Reference Data options; Tagged Value Types are exported as *Property Types*

### Learn more

- [Predefined Structured Tagged Values](#) <sup>[1622]</sup>
- [Create Structured Tags](#) <sup>[1622]</sup>
- [Predefined Reference Data Tagged Value Types](#) <sup>[1626]</sup>
- [Custom Masked Tagged Value Types](#) <sup>[1626]</sup>
- [Tagged Values Window](#) <sup>[1134]</sup>
- [Export Reference Data](#) <sup>[376]</sup>
- [Import Reference Data](#) <sup>[380]</sup>

### 5.10.1.3 Cardinality

The Cardinality Values tab of the UML Types dialog enables you to add, modify and delete values in the default cardinality list.

The cardinality values are used to define the multiplicity of source and target elements in relationships. This is the range of instances of the role that can be active in the relationship; for example, one employee can be assigned to tasks; for the target role you define the range of instances (such as tasks) the employee could be assigned to.

The cardinality values are also used to define the multiplicity of a Classifying element; that is, the number of instances of the element that can exist. For example, the Class element *Building Walls* might have a multiplicity of 2..n, meaning that at least two walls must exist (to support the roof) but there can be many walls if the building design required it.

The values have the following formats:

- \***, or **0..\*** - zero, one or many instances
- 0..n** - zero or up to n instances, but no more than n
- n** - exactly n instances
- n..\*** - n, or more than n instances.

**Access**    **Settings | UML Types > Cardinality Values**

### Reference

Field	Usage	See also
<b>Cardinality</b>	Type the new name of the Cardinality Value.	
<b>New</b>	Type a description of the Cardinality Value.	
<b>Save</b>	Saves the Cardinality value and adds it to the Cardinality List.	
<b>Delete</b>	Deletes a Cardinality value from the Cardinality List.	
<b>Cardinality List</b>	A list of already defined Cardinality values.	

#### Notes

- You can transport these cardinality values between models, using the **Export Reference Data** and **Import Reference Data** options

#### Learn more

- [Export Reference Data](#) <sup>[376]</sup>
- [Import Reference Data](#) <sup>[380]</sup>
- [Source Role](#) <sup>[1130]</sup>
- [Destination Role](#) <sup>[1132]</sup>

## 5.10.2 People

Access   **Settings | Project Types | People**

#### Use to

- Maintain Authors involved within a project
- Define role types that are captured within Enterprise Architect
- Record information on project resources
- Capture client details associated with the current model

#### Learn more

- [Project Authors](#) <sup>[1153]</sup>
- [Project Roles](#) <sup>[1155]</sup>
- [Project Resources](#) <sup>[1156]</sup>
- [Project Clients](#) <sup>[1157]</sup>



### 5.10.2.1 Project Authors



You can define the people who are working on a project, such as the authors of specific elements.

**Access**    **Settings | Project Types | People > Project Author(s)**

#### Use to

- Maintain Authors involved within a project

#### Reference

Field	Usage	See also
<b>Name(s)</b>	<p>Type the name of the person registered as a Project Author.</p> <p>If you are using a Windows Active Directory, you can select names from the directory; click on the ( ... ) (Browse) button to display the Select Users dialog.</p> <p>You can also type a list of names separated by semi-colons; this enables you to define a group of people sharing a role, such as a team of Developers, Testers or Analysts.</p> <p>Do not leave any spaces between the names and the semicolons.</p>	<a href="#">Select Users</a>  <sup>[1154]</sup>
<b>Role</b>	<p>(Mandatory) Enter the role the Project Author plays in the project (such as Designer, Analyst, or Architect).</p> <p>You can type a role name or click on the drop-down arrow and select a role defined through the Project Roles tab.</p>	<a href="#">Project Roles</a>  <sup>[1155]</sup>
<b>Notes</b>	Type any additional notes concerning the Project Author.	
<b>New</b>	Add further Authors.	
<b>Save</b>	Add the new Author to the Defined Authors list.	
<b>Delete</b>	Removes an Author from the Defined Authors list.	
<b>Defined Authors</b>	Review the Project Authors already defined.	

#### Notes

- If you enter multiple names, Enterprise Architect adds them separately and in alphabetical order to the

**Defined Authors** list; if you then click on one of these names, Enterprise Architect displays that name only in the **Name(s)** field

- If you type in a role, this is not added to the roles on the Project Roles tab
- You can transport these author definitions between models, using the Export Reference Data and Import Reference Data options

#### Learn more

- [Export Reference Data](#)<sup>[376]</sup>
- [Import Reference Data](#)<sup>[380]</sup>

#### 5.10.2.1.1 *Select Users*

If your company is using a Windows Active Directory, you can select the Project Author names from the local or corporate-wide directory.

**Access**    **Settings | Project Types | People > Project Author(s) : (...)**

#### Use to

- Select Project Authors from Windows Active Directory

#### Reference

Field	Usage	See also
<b>Object Types...</b>	Opens the Object Types dialog, which provides a choice of object types that can be used.	
<b>Locations...</b>	Defines the root location from which to begin a search.	
<b>Check Names</b>	Matches object names listed in the <b>Enter the object names to select</b> section.	
<b>Enter the object names to select</b>	Specify object names to search for.	
<b>Advanced...</b>	Opens the Advanced dialog to provide further search options. For more information on this dialog, please refer to Windows Help and Support.	

#### Notes

- Multiple Entries can be typed into the **Enter the object names to select** section; ensure that objects are separated by semicolons (for example: Name01; Name02)

### 5.10.2.2 Project Roles

People associated with a project play a *role* in analysis, design or implementation, such as Application Analyst, Architect, Developer and Project Manager. Project roles define the activities that resources can undertake.

**Access**    **Settings | Project Types | People > Project Roles**

#### Use to

- Define role types that are captured within Enterprise Architect

#### Reference

Field	Usage	See also
<b>Role</b>	Type or select the name of the role.	
<b>Description</b>	Type a description of the role .	
<b>Notes</b>	Type any additional information related to the role.	
<b>New</b>	Add further Roles.	
<b>Save</b>	Add the new role to the Defined Roles list.	
<b>Delete</b>	Removes a role from the Defined Role list.	
<b>Defined Roles</b>	<p>Review the Project Roles already defined.</p> <p>The Defined Roles list is available for selection for any element in the model; for example, you can select roles on the Project Authors tab of the People dialog, and the Resource Allocation tab of the Project Management window.</p> <p>You can also specify other roles on these dialogs, but such roles are not added to the Defined Roles list.</p>	<a href="#">Project Authors</a> <sup>1153</sup> <a href="#">Resource Allocation</a> <sup>512</sup>

#### Notes

- Deleting a role has no effect on any Project Author definition having this role; the deleted role becomes a simple text entry in the Project Author definition
- You can transport these role definitions between models, using the **Export Reference Data** and **Import Reference Data** options

Learn more

- [Export Reference Data](#)<sup>[376]</sup>
- [Import Reference Data](#)<sup>[380]</sup>

**5.10.2.3 Project Resources**

*Resources* are, for example, project authors, analysts, programmers and architects. That is, anyone who might work on the system over time, either adding to the model or programming and designing elements of the system outside Enterprise Architect.

**Access**    **Settings | Project Types | People > Project Resources**

Use to

- Record information on project resources

Reference

Field	Usage	See also
<b>Name</b>	Type or select the name of the person listed as a resource. The resource name is available for use in Resource Management.	<a href="#">Resource Management</a> <sup>[512]</sup>
<b>Organization</b>	Type the name of the organization employing the resource.	
<b>Role(s)</b>	Type or select the role the resource plays in the project (for example, Designer, Analyst, Architect).	
<b>Phone 1, Phone 2, Mobile, Fax</b>	Type the contact telephone numbers for the resource.	
<b>Email</b>	Type the email address for the resource.	
<b>Notes</b>	Type any additional notes on the resource.	
<b>Available Resources</b>	Review resources that have already been defined.	
<b>New</b>	Add further resources.	
<b>Save</b>	Add a new resource to the Available Resource list.	

<b>Delete</b>	Delete a resource from the Available Resources list.	

#### Notes

- You can transport these resource definitions between models, using the Export Reference Data and Import Reference Data options

#### Learn more

- [Export Reference Data](#)<sup>[376]</sup>
- [Import Reference Data](#)<sup>[380]</sup>

### 5.10.2.4 Project Clients

*Project clients* are the eventual owners of the software system.

**Access**    **Settings | Project Types | People > Project Clients**

#### Use to

- Capture client details associated with the current model

#### Reference

Field	Usage	See also
<b>Name</b>	Type or select the name of the client.	
<b>Organization</b>	Type the name of the organization that employs the client.	
<b>Role(s)</b>	Type the role the client plays in the project (for example, Manager, Sponsor).	
<b>Phone 1, Phone 2, Mobile, Fax</b>	Type the contact telephone numbers for the client.	
<b>Email</b>	Type the email address of the client.	
<b>Notes</b>	Type any additional notes on the client.	
<b>Defined Clients</b>	Review clients that have already been defined.	

<b>New</b>	Add details of further clients.	
<b>Save</b>	Add a new client to the Defined Client list.	
<b>Delete</b>	Delete a client record from the Defined Client list.	

### Notes

- You can transport these client definitions between models, using the Export Reference Data and Import Reference Data options

### Learn more

- [Export Reference Data](#) <sup>[376]</sup>
- [Import Reference Data](#) <sup>[380]</sup>

## 5.10.3 General Types

When you create or edit the properties of an element, you define the **type** or **status** of the property by selecting from a drop-down list that initially contains system-provided values. You can **add to or replace** any of these system values with your own **customized** values. The properties you can customize are:

- Status
- Constraint
- Constraint Status
- Difficulty
- Priority
- Test Status
- Requirement
- Scenario

Each of these can be separately managed through the General Types dialog.

**Access**   **Settings | Project Types | General Types**

### Learn more

- [Status Types](#) <sup>[1159]</sup>
- [Constraint Types](#) <sup>[1160]</sup>
- [Constraint Status Types](#) <sup>[1161]</sup>
- [Difficulty Types](#) <sup>[1163]</sup>
- [Priority Types](#) <sup>[1164]</sup>

- [Test Status Types](#) <sup>[1165]</sup>
- [Requirement Types](#) <sup>[1166]</sup>
- [Scenario Types](#) <sup>[1167]</sup>

### 5.10.3.1 Status Types

In an element Properties definition, the status of the element in the development management process is defined in the element **Status** field. You can select the appropriate value from a drop-down list, which initially contains the system values:

- **Approved**
- **Implemented**
- **Mandatory**
- **Proposed** and
- **Validated**

You can add to or replace any of these values with your own custom values. You can also assign a color band to each status type, and define the types of element that can display those colors.

**Access**    **Settings | Project Types | General Types > Status**

#### Customize Element Status Types

Field	Usage	See also
<b>Status</b>	Type the name of the status.	
<b>Description</b>	Type a short description of the status.	
<b>Status Type Color</b>	Using the drop-down arrow, select a color to be applied to the current status type.	
<b>Preview</b>	Displays a visual representation of an element with the selected status type color.	
<b>Restore Default</b>	Restore the status type color to its default setting.	
<b>Applies to...</b>	By default, status colors only apply to Requirement, Issue and Change elements. You might decide to also apply these colors to other UML elements, such as Use Cases or Classes.  To do this, click on the <b>Applies to...</b> button and, in the <b>Applied Status Colors</b> list, select the checkbox against each element type to which to apply the status colors.	<a href="#">Requirement</a> <sup>[1763]</sup> <a href="#">Issues (Defects)</a> <sup>[2631]</sup> <a href="#">Change</a> <sup>[2633]</sup>

<b>New</b>	Clear the dialog fields to create a new status.	
<b>Save</b>	Save the status, and add it to the <b>Type</b> list.	
<b>Delete</b>	Remove the currently selected status from the <b>Type</b> list.	
<b>Type</b>	Lists the current status types and descriptions for each status.	

### Notes

- To display status colors on your diagrams, you must select the **Show status colors on diagrams** checkbox on the Objects page of the Options dialog
- You can transport the status types (and the colors assigned to status types) between models, using the **Export Reference Data** and **Import Reference Data** menu options
- Requirement, Feature, Issue and Change elements have a status color **compartment**, whereas the status color for other elements is applied to the element **shadow**; to show the element shadows (and hence the status colors), select the **Element Shadows on** checkbox on the Diagram Appearance page of the Options dialog

### Learn more

- [General Settings](#) <sup>[958]</sup>
- [Export Reference Data](#) <sup>[376]</sup>
- [Import Reference Data](#) <sup>[380]</sup>
- [Object Display Options](#) <sup>[631]</sup>
- [Diagram Appearance Options](#) <sup>[622]</sup>

### 5.10.3.2 Constraint Types

In an element Properties definition, each constraint on the element is defined by type in the **Constraint** field. You can select the appropriate value from a drop-down list, which initially contains the system values:

- **Invariant**
- **Post-condition**
- **Pre-condition**
- **Process** and
- **OCL**

You can add to or replace any of these values with your own custom values.

Access    **Settings | Project Types | General Types > Constraint**

### Customize Constraint Types



Field/Button	Usage	See also
<b>Constraint</b>	Type the name of the constraint to define.	
<b>Description</b>	Type a brief description of the constraint.	
<b>Note</b>	Type any further details concerning the constraint.	
<b>New</b>	Clear the fields to define a new constraint.	
<b>Save</b>	Save the constraint definition and add it to the <b>Defined Constraint Types</b> list.	
<b>Delete</b>	Remove the currently selected constraint from the <b>Defined Constraint Types</b> list.	
<b>Defined Constraint Types</b>	Lists the currently-defined constraint types and their descriptions.	

#### Notes

- You can transport customized constraint types between models, using the **Export Reference Data** and **Import Reference Data** menu options

#### Learn more

- [Constraints](#)<sup>[963]</sup>
- [Export Reference Data](#)<sup>[376]</sup>
- [Import Reference Data](#)<sup>[380]</sup>

### 5.10.3.3 Constraint Status Types



In an element Properties definition, each constraint's status in the development process is defined in the constraint **Status** field. You can select the appropriate value from a drop-down list, which initially contains the system values:

- Implemented**
- Build**
- Validated**
- Approved**
- Mandatory** and
- Proposed**

You can add to or replace any of these values with your own custom values.

**Access**   **Settings | Project Types | General Types > Constraint Status Types**

### Customize Constraint Status Types

Field/Button	Usage	See also
<b>Status</b>	Type the name of the constraint status to define.	
<b>Make Default</b>	To make the selected constraint status the default for all constraint <b>Status</b> fields, select the checkbox.	
<b>New</b>	Clear the fields to define a new constraint status.	
<b>Save</b>	Save the constraint status definition and add it to the <b>Type</b> list.	
<b>Delete</b>	Remove the currently selected constraint status from the <b>Type</b> list.	
<b>Type</b>	Lists the currently available constraint status types.	
 	Move the selected constraint type higher or lower in the <b>Type</b> list.  This defines the ordering used in the <b>Status</b> drop down combo boxes, and in sorting any report or list when the Constraint Status Type is used as the sort-by property.	
<b>Sort Alphabetically</b>	Reorganize the entries in the <b>Type</b> list into alphabetical order.	
<b>Restore Defaults</b>	Remove all customized constraint status values and restore the system defaults.	

### Notes

- You can transport customized constraint status types between models, using the **Export Reference Data** and **Import Reference Data** menu options

### Learn more



- [Constraints](#) <sup>[963]</sup>
- [Export Reference Data](#) <sup>[376]</sup>
- [Import Reference Data](#) <sup>[380]</sup>

### 5.10.3.4 Difficulty Types

Developing a particular element might be a simple task or more complex. In the element Properties definition, the level of difficult of developing that element is defined in the **Difficulty** field. You can select the appropriate value from a drop-down list, which initially contains the system values **High**, **Medium** and **Low**. You can add to or replace any of these values with your own custom values.

**Access**    **Settings | Project Types | General Types > Difficulty**

#### Customize Difficulty Types

Field/Button	Usage	See also
<b>Difficulty</b>	Type the name of the Difficulty to define.	
<b>Make Default</b>	To make the selected Difficulty the default for all <b>Difficulty</b> fields, select the checkbox.	
<b>New</b>	Clear the fields to define a new Difficulty type.	
<b>Save</b>	Save the Difficulty definition and add it to the <b>Type</b> list.	
<b>Delete</b>	Remove the currently selected Difficulty from the <b>Type</b> list.	
<b>Type</b>	Lists the currently available Difficulty types.	
 	<p>Move the selected type higher or lower in the <b>Type</b> list.</p> <p>This defines the ordering used in the <b>Difficulty</b> drop down combo boxes, and in sorting any report or list when the Difficulty type is used as the sort-by property.</p> <p>For example: you have several Change elements on a diagram, and you select Diagram View; then you sort on the Difficulty column. The Diagram View entries are grouped by difficulty, in the order you defined in this <b>Type</b> field.</p>	
<b>Sort Alphabetically</b>	Reorganize the entries in the <b>Type</b> list into alphabetical order.	
<b>Restore Defaults</b>	Remove all customized Difficulty values and restore <b>High</b> , <b>Medium</b> and <b>Low</b> .	

#### Notes

- You can transport customized Difficulty types between models, using the **Export Reference Data** and **Import Reference Data** menu options

#### Learn more



- [Requirement Properties](#)<sup>[1774]</sup>
- [Export Reference Data](#)<sup>[376]</sup>
- [Import Reference Data](#)<sup>[380]</sup>

### 5.10.3.5 Priority Types

In an element Properties definition, the priority of developing that element is defined in the **Priority** field. You can select the appropriate value from a drop-down list, which initially contains the system values **High**, **Medium** and **Low**. You can add to or replace any of these values with your own custom values.

Access    **Settings | Project Types | General Types > Priority**

#### Customize Priority Types

Field/Button	Usage	See also
<b>Priority</b>	Type the name of the Priority to define.	
<b>Make Default</b>	To make the selected Priority the default for all <b>Priority</b> fields, select the checkbox.	
<b>New</b>	Clear the fields to define a new Priority.	
<b>Save</b>	Save the Priority definition and add it to the <b>Type</b> list.	
<b>Delete</b>	Remove the currently selected Priority from the <b>Type</b> list.	
<b>Type</b>	Lists the currently available Priority types.	
 	<p>Move the selected type higher or lower in the <b>Type</b> list.</p> <p>This defines the ordering used in the <b>Priority</b> drop down combo boxes, and in sorting any report or list when the Priority type is used as the sort-by property.</p> <p>For example: you have several Change elements on a diagram, and you select Diagram View; then you sort on the Priority column. The Diagram View entries are grouped by priority, in the order you defined in this <b>Type</b> field.</p>	
<b>Sort Alphabetically</b>	Reorganize the entries in the <b>Type</b> list into alphabetical order.	

<b>Restore Defaults</b>	Remove all customized Priority values and restore <b>High, Medium</b> and <b>Low</b> .	
-------------------------	--	--

**Notes**

- You can transport customized Priority types between models, using the **Export Reference Data** and **Import Reference Data** menu options

**Learn more**

- [Requirement Properties](#)<sup>[1774]</sup>
- [Export Reference Data](#)<sup>[376]</sup>
- [Import Reference Data](#)<sup>[380]</sup>

**5.10.3.6 Test Status Types**

All tests defined in the Testing window and Test Details dialog have a status value to show what point in the testing cycle the test has reached. The system values you can select from the test **Status** field are:



- Not Run**
- Pass**
- Fail**
- Deferred** and
- Canceled**

You can add to or replace any of these values with your own custom values.

**Access** **Settings | Project Types | General Types > Test Status**

**Customize Test Status Types**

Field/Button	Usage	See also
<b>Status</b>	Type the name of the Test status to define.	
<b>Make Default</b>	To make the selected Test status the default for all test <b>Status</b> fields, select the checkbox.	
<b>New</b>	Clear the fields to define a new Test status.	
<b>Save</b>	Save the Test status definition and add it to the <b>Type</b> list.	
<b>Delete</b>	Remove the currently selected Test status from the <b>Type</b> list.	

<b>Type</b>	Lists the currently available Test status types.	
 	<p>Move the selected type higher or lower in the <b>Type</b> list.</p> <p>This defines the ordering used in the test <b>Status</b> drop down combo boxes, and in sorting any report or list when the Test Status Type is used as the sort-by property.</p> <p>For example: you have run several tests on a package, and you run a Test Details report; then you sort on the Status column. The report entries are grouped by test status, in the order you defined in this <b>Type</b> field.</p>	
<b>Sort Alphabetically</b>	Reorganize the entries in the <b>Type</b> list into alphabetical order.	
<b>Restore Defaults</b>	Remove all customized Test status values and restore the system defaults.	

#### Notes

- You can transport customized Test status types between models, using the **Export Reference Data** and **Import Reference Data** menu options

#### Learn more

- [Working on Test Records](#) <sup>[2605]</sup>
- [Create Test Records](#) <sup>[2607]</sup>
- [Export Reference Data](#) <sup>[376]</sup>
- [Import Reference Data](#) <sup>[380]</sup>

### 5.10.3.7 Requirement Types

In a Requirement definition, the requirement is categorized according to type or function, in the **Type** field of the Requirement properties dialog. This helps to maintain a single set of typed requirements. You can select the appropriate **Type** value from a drop-down list, which initially contains the system values:

- **Display**
- **Functional**
- **Performance**
- **Printing**
- **Report**
- **Testing** and
- **Validate**

You can add to or replace any of these types with your own customized values.

**Access**   **Settings | Project Types | General Types > Requirement**

#### Customize Requirement Types

Field	Usage	See also
<b>Requirement</b>	Type the name of the requirement.	
<b>Description</b>	Type a short description of the requirement.	
<b>Weight</b>	Type a weighting to apply to the requirement type.	
<b>New</b>	Clears the dialog fields so that you can define a new requirement type.	
<b>Save</b>	Saves the requirement type details and adds it to the Defined Requirement Types list.	
<b>Delete</b>	Deletes a selected requirement type from the Defined Requirement Types list.	
<b>Defined Requirement Types</b>	Lists the currently available requirement types.	

#### Notes

- You can transport these requirement types between models, using the **Export Reference Data** and **Import Reference Data** menu options

#### Learn more

- [Requirement Properties](#)<sup>[1774]</sup>
- [Export Reference Data](#)<sup>[376]</sup>
- [Import Reference Data](#)<sup>[380]</sup>

### **5.10.3.8 Scenario Types**

In the Scenario definitions for an element each scenario is of a specific type, which you define in the **Type** field on the Scenario page of the Properties dialog. You can select the appropriate value from a drop-down list, which initially contains the system values:

- **Exception**
- **Alternate** and
- **Basic Path**

You can add to or replace any of these system types with your own customized values.

**Access**   **Settings | Project Types | General Types > Scenario**

### Customize Scenario Types

Field	Usage	See also
<b>Scenario Type</b>	Type the name of the scenario type.	
<b>Description</b>	Type a short description of the scenario type.	
<b>Weight</b>	Type a value for the weighting to apply to the scenario type.	
<b>Notes</b>	Type any additional information to describe the scenario type.	
<b>New</b>	Clears the dialog fields so that you can define a new Scenario type.	
<b>Save</b>	Saves the scenario type and adds it to the <b>Defined Scenario Types</b> list.	
<b>Delete</b>	Deletes a selected scenario type from the <b>Defined Scenario Types</b> list.	
<b>Defined Scenario Types</b>	Lists the currently available scenario types to offer in the drop-down list of a <b>Scenario Type</b> field.	

### Notes

- You can transport customized scenario types between models, using the **Export Reference Data** and **Import Reference Data** menu options

### Learn more

- [Scenarios](#)<sup>[965]</sup>
- [Export Reference Data](#)<sup>[376]</sup>
- [Import Reference Data](#)<sup>[380]</sup>



### 5.10.4 Metrics and Estimation

TCF values, EFC values and Default Hour Rate for a project are controlled from the Estimation Factors dialog.

Risk, metric and effort types for a project are controlled from the Project Indicators dialog.

For further information on these see the [Project Management](#)<sup>[509]</sup> and [Resource Management](#)<sup>[512]</sup> topics, or specifically:

- [Technical Complexity Factors](#)<sup>[585]</sup>
- [Environment Complexity Factors](#)<sup>[586]</sup>
- [Default Hours](#)<sup>[588]</sup>
- [Effort Types](#)<sup>[522]</sup>
- [Metric Types](#)<sup>[523]</sup>
- [Risk Types](#)<sup>[524]</sup>

### 5.10.5 Maintenance

**Access**    **Settings | Project Types | Maintenance**

**Use to**

- Set the base Problem Types that are handled

**Learn more**

- [Testing Types](#)<sup>[1170]</sup>

#### 5.10.5.1 Problem Types

**NOT CURRENTLY USED**

For the maintenance and change control screens, you can use the Maintenance dialog to set the base Problem Types that are handled. Examples are hardware-related issues, performance problems, software bugs and network problems.

**Access**    **Settings | Project Types | Maintenance > Problem Types**

**Use to**

- Set the base Problem Types that are handled

**Reference**

Field	Usage	See also
<b>Problem Type</b>	Indicates the name of the problem type.	

<b>Description</b>	Indicates a short description of the problem type.	
<b>Weight</b>	Indicates a weighting to apply to the problem type.	
<b>Note</b>	Indicates any additional information on the problem type.	
<b>Defined Types</b>	Displays all of the pre-defined and saved problem types.	
<b>New</b>	Add a new problem type.	
<b>Save</b>	Saves the scenario details and adds it to the Defined Types list.	
<b>Delete</b>	Deletes a scenario from the Defined Types list.	

#### Notes

- You can transport these problem types between models, using the Export Reference Data and Import Reference Data options
- You transport the problem types together with test types as a *Maintenance Types* file

#### Learn more

- [Export Reference Data](#)<sup>[376]</sup>
- [Import Reference Data](#)<sup>[380]</sup>

### 5.10.5.2 Testing Types

Use the Test Types tab of the Maintenance dialog to add testing types to the basic set that comes with Enterprise Architect. Typical test types are load tests, performance tests and function tests.

**Access**    **Settings | Project Types | Maintenance > Test Types**

#### Use to

- Add Testing Types

#### Reference

Field	Usage	See also
<b>Test Type</b>	Type the name of the test type.	

<b>Description</b>	Type a short description of the test type.	
<b>Weight</b>	Type a weighting to apply to the test type.	
<b>Note</b>	Type any additional information on the test type.	
<b>Defined Types</b>	Displays all of the pre-defined and saved test types.	
<b>New</b>	Add a new test type.	
<b>Save</b>	Saves the scenario details and adds it to the Defined Types list.	
<b>Delete</b>	Deletes a scenario from the Defined Types list.	

**Notes**

- You can transport these test types between models, using the Export Reference Data and Import Reference Data options
- You can either export the test types together with the default problem types, as a *Maintenance Types* file, or separately as a *Test Types* file

**Learn more**

- [Export Reference Data](#)<sup>[376]</sup>
- [Import Reference Data](#)<sup>[380]</sup>

**5.10.6 Data Types**

Different programming languages support different inbuilt data types. It is possible for you to manage and extend this set of inbuilt data types for a standard programming language, as well as define new programming languages for use within Enterprise Architect.

**Access**    **Settings > Code Engineering Datatypes**

**Programming Language Datatype Options**

Field	Usage	See also
<b>Product Name</b>	Click on the drop-down arrow and select the name of the programming language.	
<b>Add Product</b>	Click on this button to display a prompt for the name of a new programming	

Field	Usage	See also
	language. Type the name and click on the <b>OK</b> button.	
<b>Datatype</b>	Type the language-specific name of the datatype.	
<b>Common Type</b>	Type the common (or generic) name of the datatype; for example, the Java <i>boolean</i> datatype has a common datatype <i>Boolean</i> .	
<b>New</b>	Click on this button to clear the fields to create a new datatype.	
<b>Save</b>	Click on this button to save the newly created or updated datatype.	
<b>Delete</b>	Click on this button to delete the selected datatype.  You can delete your own custom data types, but you cannot delete any of the predefined data types.	
<b>Defined Datatypes for Programming Languages</b>	Review the list of datatypes for the selected language, and click on any that you want to edit or delete. The details for the selected datatype display in the dialog fields.	

### Notes

- You do not need to set the options in the Size panel
- Once you have defined at least **one** datatype for a new language, that language name is added to the drop-down list for the **Language** fields in the Programming Languages Datatypes dialog, in the Code Template Editor, and in the Properties dialog for each Class element within the model
- You can transport these data types between models, using the **Export Reference Data** and **Import Reference Data** menu options
- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Configure Datatypes** permission to update and delete data types

### Learn more






- [Code Template Editor](#)<sup>[1641]</sup>
- [Export Reference Data](#)<sup>[376]</sup>
- [Import Reference Data](#)<sup>[380]</sup>
- [Permission List](#)<sup>[1130]</sup>

### 5.10.7 Resources









The Resources window is a repository of project-standard and re-usable document generation templates, MDG Technologies, Relationship Matrix profiles, commonly-used model elements, stylesheets, UML Profiles and UML Patterns. The window provides facilities for creating and modifying these structures and facilities and, in some cases, for applying the facility to the current model directly from the window.







**Access** **Project | Resources** (Alt+6)

#### Resources available through the Resources window

Resource	Description	See also
<b>Document Generation</b>	<p>The <b>Document Generation</b> folder holds a number of sub-folders, each of which provides the facilities for generating a particular type of document or report from your model.</p> <ul style="list-style-type: none"> <li> <b>System Templates</b> - this sub-folder itself has sub-folders containing system-defined report <b>templates</b>, <b>Fragments</b>, <b>Cover Pages</b>, <b>Style Sheets</b> and <b>Table of Contents definitions</b> <p>You can right-click on these template names and select options to display the template contents or to copy the template as the basis of a user-defined template that you work on in the <b>Document Template Editor</b>, under a <b>different name</b>; you can also select the <b>Override</b> option to create a copy of the system template as a user template with the <b>same</b> name</p> </li> <li> <b>User Templates</b> - this sub-folder also has sub-folders for report templates, Fragments, Cover Pages, Style Sheets and Tables of Contents, but these are the templates that you have created yourself using the Document Template Editor, either as new or by copying and editing another template <p>You can right-click on these template names and select options to create, delete, rename, modify, move or copy a user-defined template; you have additional options to <b>Mark</b> a report template as a Fragment, or a Fragment as a report template, if they are suitable for this reclassification</p> <p>You can also 'drag and drop' user-defined templates between user-defined groups, and between these groups and the appropriate standard 'type' folders; for example, from <i>DavidCover</i> to <i>Cover Pages</i></p> </li> <li> <b>Technology Templates</b> - If you have loaded an MDG Technology containing document templates, those are also listed under the Technology name </li> <li> <b>Web Style Templates</b> - this sub-folder contains the <b>HTML style templates</b> that you have created as alternatives to the system-supplied <b>&lt;default&gt;</b> template </li> </ul>	<p><a href="#">Document Templates</a>  <sup>[2681]</sup></p> <p><a href="#">Template Fragments</a>  <sup>[2708]</sup></p> <p><a href="#">Design Custom Document Templates</a>  <sup>[2684]</sup></p> <p><a href="#">Create Web Style Templates</a>  <sup>[2747]</sup></p> <p><a href="#">Resource Documents</a>  <sup>[2668]</sup></p>

Resource	Description	See also
	<p>To create a new template, right-click on the folder name and select the <b>Create HTML template</b> option, and provide a template name; this opens the <b>HTML and CSS Style Editor</b>, in which you create the template from HTML fragments</p> <p>You can right-click on the template names and select options to delete the template or modify the template contents</p> <ul style="list-style-type: none"> <li>• <b>Defined Documents</b> - this folder holds the <b>Resource documents</b> (report specifications) you have created in the Document Report Generator; you can right-click on the document name to generate documents individually or as a batch, open the generated document, or delete the report specification</li> <li>• <b>Linked Document Templates</b> - this folder holds the <b>non-standard</b> Linked Document templates, either user-defined or supplied with each MDG Technology enabled on the system; you can edit or delete any of these existing templates, or create new ones from scratch or by copying an existing template, including the system-provided linked document templates</li> </ul> <p>You can also re-assign the templates to different template groups, either another of the existing groups or a new one that you create as part of the assignment</p> <ul style="list-style-type: none"> <li>• <b>Legacy Templates</b> - if you are, or have been, using the original (legacy) Enterprise Architect report generator, this sub-folder contains the report templates that you use to generate the documents</li> </ul> <p>You can continue to create, modify, use and delete legacy style templates, but it is recommended that you switch to the later report generator</p>	<p><a href="#">Linked Documents</a> <small>[1044]</small></p> <p><a href="#">Create Linked Document Templates</a> <small>[1098]</small></p> <p><a href="#">The Legacy Report Generator</a> <small>[2720]</small></p> <p><a href="#">Legacy Report Style Templates</a> <small>[2726]</small></p>
<b>MDG Technologies</b>	<p>The MDG Technologies folder lists the MDG Technologies that have been imported directly into the Resources window. Each technology extends the modeling capabilities and facilities of Enterprise Architect.</p> <p>You can import Technologies into your project by right-clicking on the MDG Technologies folder and selecting the <b>Import Technology</b> menu option. The Copy Technology to Application Data dialog then gives you the option to import the Technology to:</p> <ul style="list-style-type: none"> <li>• <b>User</b>, for your individual use; the Technology is imported into the %APPDATA% folder and is not listed in the Resources window</li> <li>• <b>Model</b>, for all project users to access; the Technology is imported into the Resources window</li> </ul> <p>If you have any MDG Technologies in the folder, each has its own subfolder containing the Profiles, Patterns and code modules</p>	<p><a href="#">MDG Technologies</a> <small>[1475]</small></p> <p><a href="#">Import MDG Technologies to Model</a> <small>[1480]</small></p>

Resource	Description	See also
	<p>used in the technology. You can right-click on these and select options to delete the technology, review the Patterns or Profiles, or invoke the code modules.</p> <p>If you remove the technology, it is removed from the Resources window, the MDG Technologies dialog, and the model.</p> <p>Consider the fact that some MDG Technologies can be large and might impose some delays on the workstation as they load each time a user connects to the model.</p>	<a href="#">Manage MDG Technologies</a>  <sup>[1477]</sup>
<b>Matrix Profiles</b>	<p>The Matrix Profiles folder contains a list of Relationship Matrix Profiles that you or your team members have created using the Relationship Matrix or Specification Manager. You simply double-click on a Matrix Profile name to quickly open the Relationship Matrix under the settings defined in the Profile.</p>	<a href="#">Matrix Profiles</a>  <sup>[735]</sup> <a href="#">Create Matrix Profile</a>  <sup>[1744]</sup>
<b>Favorites</b>	<p>The Favorites folder provides a shortcut to each commonly-used element that you have added to the folder, so that you can quickly and easily locate the elements and add them to diagrams.</p>	<a href="#">Favorites</a>  <sup>[1176]</sup>
<b>Stylesheets</b>	<p>The Stylesheets folder contains <b>XSL stylesheets</b> that you or your team members have imported into the model. These XSL stylesheets support the optional step of converting XML files exported from the model into alternative formats such as HTML, XSL or source code. Stylesheets imported here are then available in the drop-down lists on the XML Export dialog and the Generate GML Application Schema dialog.</p> <p>Combined with UML Profiles, this is a powerful means of extending Enterprise Architect to generate almost any content required.</p> <p>To import a stylesheet:</p> <ol style="list-style-type: none"> <li>1. Right-click on the folder name and select the <b>Import Stylesheet</b> option.</li> <li>2. Locate and select the stylesheet file in the browser, and click on the <b>Open</b> button.</li> <li>3. Provide a reference name for the file, and click on the <b>OK</b> button.</li> </ol> <p>Once the stylesheet is in the folder, you can select context menu options to reload it (if it has been changed at the source) or delete it.</p>	<a href="#">Publish Model Package</a>  <sup>[476]</sup> <a href="#">Generate GML Application Schema</a>  <sup>[1987]</sup>
<b>UML Profiles</b>	<p>A Profile is a source of modified or extended element and connector types that you have developed from the UML base types. You can import each new Profile into this folder, and then create elements and connectors of the new types by dragging the icons from the Profile onto a diagram.</p> <p>Once you have Profiles in the folder, you can select options for arranging them in order of name or type, viewing a Profile</p>	<a href="#">UML Profiles in the Resources Window</a>  <sup>[1526]</sup> <a href="#">Import UML Profiles Into the Resources Window</a>  <sup>[1526]</sup>

Resource	Description	See also
	<p>description, or deleting a Profile.</p> <p>It is possible to synchronize the Tagged Values and constraints for any elements created from a Profile element in the Resources window.</p> <p>Importing and applying Profiles through the Resources window is <b>no longer recommended</b>; from Enterprise Architect release 7.0. it is recommended that you load and apply Profiles as components of MDG Technologies.</p>	<p><a href="#">Synchronize Tagged Values and Constraints</a>  <sup>1473</sup></p> <p><a href="#">Developing Profiles</a>  <sup>1485</sup></p>
<b>UML Patterns</b>	<p>The UML Patterns folder is a folder of UML Design Patterns that you or your team members have imported into the model. A Pattern captures complex new elements and features that you can drag into a diagram without having to retype or reconfigure each element.</p> <p>Within the folder, the Patterns are grouped by type. Once you have Patterns within a group, you can select options to delete all Patterns in the group or just the selected Pattern, view a Pattern's details, or apply a Pattern to an empty diagram.</p>	<p><a href="#">Design Patterns</a>  <sup>1464</sup></p> <p><a href="#">Create a Pattern</a>  <sup>1464</sup></p> <p><a href="#">Import a Pattern</a>  <sup>1466</sup></p> <p><a href="#">Use a Pattern</a>  <sup>1467</sup></p>

### Notes

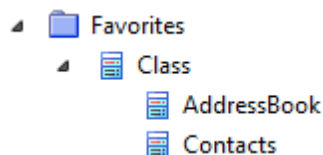
- In the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Configure Resources** permission to maintain Resources window items
- For a number of folders in the Resources window, the right-click context menu also offers a **Help** option that, when you select it, displays a brief description of the folder

### Learn more

- [Permission List](#)  <sup>329</sup>

#### 5.10.7.1 Favorites


If you have any elements that you or your team use frequently - such as the list of Actors in a system - you can add them to the **Favorites** folder in the Resources window, and use that as the source for conveniently dragging and dropping instances of or links to these elements into other diagrams. This makes it much easier to locate and use the elements as you create and manage your model.



**Access**   **Project | Resources | Favorites**

#### Actions on the Favorites Folder



Action	Detail	See also
<b>Add an element to the Favorites Folder</b>	<ul style="list-style-type: none"> <li>In a diagram, right-click on the element to add</li> <li>From the context menu select the <b>Find   Add to Favorites</b> option</li> <li>Switch to the Resources window and expand the <i>Favorites</i> folder; the added element is listed within its element-type category (such as <b>Class</b>)</li> </ul> <p>You add elements to the Favorites folder individually.</p>	<a href="#">Finding Elements</a> 
<b>Delete an element from the Favorites Folder</b>	<ul style="list-style-type: none"> <li>Right-click on the element within the <i>Favorites</i> folder in the Resources window</li> <li>From the context menu select the <b>Delete Favorite</b> option</li> <li>A confirmation prompt displays; click on the <b>Yes</b> button</li> </ul>	
<b>View Properties of an element in the Favorites Folder</b>	<ul style="list-style-type: none"> <li>Select and right-click on the element in the folder</li> <li>From the context menu, select the <b>Element Properties</b> option; the element Properties dialog displays</li> </ul>	

**Notes**

- When you have had the Favorites folder open for some time and other users might have been adding to it, right-click on the folder name and select the **Refresh Favorites** option to update your view of the folder with the new contents

**Learn more**

- [Resources](#) 

**Part**

---

**VI**

## 6 Standard UML Models



This topic provides an introduction to Enterprise Architect's diagrams, elements and connectors, and its modeling process.

It also illustrates its alignment to the Unified Modeling Language (UML) 2.4.1, an open modeling standard, defined and maintained by the Object Management Group.

Topic	Detail	See also
<b>The Unified Modeling Language (UML)</b>	<p>The UML standard defines notations and rules for specifying business and software systems; the notation supplies a rich set of graphic elements for modeling object oriented systems, and the rules state how those elements can be connected and used.</p> <p>UML is not a tool for creating software systems; instead, it is a visual language for communicating, modeling, specifying and defining systems.</p> <p>UML is not a prescriptive process for modeling software systems; it does not supply a method or process, simply the language. You can therefore use UML in a variety of ways to specify and develop your software engineering project.</p> <p>This language is designed to be flexible, extendable and comprehensive, yet generic enough to serve as a foundation for all system modeling requirements. With its specification, there is a wide range of elements characterized by the kinds of diagrams they serve, and the attributes they provide. All can be further specified by using stereotypes, Tagged Values and profiles.</p> <p>Enterprise Architect supports many different kinds of UML elements (as well as some custom extensions); together with the connectors between elements, these form the basis of the model.</p>	<a href="#">UML Diagrams</a> <small>[1181]</small> <a href="#">UML Elements</a> <small>[1265]</small> <a href="#">UML Connectors</a> <small>[1389]</small> <a href="#">UML Stereotypes</a> <small>[1452]</small> <a href="#">Tagged Values</a> <small>[1134]</small>
<b>Wide Range of Applications</b>	<p>Although initially conceived as a language for software development, UML can be used to model a wide range of real world domains and processes (in business, science, industry, education and elsewhere), organizational hierarchies, deployment maps and much more.</p> <p>Enterprise Architect also provides additional custom diagrams and elements, to address further modeling interests.</p>	<a href="#">Modeling Basics</a> <small>[750]</small>
<b>Extending UML for New Domains</b>	<p>Using UML Profiles, UML Patterns, Grammars, Data Types, Constraints, MDG Technologies and other extensions, UML and Enterprise Architect can be tailored to address a particular modeling domain not explicitly covered in the original UML specification</p> <p>Enterprise Architect makes extending UML simple and</p>	<a href="#">Using UML Profiles</a> <small>[1472]</small> <a href="#">Design Patterns</a> <small>[1464]</small> <a href="#">MDG</a>

Topic	Detail	See also
	straightforward and, best of all, the extension mechanism is still part of the UML Specification.	<a href="#">Technologies</a> <small>[1475]</small>
<b>Recommended Reading</b>	<p>In addition to the UML Specification available from the OMG, two books that provide excellent introductions to UML are:</p> <ul style="list-style-type: none"> <li>• <i>Schaum's Outlines: UML</i> by Bennet, Skelton and Lunn Published by McGraw Hill. ISBN 0-07-709673-8</li> <li>• <i>Developing Software with UML</i> by Bern Oestereich Published by Addison Wesley. ISBN 0-201-36826-5</li> </ul>	

**Learn more**

- [OMG website](#) (Online Resource)
- Diagram [Toolbox](#)[792] descriptions
- The *EAExample.eap* project supplied with Enterprise Architect
- Online [UML Tutorial](#) (parts 1 and 2) and [UML 2.0 Tutorial](#)

## 6.1 UML Diagrams

A **UML diagram** is a representation of the components or elements of a system or process model and, depending on the type of diagram, how those elements are connected or how they interact from a particular perspective. For example, how and why an object changes state, or how requirements are realized by the process or a system.

Diagrams are developed in the main workspace (Diagram View), in which you create and connect model elements. You create diagrams by right-clicking a Package and selecting the **New Diagram** context menu option, or load them by double-clicking on their diagram icon in the Project Browser.

### Diagram Grouping

Group	Detail	See also
<b>Structural Diagrams</b>	Structural Diagrams depict the structural elements composing a system or function, reflecting the static relationships of a structure, or run-time architectures.	<a href="#">Structural Diagrams</a> <small>[1182]</small>
<b>Behavioral Diagrams</b>	Behavioral Diagrams show a dynamic view of the model, depicting the behavioral features of a system or business process.	<a href="#">Behavioral Diagrams</a> <small>[1198]</small>
<b>Extended Diagrams</b>	Enterprise Architect provides a set of additional diagram types that extend the core UML diagrams for domain-specific models.	<a href="#">Extended Diagrams</a> <small>[1793]</small>
<b>Custom Diagrams</b>	Enterprise Architect also supports diagram types specific to MDG Technologies, including integrated technologies.	<a href="#">Specialized UML Models</a> <small>[1789]</small>

### Learn more

- [Diagram Tasks](#) [820]

## 6.2 UML Structural Models

**UML Structural diagrams** depict the structural elements composing a system or function. These diagrams reflect the **static relationships** of a **structure** (Class or Package diagrams) or **run-time architecture** (Object or Composite Structure diagrams).

### Structural Diagram types

Diagram Type	Detail	See also
<b>Class</b>	Class diagrams capture the logical structure of the system, the Classes and objects that make up the model, describing what exists and what attributes and behavior it has.	<a href="#">Class diagrams</a> <sup>[1184]</sup>
<b>Composite Structure</b>	Composite Structure diagrams reflect the internal collaboration of Classes, Interfaces and Components (and their properties) to describe a functionality.	<a href="#">Composite Structure diagrams</a> <sup>[1188]</sup>
<b>Component</b>	Component diagrams illustrate the pieces of software, embedded controllers and such that make up a system, and their organization and dependencies.	<a href="#">Component diagrams</a> <sup>[1194]</sup>
<b>Deployment</b>	Deployment diagrams show how and where the system is to be deployed; that is, its execution architecture.	<a href="#">Deployment diagrams</a> <sup>[1191]</sup>
<b>Object</b>	Object diagrams depict object instances of Classes and their relationships at a point in time.	<a href="#">Object diagrams</a> <sup>[1186]</sup>
<b>Package</b>	Package diagrams depict the organization of model elements into Packages and the dependencies amongst them.	<a href="#">Package diagrams</a> <sup>[1182]</sup>
<b>Profile</b>	Profile Diagrams are those created in a «profile» Package, to extend UML elements, connectors and components.	<a href="#">Profile Diagrams</a> <sup>[1196]</sup>




















### 6.2.1 Package Diagram

**Package diagrams** depict the organization of model elements into Packages and the dependencies amongst them, including Package imports and Package extensions. They also provide a visualization of the corresponding namespaces. You can use them to:

**Example Diagram** [Example Package Diagram](#) <sup>[1183]</sup>

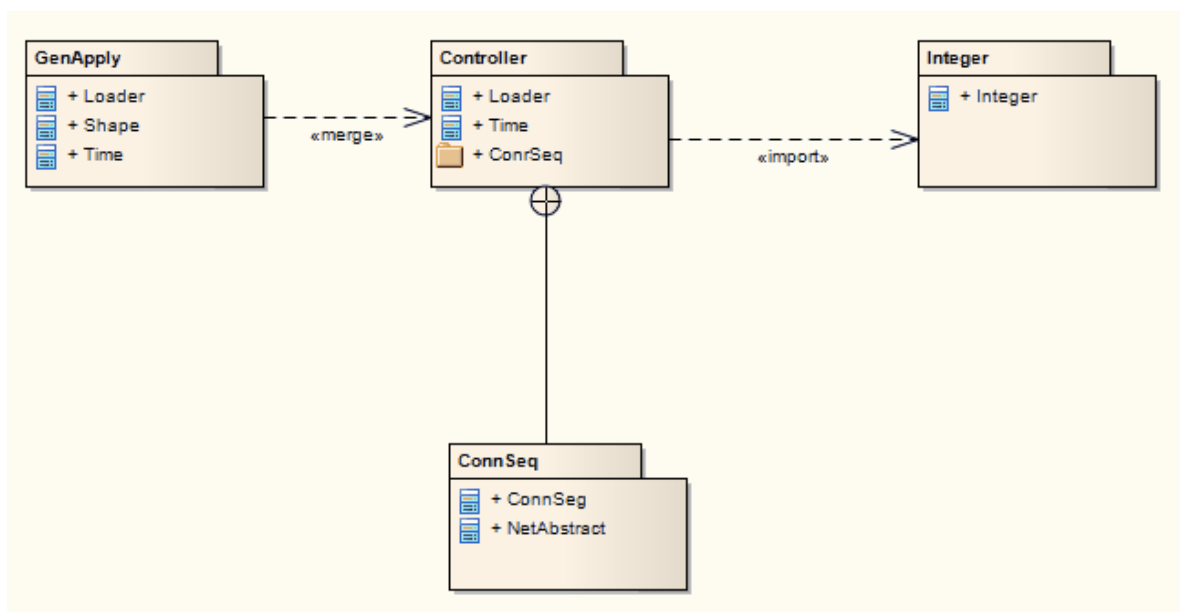
#### Tools

Select Package diagram elements and connectors from the Class pages of the Toolbox.

Package Diagram Elements	Package Diagram Connectors
 Package	 Associate
 Class	 Generalize
 Interface	 Compose
 Data Type	 Aggregate
 Enumeration	 Association Class
 Primitive	 Assembly
 Table	 Realize
 Signal	 Nesting
 Association	 Package Merge
	 Package Import

### 6.2.1.1 Example Package Diagram

The following example demonstrates a basic Package diagram.



Item	Description	See also
Nesting connector between <i>ConnSeq</i> and <i>Controller</i>	<p>The nesting connector between <i>ConnSeq</i> and <i>Controller</i> reflects what the Package contents reveal.</p> <p>Package contents can be listed by clicking on the diagram background to display the diagram's Properties dialog, selecting the Elements tab and selecting the <b>Package Contents</b> checkbox.</p>	<a href="#">Properties</a> <sup>[823]</sup> dialog
«import» connector	<p>The «import» connector indicates that the elements within the target Integer Package, which in this example is the single Class <i>Integer</i>, are imported into the Package <i>Controller</i>.</p> <p>The <i>Controller</i>'s namespace gains access to the <i>Integer</i> Class; the <i>Integer</i> namespace is not affected.</p>	
«merge» connector	<p>The «merge» connector indicates that the package <i>Controller</i>'s elements are imported into <i>GenApply</i>, including <i>Controller</i>'s nested and imported contents.</p> <p>If an element already exists within <i>GenApply</i>, such as <i>Loader</i> and <i>Time</i>, these elements' definitions are expanded by those included in the package <i>Controller</i>. All elements added or updated by the merge are noted by a generalization relationship back to that package.</p>	











## 6.2.2 Class Diagram

The **Class diagram** captures the logical structure of the system - the Classes - and things that make up the model. It is a static model, describing what exists and what attributes and behavior it has, rather than how something is done. On a Class diagram you can illustrate relationships between Classes and Interfaces using Generalizations, Aggregations and Associations, which are valuable in reflecting inheritance, composition or usage, and connections respectively.














**Example Diagram** [Example Class Diagram](#) <sup>[1185]</sup>

### Tools

Select Class diagram elements and connectors from the Class pages of the Toolbox.

Class Diagram Elements	Class Diagram Connectors
 Package	 Associate
 Class	 Generalize
 Interface	 Compose
 Data Type	 Aggregate
 Enumeration	 Association Class



Class Diagram Elements	Class Diagram Connectors
 Primitive	 Assembly
 Table	 Realize
 Signal	 Template Binding
 Association	 Nesting
	 Package Merge
	 Package Import
	 Abstraction
	 Substitution
	 Usage

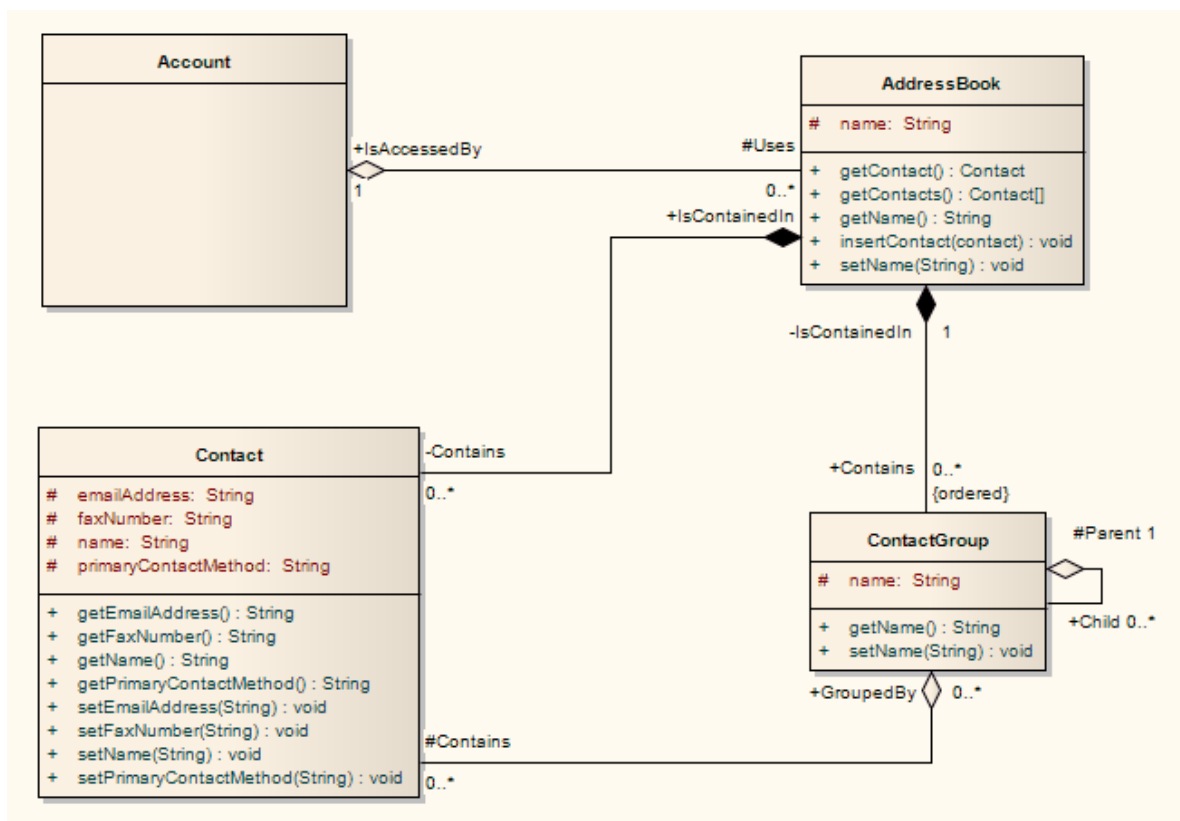
### Learn more

- [Active Classes](#) <sup>[1365]</sup>
- [Parameterized \(template\) Classes](#) <sup>[1365]</sup>
- [Stereotyped Business Interaction Elements](#) <sup>[1805]</sup>

### 6.2.2.1 Example Class Diagram

In this example Class diagram, there are two forms of the [Aggregation](#) <sup>[1392]</sup> relationship:

- The pale form indicates that the Class *Account* uses *AddressBook*, but does not necessarily contain *AddressBook*
- The dark *Composite* Aggregation form indicates ownership or containment by the target Classes (at the diamond end) of the source Classes



### 6.2.3 Object Diagram





An **Object diagram** is closely related to a Class diagram, with the distinction that it depicts **object instances** of Classes and their relationships at a point in time. Object diagrams do not reveal architectures varying from their corresponding Class diagrams, but reflect multiplicity and the roles instantiated Classes could serve. They are useful in understanding a complex Class diagram, by creating different cases in which the relationships and Classes are applied.








This might appear similar to a Composite Structure diagram, which also models run-time behavior; the difference is that Object diagrams exemplify the **static** Class diagrams, whereas Composite Structure diagrams reflect **run-time** architectures different from their static counterparts. An Object diagram can also be a kind of Communication diagram (which also models the connections between objects, but additionally sequences events along each path).

**Example Diagram** [Example Object Diagram](#) 1187

#### Tools

Select Object diagram elements and connectors from the Object pages of the Toolbox.

Object Diagram Elements	Object Diagram Connectors
 <b>Actor</b>	 <b>Information Flow</b>
 <b>Object</b>	 <b>Associate</b>

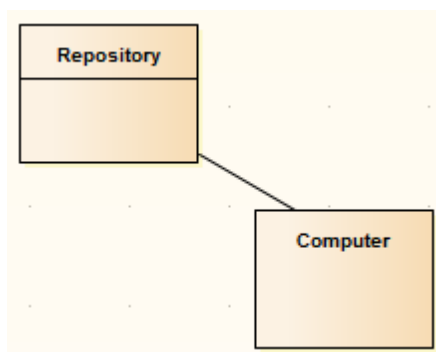
Object Diagram Elements	Object Diagram Connectors
 Collaboration	 Dependency
 Collaboration Use	
 Information Item	
 Boundary	
 Control	
 Entity	

#### Learn more

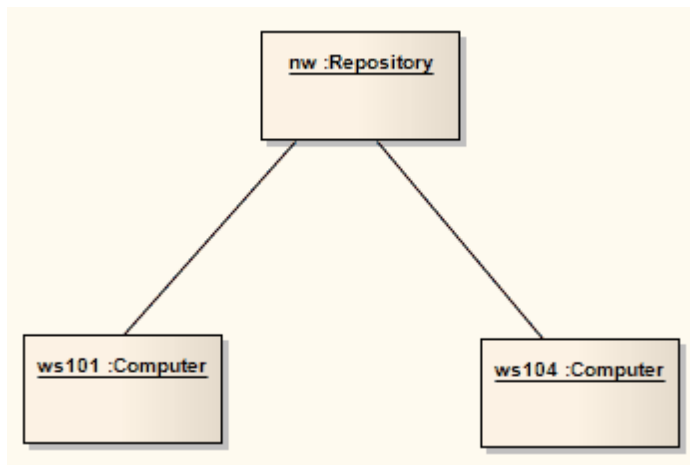
- [Business Modeling Objects](#) <sup>[1805]</sup>
- [Communication diagram](#) <sup>[1259]</sup>

### 6.2.3.1 Example Object Diagram

This example shows a simple Class diagram, with two Class elements connected.



These Classes are instantiated as Objects in an Object diagram. There are two instances of *Computer* in this model, demonstrating the usefulness of Object diagrams in considering the relationships and interactions Classes might have in practice.



## 6.2.4 Composite Structure Diagram














A **Composite Structure diagram** reflects the **internal collaboration** of Classes, Interfaces or Components (and their properties) to describe a functionality. Composite Structure diagrams are similar to Class diagrams, but whilst Class diagrams model a static view of Class structures, including their attributes and behaviors, Composite Structure diagrams model a **specific usage** of the structure. You can use them to express run-time architectures, usage patterns and the participating elements' relationships, which might not be reflected by static diagrams.

In a Composite Structure diagram, Classes are accessed as Parts or run-time instances fulfilling a particular role. These Parts can have multiplicity, if the role filled by the Class requires multiple instances. Ports defined by a Part's Class should be represented in the composite structure, so that all connecting Parts provide the required interfaces specified by the Port. There is extensive flexibility, and a consequent complexity, that come with modeling composite structures. To optimize your modeling, consider building Collaborations to represent reusable patterns responding to your design issues.

**Example Diagram** [Example Composite Structure Diagram](#) 

### Tools

Select Composite Structure diagram elements and connectors from the Composite pages of the Toolbox.

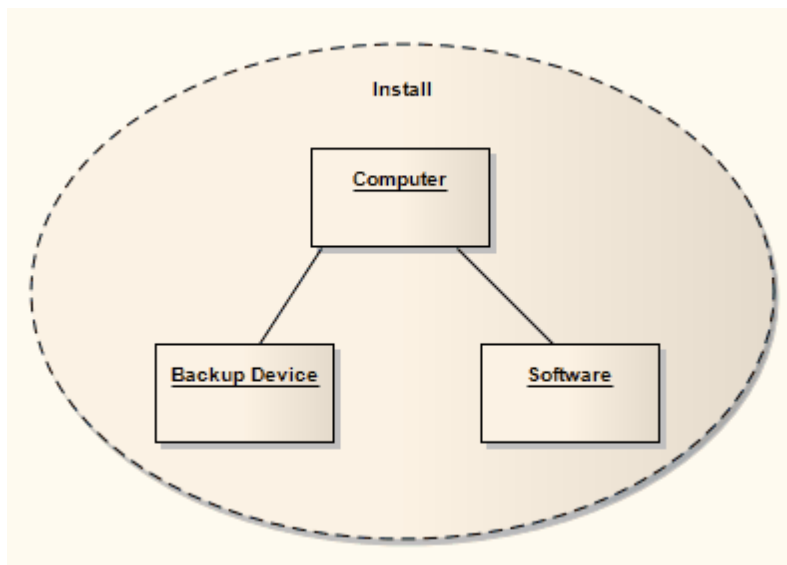
Composite Structure Diagram Elements	Composite Structure Diagram Connectors
 Class	 Connector
 Interface	 Assembly
 Part	 Role Binding
 Port	 Represents
 Collaboration	 Occurrence
 Collaboration Use	 Delegate
 Expose Interface	

Learn more

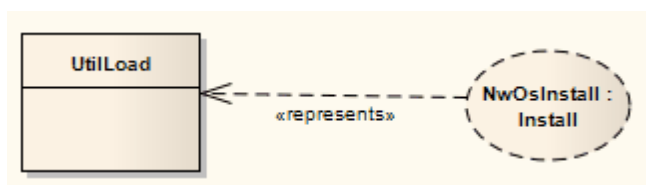
- [Business Use Case Realization](#) <sup>[1805]</sup>

**6.2.4.1 Example Composite Structure Diagram**

This diagram shows a Collaboration, used in Composite Structure diagrams to model common patterns. This particular example shows a relationship for performing an installation.



The next diagram uses this *Install* Collaboration in a Collaboration Use, and applies it to the *UtilLoad* Class via a «represents» relationship. This indicates that the classifier *UtilLoad* uses the Collaboration pattern within its implementation.

Learn more

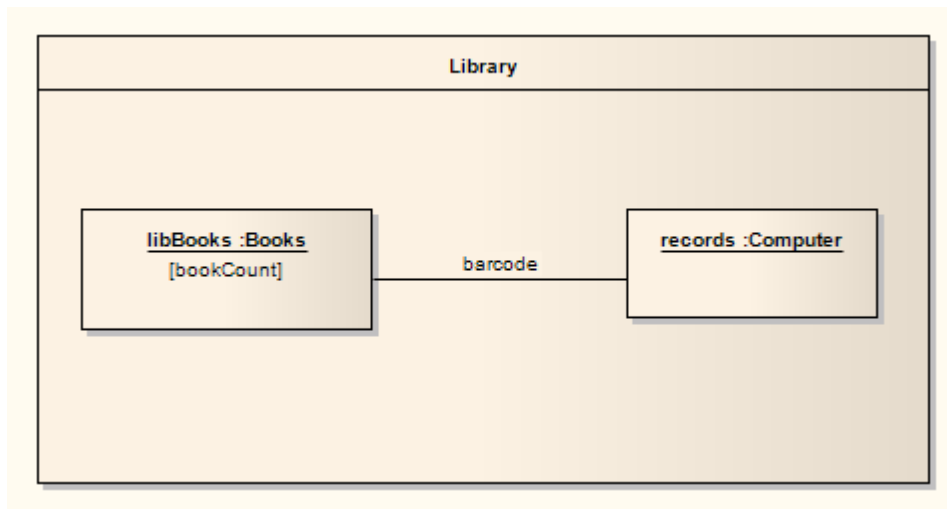
- [Collaboration Use](#) <sup>[1368]</sup>

**6.2.4.2 Properties**

A **property** is a nested structure within a classifier, usually a Class or an Interface, on a Composite Structure diagram. The contained structure reflects instances and relationships reflected within the containing classifier. Properties can have multiplicity, and can be displayed as:

- Parts (preferred) or
- Association Roles

### Parts



In this diagram there are two Parts, *libBooks* and *records*, which are instances corresponding to the Classes *Books* and *Computer* respectively. The relationship between the two Parts is indicated by the connector, reflecting that communication between the Parts is via the *barcode*. This contained structure and its Parts are properties owned by the *Library* Class.

After dragging Parts from the Diagram Toolbox onto the Class, right-click on a Part and select **Advanced | Set Property Type** to connect to a classifier. If Parts disappear when dragged onto the Class, adjust the Z-order of the Class to move it behind the Parts (right-click on the Class and select the **Z-Order** context menu option).

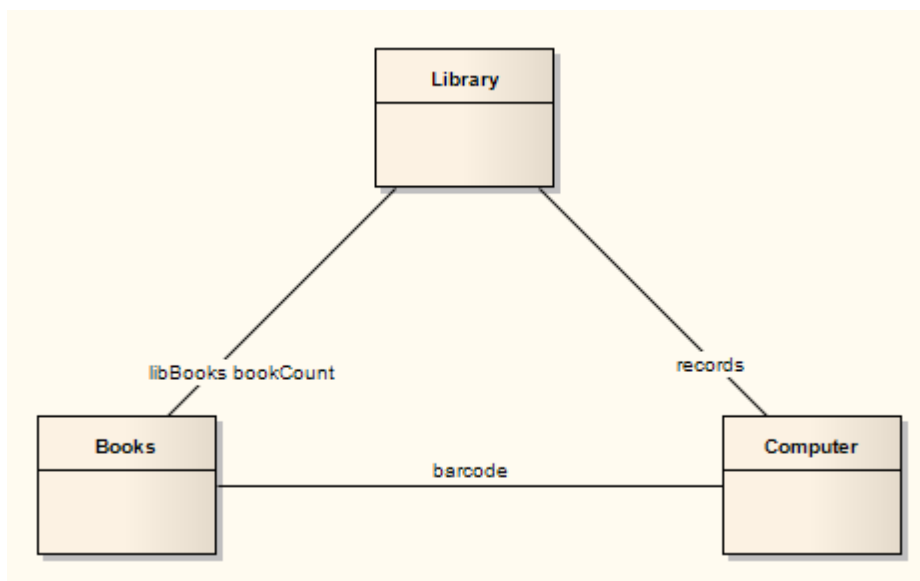
To indicate a property that is **not** owned by composition to the containing classifier, use a box symbol with a dashed outline, indicating association; to do this:

1. Right-click on the Part and select the **Properties** context menu option.
2. Select the Advanced page of the Properties dialog.
3. Set the **IsReference** option to **true**.

### Association Roles

Properties can also be reflected using a normal composite structure (**without** containing it in a Class), with the appropriate connectors, Parts and relationships indicated through connections to the Class.

The alternative representation is shown here; however, this representation fails to express the ownership immediately reflected by containing properties within a classifier.



### 6.2.5 Deployment Diagram















A **Deployment diagram** shows how and where the system is to be deployed; that is, its execution architecture.




Hardware devices, processors and software execution environments (system **Artifacts**) are reflected as **Nodes**, and the internal construction can be depicted by embedding or nesting Nodes. **Deployment** relationships indicate the deployment of Artifacts, and **Manifest** relationships reveal the physical implementation of **Components**. As Artifacts are allocated to Nodes to model the system's deployment, the allocation is guided by the use of **Deployment Specifications**.

**Example Diagram** [Example Deployment Diagram](#) 

#### Tools

Select Deployment diagram elements and connectors from the Deployment pages of the Toolbox.

Deployment Diagram Elements	Deployment Diagram Connectors
 Node	 Associate
 Device	 Communication Path
 Execution Environment	 Association Class
 Component	 Generalize
 Interface	 Realize
 Artifact	 Deployment
 Document Artifact	 Manifest

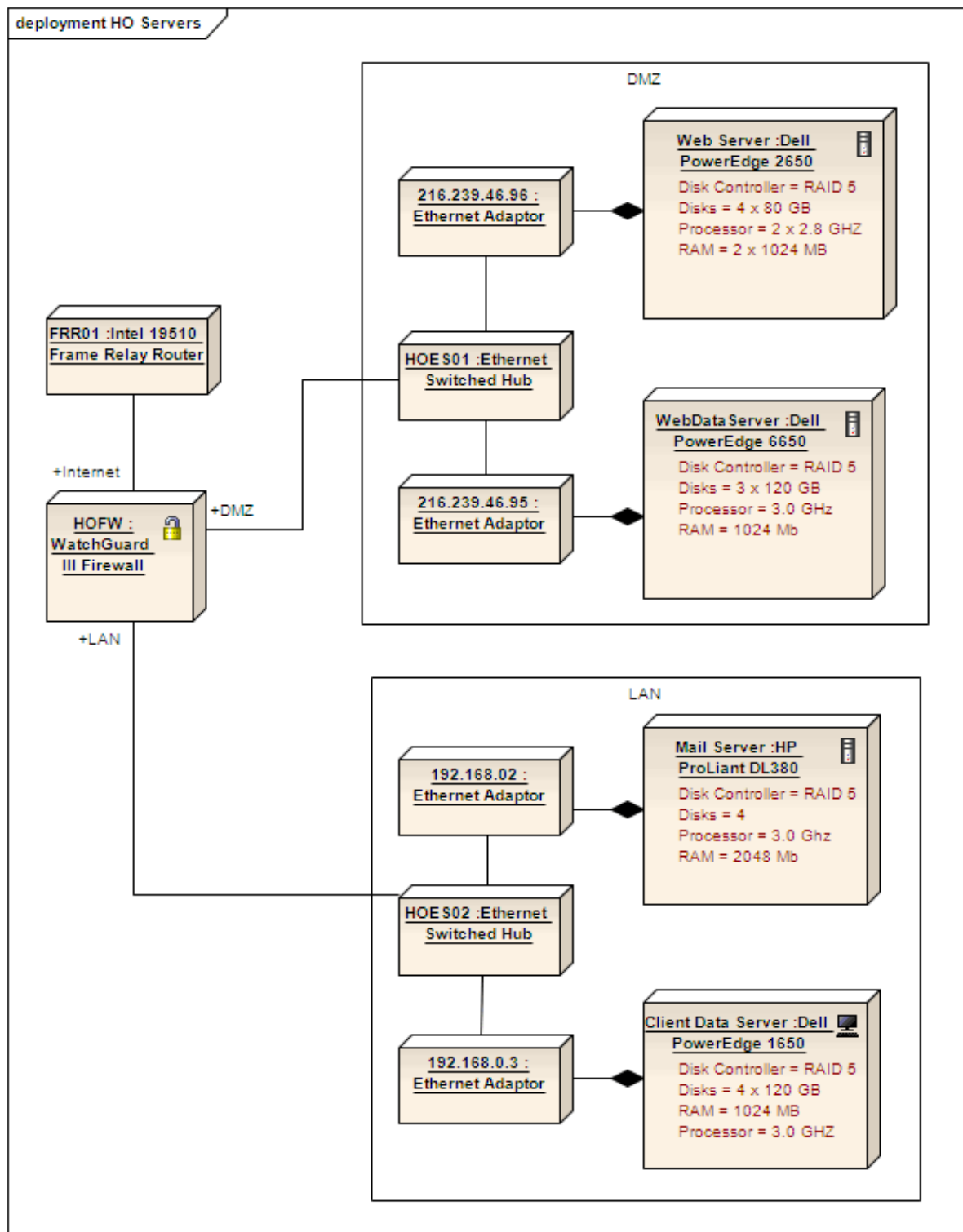
Deployment Diagram Elements	Deployment Diagram Connectors
 Deployment Specification	 Nesting
 Package	



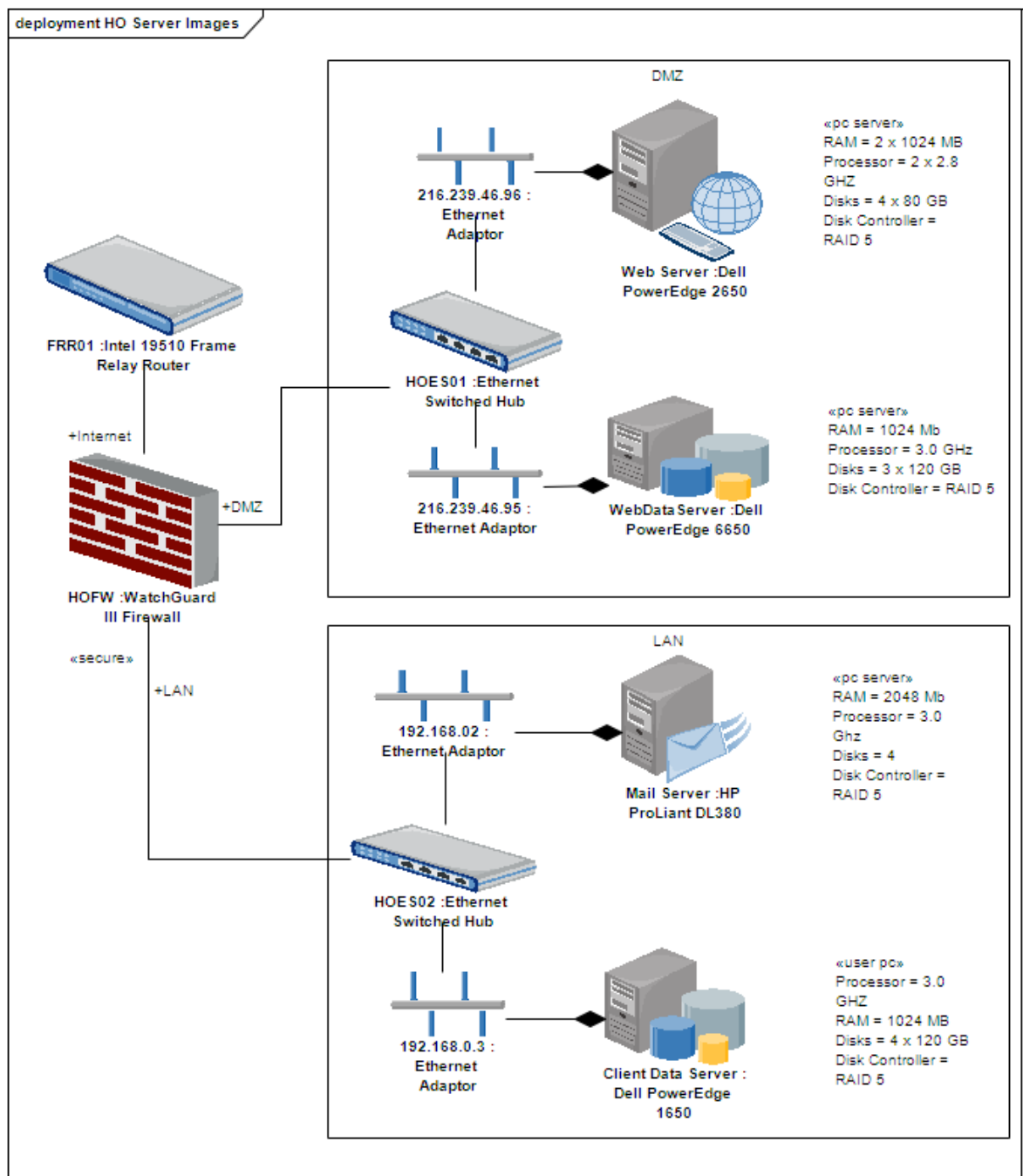
### 6.2.5.1 Example Deployment Diagram

This is a simple Deployment diagram, representing the arrangement of servers at a head office.

The servers are represented by Nodes linked by either simple or aggregate **Association** relationships.



Deployment diagrams are ideal for using alternative images for the objects that the elements represent. Such images can be substituted for the elements in the diagram, as shown below:



#### Learn more

- [Using the Image Manager](#) <sup>860</sup>

## 6.2.6 Component Diagram

A **Component diagram** illustrates the pieces of software, embedded controllers and such that make up a system, and their organization and dependencies.
















A Component diagram has a higher level of abstraction than a Class diagram; usually a component is

implemented by one or more Classes (or Objects) at runtime. They are building blocks, built up so that eventually a component can encompass a large portion of a system.

**Example Diagram** [Example Component Diagram](#) 

### Tools

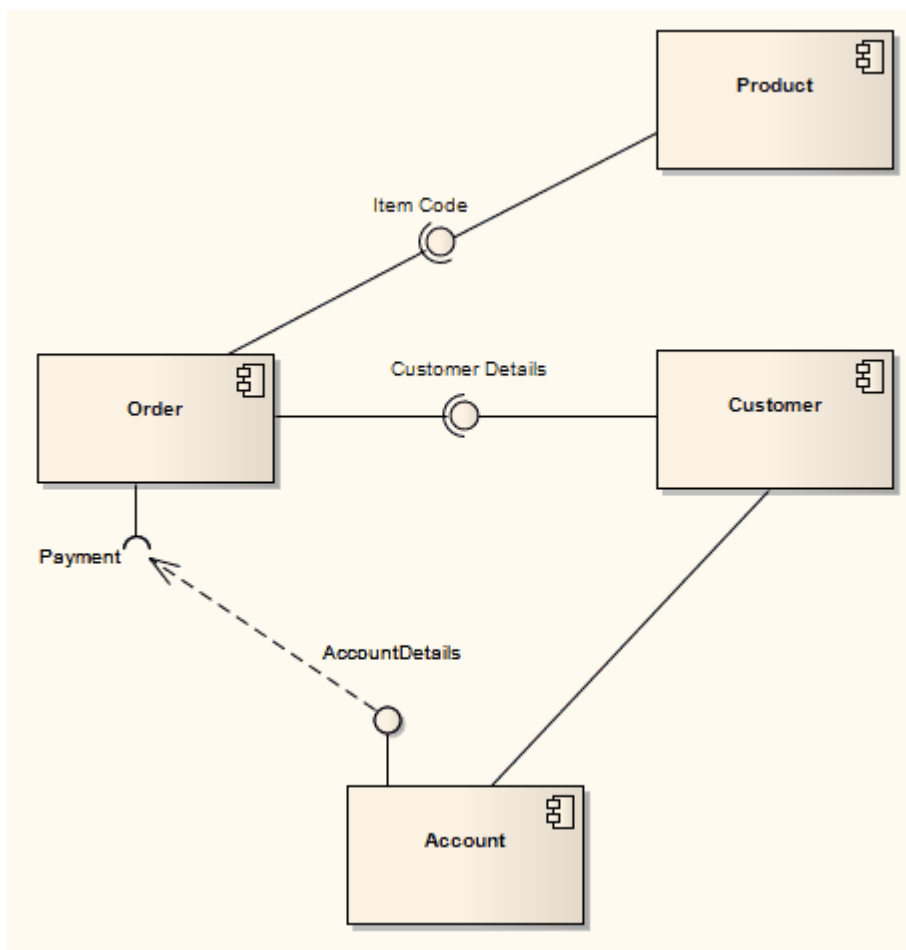
Select Component diagram elements and connectors from the Component pages of the Toolbox.

Component Diagram Elements	Component Diagram Connectors
 Package	 Assembly
 Packaging Component	 Delegate
 Component	 Associate
 Class	 Realize
 Interface	 Generalize
 Object	
 Port	
 Expose Interface	
 Artifact	
 Document Artifact	

#### 6.2.6.1 Example Component Diagram

This diagram demonstrates a number of components and their inter-relationships.

**Assembly** connectors connect the **provided interfaces** supplied by *Product* and *Customer* to the **required interfaces** specified by *Order*. A **Dependency** relationship maps a customer's associated *account details* to the required interface *Payment*, also specified by *Order*.



#### Learn more

- [Assembly](#)<sup>[1393]</sup>
- [Dependency](#)<sup>[1404]</sup>

### 6.2.7 Profile Diagram








A **Profile** diagram is any diagram created in a «*profile*» Package.

Profiles provide a means of extending the UML. They are based on additional stereotypes and Tagged Values that are applied to UML elements, connectors and their components. A Profile is a collection of such extensions that together describe some particular modeling problem and facilitate modeling constructs in that domain.

**Example Diagram** [Example Profile Diagram](#)<sup>[1197]</sup>

#### Tools

Select the following Profile diagram elements and connectors from the Profile pages of the Toolbox.

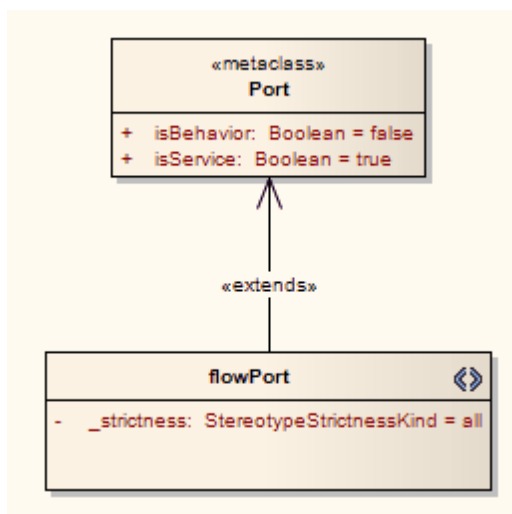
Profile Diagram Elements	Profile Diagram Connectors
 Profile	 Extension
 Stereotype	 Generalize
 Metaclass	 Tagged Value
 Enumeration	

#### Learn more

- [Developing Profiles](#)  1485
- [Create UML Profiles](#)  1485

### 6.2.7.1 Example Profile Diagram

A typical unit on a Profile diagram resembles this:



## 6.3 UML Behavioral Models

UML Behavioral diagrams depict the behavioral features of a system or business process. Behavioral diagrams include the following diagram types:

Diagram Type	Detail	See also
<b>Activity Diagrams</b>	Activity diagrams model the behaviors of a system, and the way in which these behaviors are related in an overall flow of the system.	<a href="#">Activity diagrams</a> <sup>[1199]</sup>
<b>Use Case Diagrams</b>	Use Case diagrams capture Use Cases and relationships among Actors and the system; they describes the functional requirements of the system, the manner in which external operators interact at the system boundary, and the response of the system.	<a href="#">Use Case diagrams</a> <sup>[1201]</sup>
<b>State Machine Diagrams</b>	State Machine diagrams illustrate how an element can move between states, classifying its behavior according to transition triggers and constraining guards.	<a href="#">State Machine diagrams</a> <sup>[1203]</sup>
<b>Timing Diagrams</b>	Timing diagrams define the behavior of different objects within a time-scale, providing a visual representation of objects changing state and interacting over time.	<a href="#">Timing diagrams</a> <sup>[1225]</sup>
<b>Sequence Diagrams</b>	Sequence diagrams are structured representations of behavior as a series of sequential steps over time. They are used to depict work flow, message passing and how elements in general cooperate over time to achieve a result.	<a href="#">Sequence diagrams</a> <sup>[1249]</sup>
<b>Communication Diagrams</b>	Communication diagrams show the interactions between elements at run-time, visualizing inter-object relationships.	<a href="#">Communication diagrams</a> <sup>[1259]</sup>
<b>Interaction Overview Diagrams</b>	Interaction Overview diagrams visualize the cooperation between interaction diagrams (Timing, Sequence, Communication and other Interaction Overview diagrams) to illustrate a control flow serving an encompassing purpose.	<a href="#">Interaction Overview diagrams</a> <sup>[1262]</sup>

### Learn more

- [Behavioral Modeling](#) <sup>[1014]</sup>
- [Code Generation from Behavioral Models](#) <sup>[2121]</sup>

### 6.3.1 Activity Diagram





















Activity diagrams are used to model system behaviors, and the way in which these behaviors are related in an overall flow of the system (that is, dynamic element interactions). The logical paths a process follows, based on various conditions, concurrent processing, data access, interruptions and other logical path distinctions, are all used to construct a process, system or procedure.



**Example Diagram** [Example Activity Diagram](#) 

#### Tools

Select Activity diagram elements and connectors from the Activity pages of the Toolbox.

Click on the following elements and connectors for more information.

Activity Diagram Elements	Activity Diagram Connectors
 Activity	 Control Flow
 Structured Activity	 Object Flow
 Action	 Interrupt Flow
 Partition	
 Object	
 Central Buffer Node	
 Datastore	
 Decision	
 Merge	
 Send	
 Receive	
 Synch	
 Initial	
 Final	
 Flow Final	
 Region	
 Exception	

Activity Diagram Elements	Activity Diagram Connectors
 Fork/Join	
 Fork/Join	

#### Notes

- You can create Analysis diagrams (Simplified Activity diagrams) containing the elements most useful for business process modeling, using the New Diagram dialog

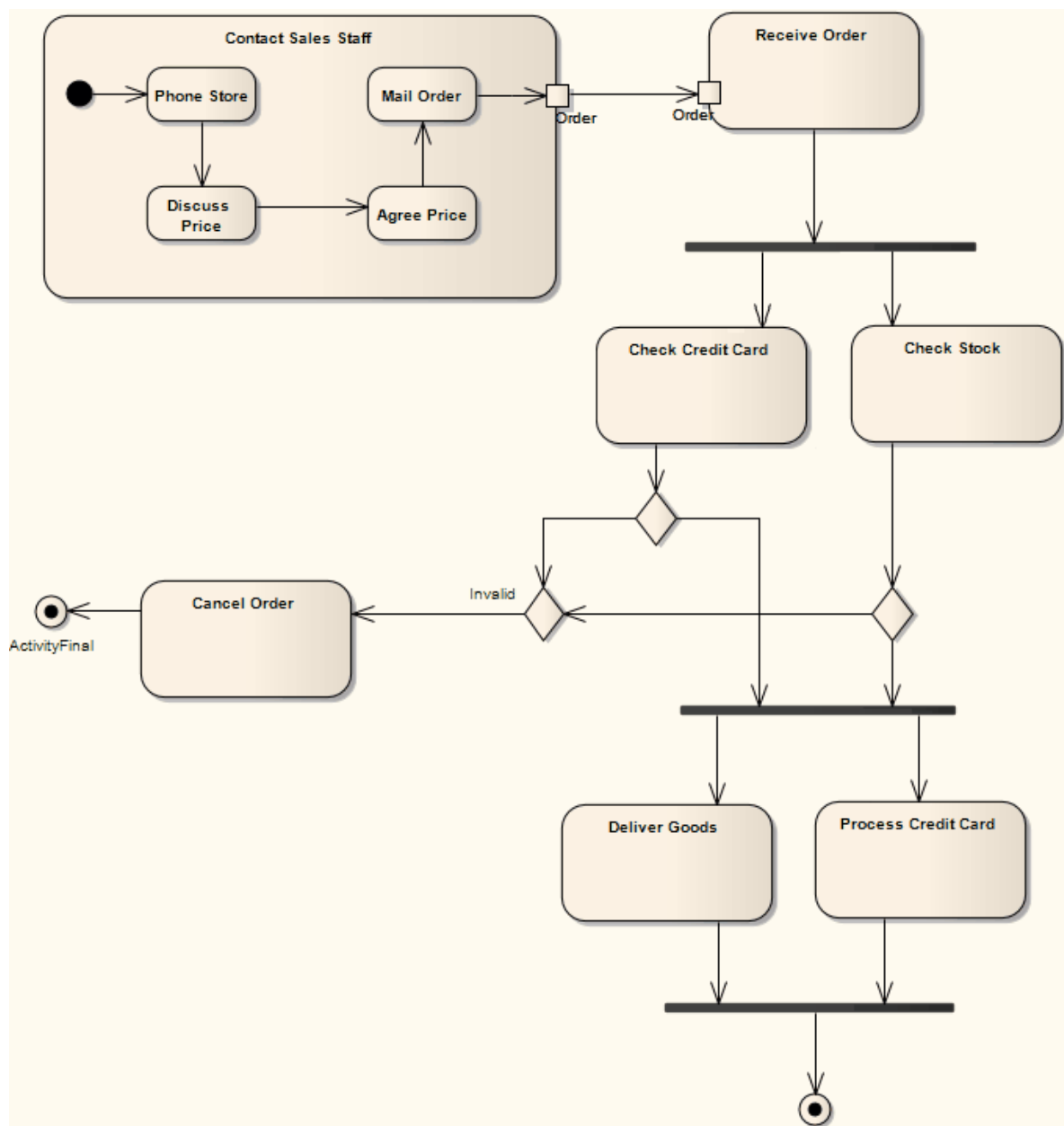
#### Learn more

- [Analysis Diagram](#)<sup>[1801]</sup>
- [New Diagram](#)<sup>[822]</sup>
- [Activity Toolbox](#)<sup>[808]</sup>

#### 6.3.1.1 Example Activity Diagram

The following diagram illustrates some of the features of Activity diagrams, including Activities, Actions, Start Nodes, End Nodes and Decision points.





### 6.3.2 Use Case Diagram

Use Case diagrams capture Use Cases and the relationships between Actors and the subject (system). You can use them to:
















- Describe the functional requirements of the system
- Describe the manner in which outside things (Actors) interact at the system boundary
- Describe the response of the system.

**Example Diagram** [Example Use Case Diagram](#) <sup>1203</sup>

#### Tools

Select Use Case diagram elements and connectors from the Use Case pages of the Toolbox. Click on the

following elements and connectors for more information.

Use Case Diagram Elements	Use Case Diagram Connectors
 Actor	 Use
 Use Case	 Associate
 Test Case	 Generalize
 Collaboration	 Include
 Collaboration Use	 Extend
 Boundary	 Realize
 Package	 Invokes
	 Precedes

### Notes

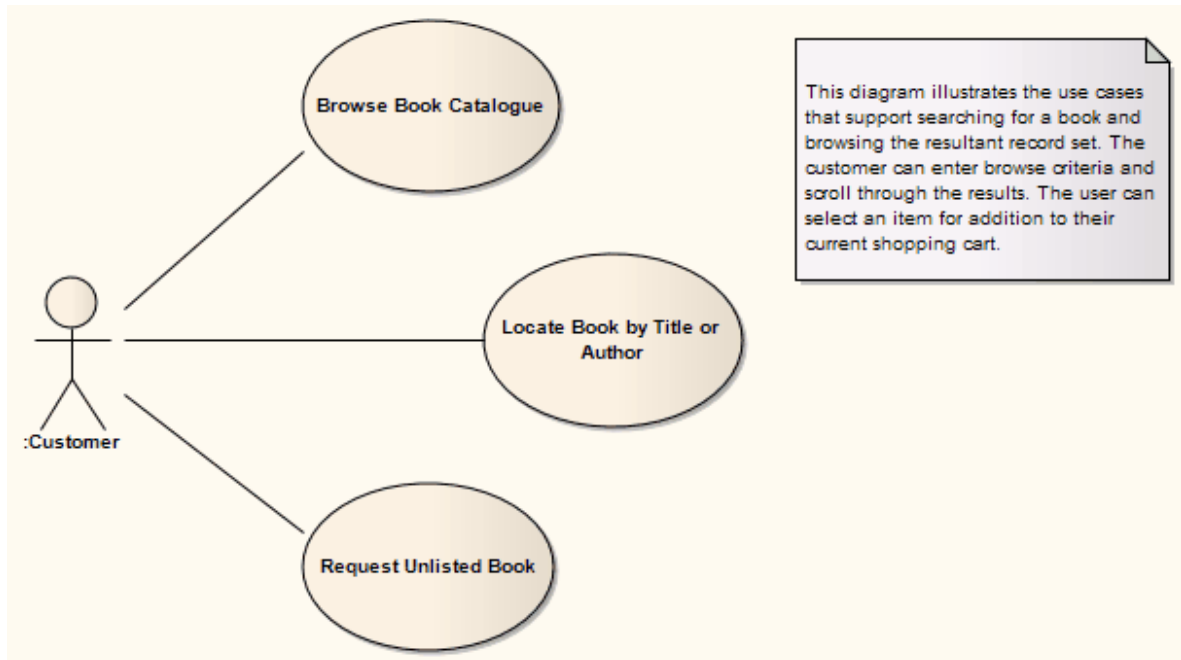
- Invokes and Precedes relationships are defined by the Open Modeling Language (OML); they are stereotyped Dependency relationships
- Invokes indicates that Use Case A, at some point, causes Use Case B to happen
- Precedes indicates that Use Case C must complete before Use Case D can begin

### Learn more

- [Use Case Extension Points](#) <sup>[1354]</sup>
- [Use Rectangle Notation](#) <sup>[1355]</sup>
- [Business Use Case](#) <sup>[1805]</sup>
- [Use Case pages](#) <sup>[801]</sup>

### 6.3.2.1 Example Use Case Diagram

The following diagram illustrates some features of Use Case diagrams:



### 6.3.3 State Machines

State Machines illustrate how an element (often a Class) can move between states, classifying its behavior according to transition triggers and constraining guards.

#### Naming

- State Machines were formerly known as State diagrams
- State Machine representations in UML are based on the Harel State Chart Notation and therefore are sometimes referred to as State Charts

#### State Tables

You can display a State Machine as a diagram, or as a table in one of three relationship formats.

#### Select the display format

















Step	Action	See also
1	Right-click on the diagram background and select the <b>Statechart Editor</b> context menu option.	
2	Select the appropriate display option: <ul style="list-style-type: none"> <li>• <b>Diagram</b></li> </ul>	

Step	Action	See also
	<ul style="list-style-type: none"> <li>• <b>Table (State-Next State)</b></li> <li>• <b>Table (State-Trigger)</b></li> <li>• <b>Table (Trigger-State)</b></li> </ul>	<a href="#">State Machine Table</a> <sup>[1212]</sup>

**Example Diagram** [Example State Machine Diagram](#) <sup>[1205]</sup>

### Toolbox Icons

Select State Machine elements and connectors from the State pages of the Toolbox; click on the element and connector icons below for more information.

State Machine Diagram Elements	State Machine Diagram Connectors
 <b>State</b>	 <b>Transition</b>
 <b>State Machine</b>	 <b>Object Flow</b>
 <b>Initial</b>	
 <b>Final</b>	
 <b>History</b>	
 <b>Synch</b>	
 <b>Object</b>	
 <b>Choice</b>	
 <b>Junction</b>	
 <b>Entry</b>	
 <b>Exit</b>	
 <b>Terminate</b>	
 <b>Fork/Join</b>	
 <b>Fork/Join</b>	

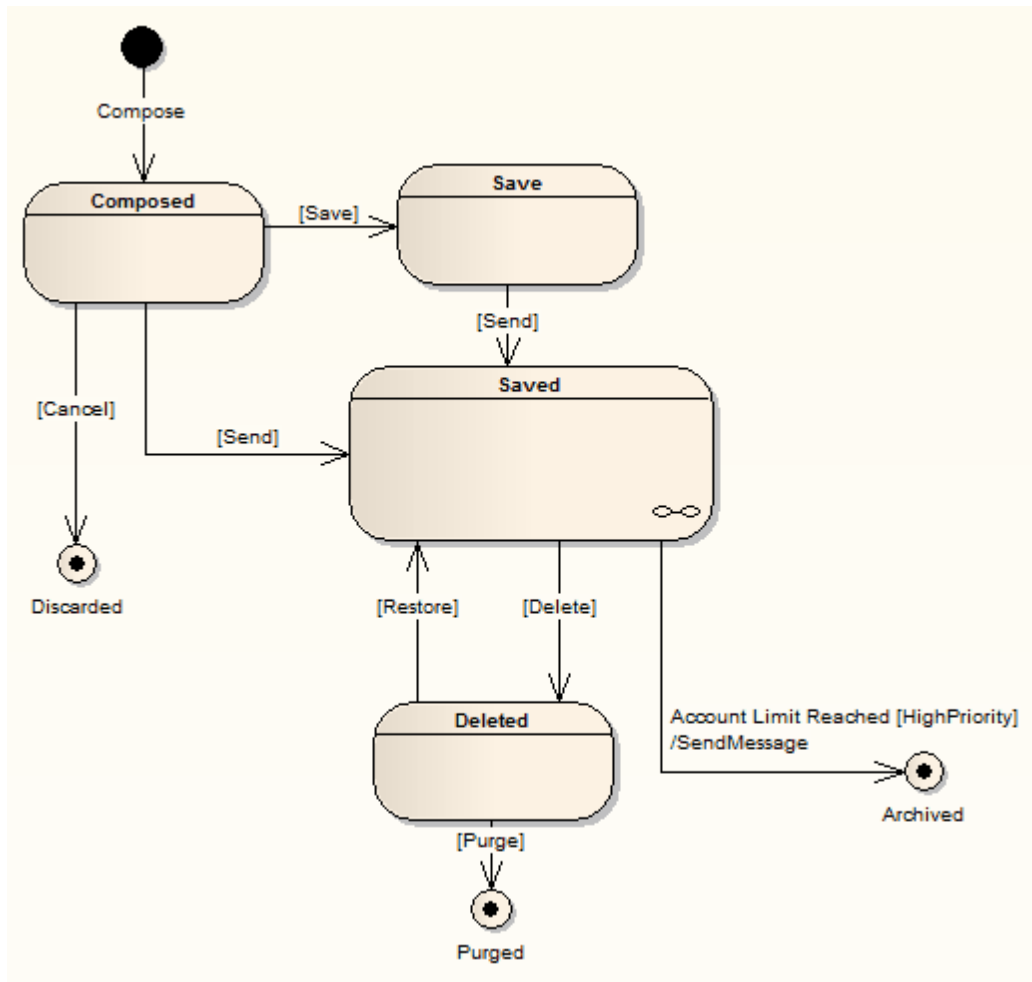
### Learn more

- [Legacy State Machine Templates](#) <sup>[2126]</sup>
- [State Machine Modeling for HDLs](#) <sup>[2131]</sup>

- [State Toolbox](#)<sup>[80]</sup>

### 6.3.3.1 Example State Machine

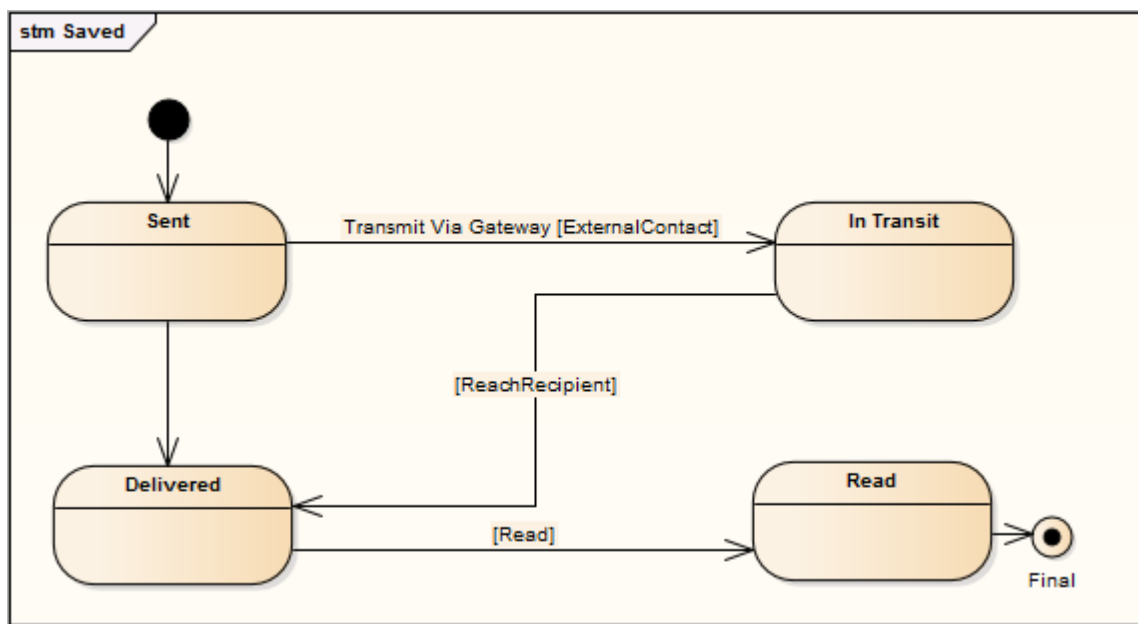
This diagram illustrates some features of State Machines:



#### Composite Diagram States

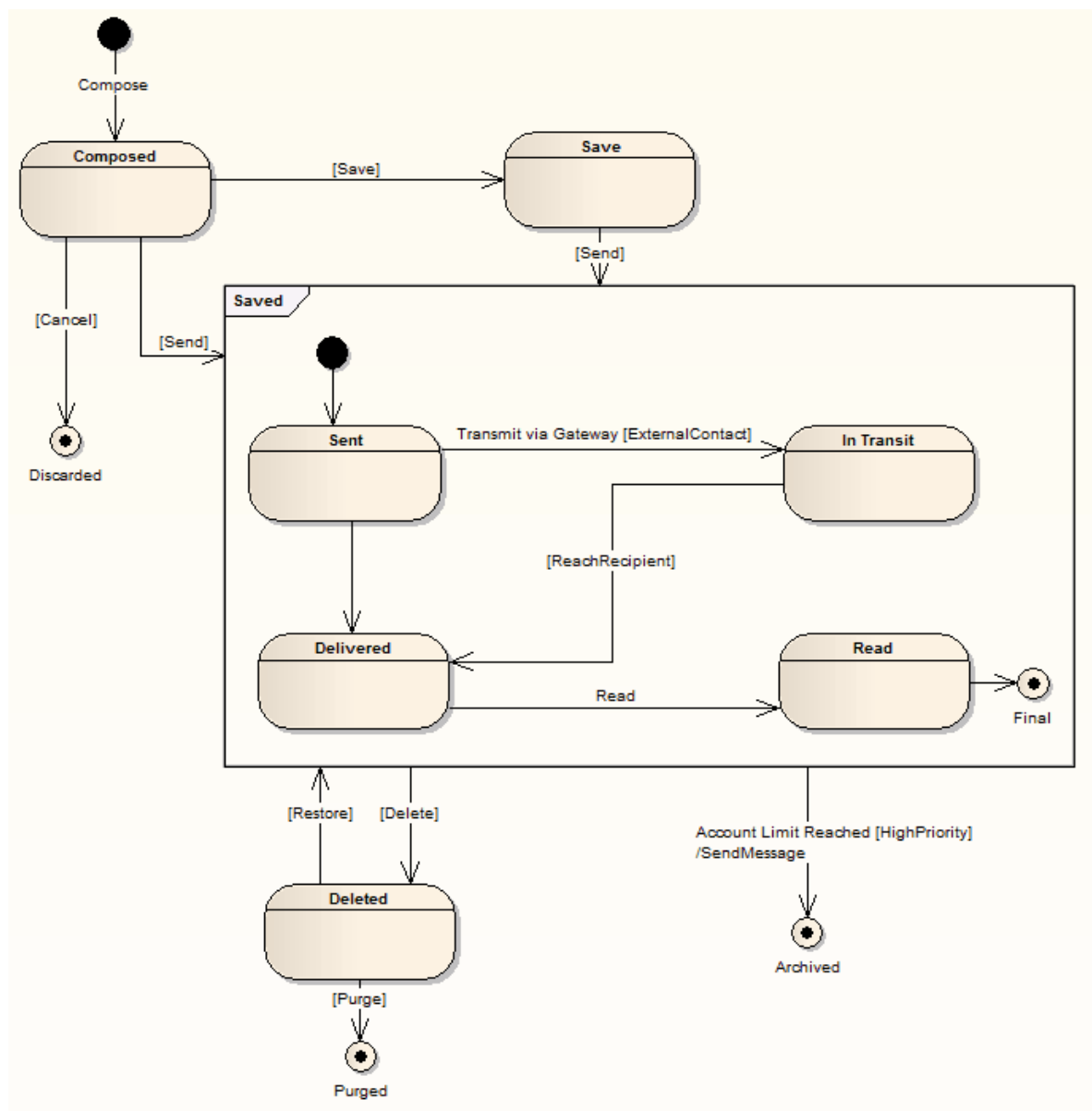
The chain-link symbol in the bottom right corner of the **Saved** State indicates that it is a State with a Composite diagram.

You have two options for displaying the contents of a State's Composite diagram. Firstly, you can **double-click on the parent element** to display its child diagram separately, as shown below:



By default, the child diagram displays within a labeled frame that represents the parent object in the context of the child diagram. You can right-click on the background and select the **Hide Diagram Frame** option to hide the frame, and on the **Show Diagram Frame** option to show the frame again.









Alternatively, you can right-click on the composite element on the main diagram and select the **Advanced | Show Composite Diagram** context menu option, which again displays the child diagram in a labeled frame, but this time within the context of the parent diagram.



### Notes

- State elements can display either with or without a line across them; the line - as shown above - displays when the element has features such as attributes (which could be hidden) or when the **Show State Compartment** checkbox is selected in the Objects page of the Options dialog
- It is possible to add **Entry Point** and **Exit Point** elements to the border of a State or State Machine element - right-click on the element and select the **New Element | Entry Point** or **Exit Point** option; if the element is a composite element and represented by a frame, you can also right-click on the selected **frame** and add the Entry Point or Exit Point elements
- If you have Entry Point and/or Exit Points on a State Machine that is a classifier for another State, you can create **ConnectionPointReferences** to the classifier from the other State
- It is also possible to add **Regions** to a State element or State Machine element **frame**; right-click on the selected frame and select the **Define Concurrent Substates** option























**Learn more**

- [Object Display Options](#)  <sup>63</sup>
- [Composite States](#)  <sup>1330</sup>
- [Pseudo-states](#)  <sup>1208</sup>
- [State Elements](#)  <sup>1329</sup>
- [Regions](#)  <sup>1209</sup>
- [Entry Point](#)  <sup>1300</sup>
- [Exit Point](#)  <sup>1304</sup>
- [Create a Connection Point Reference](#)  <sup>1210</sup>

**6.3.3.2 Pseudo-States**

Pseudo-states are a UML abstraction for various types of transient vertices used in State Machine diagrams. Pseudo-states are used to express complex transition paths.

The following types of pseudo-state are available:

Type	See
 Initial	<a href="#">Initial</a>  <sup>1313</sup>
 Entry	<a href="#">Entry Point</a>  <sup>1300</sup>
 Exit	<a href="#">Exit Point</a>  <sup>1304</sup>
 Choice	<a href="#">Choice</a>  <sup>1286</sup>
 Junction	<a href="#">Junction</a>  <sup>1319</sup>
 History	<a href="#">History</a>  <sup>1311</sup>
 Terminate	<a href="#">Terminate</a>  <sup>1350</sup>
 Final	<a href="#">Final</a>  <sup>1305</sup>
 Fork/Join	 Fork/Join <a href="#">Fork</a>  <sup>1309</sup>
 Fork/Join	 Fork/Join <a href="#">Join</a>  <sup>1310</sup>

**Notes**

- All the listed types of pseudo state can be represented in code, and can generate code under the State Machine code generation templates from Enterprise Architect release 11 onwards

**Learn more**

- [State Machines](#)  <sup>1203</sup>



### 6.3.3.3 Regions

If you are modeling an active State configuration on a State Machine diagram, and you need to represent several States as being active concurrently, you can achieve this by firstly creating a State Machine element or Composite State element and secondly subdividing that element with Regions. You set out the State configuration such that there is only ever one of the concurrently active States per Region. Multiple transitions can occur from a single event dispatch, so long as the similarly-triggered transitions are divided by Regions.

Regions display on an element on a diagram as subdivisions of a structured compartment, underneath other compartments such as tags, responsibilities, attributes and operations.

**Access**   **Right-click on element | Advanced | Define Concurrent Substates**

#### Create a Region in a Composite State or State Machine element

Step	Action
1	On the State Regions dialog, the <b>Name</b> field defaults to <b>&lt;anonymous&gt;</b> .
2	If you want to create Regions that have no title, simply click on the <b>Save</b> button once for each Region to create.  If you want to create named Regions, type the name and click on the <b>Save</b> button for each Region.
3	When you have created as many Regions as you need, click on the <b>Close</b> button.  You can now populate the Regions with elements from the State Diagram Toolbox.

#### Notes

- Changes to the elements in a Region are committed when the diagram is saved; if you want to undo the changes, reload the diagram without saving
- Any States, State Nodes (Pseudo-States) or Synch elements added to a Region are owned by that Region and, ordinarily, cannot be dragged into another Region; however, if you attempt to drag a **State** between Regions, the **move embedded element to region** menu option displays which - if you select it - allows the transfer to complete

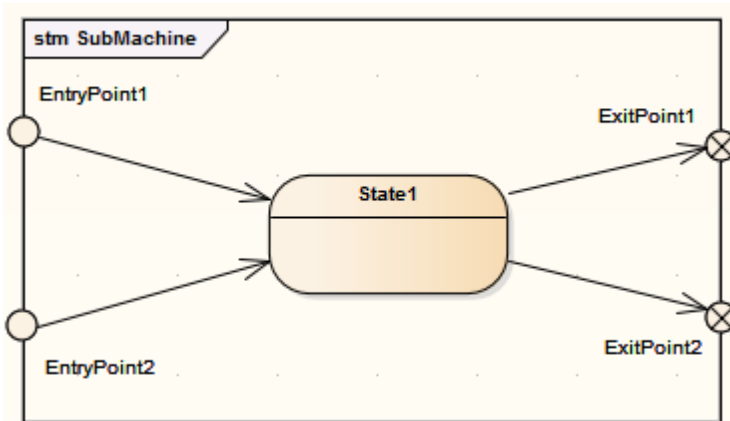
#### Learn more

- [State](#)<sup>[1329]</sup>
- [Composite State](#)<sup>[1330]</sup>
- [Pseudo-States](#)<sup>[1208]</sup>
- [State Machines](#)<sup>[1203]</sup> (diagrams)
- [State Machine](#)<sup>[1338]</sup> (element type)

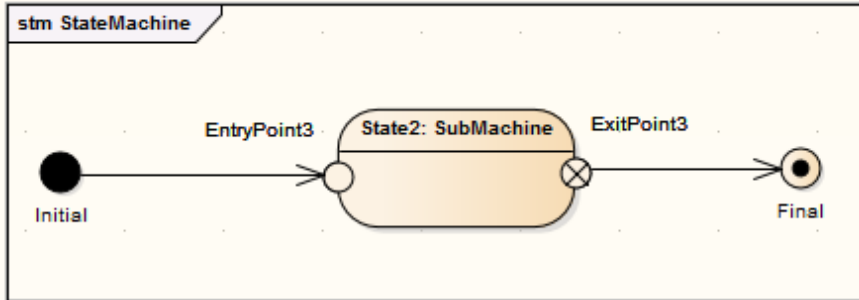
### 6.3.3.4 Create a Connection Point Reference

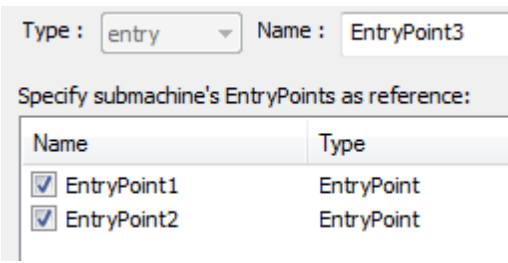
A Connection Point Reference represents the use, by a Submachine State, of an Entry Point or Exit Point pseudostate defined in the State element's classifier State Machine. You initially create the Connection Point Reference elements themselves as Entry Points or Exit Points.

#### Create Entry Points and/or Exit Points

Step	Action	See also
1	<p>Create or open the classifier State Machine (as a child diagram of a Class element).</p> <p>The State Machine is represented by a labeled frame.</p>	<a href="#">State Machine</a> <sup>[1338]</sup> <a href="#">Example State Machine</a> <sup>[1205]</sup>
2	<p>If the Entry Points and/or Exit Points do not already exist, right-click on the inside edge of the frame and select the <b>New Element   Entry Point</b> or <b>New Element   Exit Point</b> option, as necessary.</p> <p>The corresponding pseudostate element is immediately created on the edge of the frame. If you prefer, you can double-click on the element and give it a specific name.</p>	<a href="#">Entry Point</a> <sup>[1300]</sup> <a href="#">Exit Point</a> <sup>[1304]</sup>
3	Create as many additional Entry Point and/or Exit Point elements as you need.	
4	<p>If the corresponding State element does not already exist, drag a <b>State</b> icon from the Diagram Toolbox into the frame.</p> <p>Create the appropriate connectors between the State element and the Entry Point and Exit Point elements.</p>	<a href="#">State</a> <sup>[1329]</sup>
	 <p>The diagram shows a rectangular frame labeled 'stm SubMachine'. Inside the frame, there is a central rounded rectangle labeled 'State1'. On the left side of the frame, there are two small circles labeled 'EntryPoint1' (top) and 'EntryPoint2' (bottom). On the right side, there are two small circles with an 'X' inside, labeled 'ExitPoint1' (top) and 'ExitPoint2' (bottom). Arrows point from 'EntryPoint1' and 'EntryPoint2' to 'State1'. Arrows point from 'State1' to 'ExitPoint1' and 'ExitPoint2'.</p>	
5	Save the diagram.	

**Create Connection Point References**

Step	Action	See also
1	Create or open the calling State Machine (as a child diagram of a Class element).	<a href="#">State Machine</a> <sup>[1338]</sup>
2	If the elements do not already exist, create the appropriate State and pseudostate elements and connectors in the diagram.	<a href="#">Example State Machine</a> <sup>[1205]</sup>
3	Click on the calling State element and press <b>Ctrl+L</b> to display the Select Element dialog.  Browse for and select the classifier State Machine from the <i>Create Entry Points and/or Exit Points</i> stage.	<a href="#">Select &lt;item&gt; Dialog</a> <sup>[994]</sup>
4	Right-click on the State element, and select the <b>New Element   Entry Point</b> or <b>New Element   Exit Point</b> option, as you need.  The corresponding pseudostate element is immediately created on the border of the element.   <p>The diagram shows a state machine with an initial state (solid black circle) labeled 'Initial'. A transition arrow points from 'Initial' to an entry point (open circle) labeled 'EntryPoint3'. This entry point leads into a submachine state (rounded rectangle) labeled 'State2: SubMachine'. An exit point (circle with an 'X') labeled 'ExitPoint3' is on the right side of the submachine state. A transition arrow points from 'ExitPoint3' to a final state (bullseye circle) labeled 'Final'.</p>	<a href="#">Entry Point</a> <sup>[1300]</sup> <a href="#">Exit Point</a> <sup>[1304]</sup>
5	Double-click on the Entry Point element.  The Edit ConnectionPointReference dialog displays.	
6	If you prefer, in the <b>Name</b> field type a new name for the selected Entry Point.  In the <b>Specify submachine's EntryPoints as reference</b> panel, select the check box against <b>each</b> of the classifier's Entry Points you want to create a reference to. You can select more than one checkbox.	

Step	Action	See also
		
7	Click on the <b>OK</b> button.	
8	If necessary, repeat steps 4 to 7 for the State element's Exit Point.	

### 6.3.4 State Machine Table

A State Machine table is one of two variants of a State Machine (the other is the State Machine diagram). It displays the information of the State Machine in table form, and is a method of specifying the discrete behavior of a finite state-transition system; that is, what state the State Machine moves to and the conditions under which the transition takes place.

You can display the state transition in the table as one of two different relationships:

Display Type	Description	See also
<b>State - Trigger</b>	<p>The rows indicate the current states and the columns indicate trigger events.</p> <p>The cell at the intersection of a row and column identifies the target state in the transition if the trigger occurs, and the condition (or guard) of the transition, or the other way around if you prefer, in a <i>Trigger - State</i> format.</p>	<a href="#">Example State-Trigger Table</a> <sup>[1222]</sup>
<b>State - Next State</b>	<p>The rows and columns both indicate states, and the cell at the intersection of a row and column indicates:</p> <ul style="list-style-type: none"> <li>• The event that triggers a transition from the current (row) state to the next (column) state</li> <li>• The condition (or guard) of the event, and</li> <li>• The effect of the transition</li> </ul>	<a href="#">Example State-Next State Table</a> <sup>[1223]</sup>

#### Select the display format

Step	Action
1	Right-click on the diagram background and select the <b>Statechart Editor</b> context menu option.

2	Select the appropriate display option: <ul style="list-style-type: none"> <li>• <b>Diagram</b></li> <li>• <b>Table (State-Next State)</b></li> <li>• <b>Table (State-Trigger)</b></li> <li>• <b>Table (Trigger-State)</b></li> </ul>
---	--

#### Learn more

- [State Machines](#) <sup>[1203]</sup>
- [State Machine Table Options](#) <sup>[1213]</sup>
- [State Machine Table Operations](#) <sup>[1215]</sup>
- [State Machine Table Simulation](#) <sup>[1224]</sup>

### 6.3.4.1 State Machine Table Options

You can choose the State Machine table layout and set other options from the State Machine Diagram: Options dialog, which you display by either:

- Double-clicking on the State Machine table background or
- Right-clicking on the background and selecting the **State Table Options** context menu option

Field	Usage	See also
<b>Table Format</b>	Select the required table format: <ul style="list-style-type: none"> <li>• <b>State - Trigger:</b> <ul style="list-style-type: none"> <li>• Rows represent States, each State name in a left edge cell</li> <li>• Columns represent Triggers, each Trigger name in a column header cell</li> <li>• The intersection of a row and column identifies the Transition (if there is one)</li> <li>• The Transition cell displays information about the next State and the condition (guard) of the Transition</li> </ul> </li> <li>• <b>Trigger - State:</b> as above, except that rows represent Triggers and columns represent States</li> <li>• <b>State - Next State:</b> <ul style="list-style-type: none"> <li>• Both rows and columns represent States</li> <li>• The intersection of row and column defines the transition (if there is one) from the row State to the column State</li> </ul> </li> </ul>	
<b>Cell Size</b>		
<b>Transition Cell Width</b>	Specify the width of the transition cells (that is, the column width).	
<b>Transition Cell</b>	Specify the height of the transition cells (that is, the row height).	

Field	Usage	See also
<b>Height</b>		
<b>Left Edge Cell Width</b>	Specify the width of the left edge (row title) cells.	
<b>Top Edge Cell Height</b>	Specify the height of the top edge (column title) cells.	
<b>Cell Color</b>		
<b>State/Trigger Cell</b>	Select the color of the row and column title cells.	
<b>State/Trigger Enumeration</b>	Select the color of the enumeration (row/column numbering) cells. You must select at least one of the <b>Enable State Enumeration</b> and <b>Enable Event Enumeration</b> checkboxes to set this color.	
<b>Transition Cell</b>	Select the color of the transition cells (in the main body of the table).	
<b>Highlight Options</b>		
<b>Highlight Zones Related to Selected Transition</b>	Highlight the cells for all elements involved in a selected transition - the initial state, the target state, and the trigger.	
<b>Highlight Color</b>	Select the color of the highlight.	
<b>Use Different Color for Target State</b>	Highlight the cell for the target element in a transition in a different color to the cell for the source element.	
<b>Target Zone Color</b>	Select the color of the highlight.	
<b>Display Options</b>		
<b>Always Display an Empty State Zone</b>	Add an empty row (and, on a <i>State - Next State</i> table, an empty column) to the end of the table.  The title cell contains an ellipsis (...). You can click twice (not double-click) on the ellipsis to edit it and identify a new state. In this case, another empty state zone is automatically added.	
<b>Enable State Enumeration</b>	Add a cell to each state title cell, to number the state. Numbering starts at 0.	

Field	Usage	See also
<b>Prefix</b>	If required, type a prefix for the state number or delete the default <b>S</b> to have no prefix.	
<b>Enable Event Enumeration</b>	Add a cell to each event or trigger title cell, to number the event. Numbering starts at <b>0</b> .	
<b>Prefix</b>	If required, type a prefix for the event number or delete the default <b>E</b> to have no prefix.	
<b>Sample State Table</b>	Display a preview of the table format as you define it.	
<b>Advanced</b>	Define diagram options. The State Machine Diagram Properties dialog displays.	<a href="#">Set Appearance Options</a> [823]
<b>Restore Defaults</b>	Reapply the State Table diagram default values.	
<b>Apply</b>	Apply the changed options to the State Table diagram.	

### 6.3.4.2 State Machine Table Operations

As a State Machine table is a variant of a State Machine diagram, most of the operations for manipulating the data are the same as for State Machine diagrams. The operations specific to State Machine tables are described in the following topics:

Operation	See also
Change State Machine Table Position	<a href="#">Change Position of State Machine Table</a> [1216]
Change State Machine Table Size	<a href="#">Change Size of State Machine Table</a> [1216]
Insert New State	<a href="#">Insert New State (and Substate)</a> [1217]
Insert Trigger	<a href="#">Insert Trigger</a> [1218]
Insert/Change Transition	<a href="#">Insert/Change Transition</a> [1218]
Reposition State or Trigger Cells	<a href="#">Reposition State/Trigger Cells</a> [1219]

Add Legend	<a href="#">Add Legend</a> <sup>[1219]</sup>
Locate Cell in State Machine Diagram	<a href="#">Find Cell in State Machine Diagram</a> <sup>[1220]</sup>
State Machine Table Conventions	<a href="#">State Machine Table Conventions</a> <sup>[1221]</sup>
Export State Table To CSV File	<a href="#">Export State Table To CSV File</a> <sup>[1222]</sup>

#### Learn more

- [State Machines](#) <sup>[1203]</sup>
- [State Machine Table](#) <sup>[1212]</sup>

#### **6.3.4.2.1 Change State Machine Table Position**

If necessary, you can move the State Machine table around in the Diagram View.

##### **Change the position of the State Machine table**

Step	Action
<b>1</b>	Press ( <b>Ctrl+A</b> ) or double click on the top left cell to select the whole State Machine table.
<b>2</b>	Drag and drop the State Machine table to the required position. Alternatively, use ( <b>Shift</b> ) + ( <b>←</b> ) , ( <b>↑</b> ) , ( <b>→</b> ) or ( <b>↓</b> ) to move the State Machine table.

#### **6.3.4.2.2 Change State Machine Table Size**

There are three ways to change the size of the State Machine table:

Action	See also
Change the cell size on the State Machine Diagram: Options dialog.	<a href="#">State Machine Table Options</a> <sup>[1213]</sup>
Press ( <b>Ctrl+A</b> ) or double click on the top left cell to select the whole State Machine table, then press ( <b>Ctrl</b> ) + ( <b>←</b> ) , ( <b>→</b> ) , ( <b>↑</b> ) or ( <b>↓</b> ) to change the size.	
Select the State Machine table, then drag the shape handles to change the size.	



### 6.3.4.2.3 Insert New State

#### Topics

Action	Description
<b>Insert a new State in the State Machine table</b>	<p>You can insert a new State in the State Machine table, using one of following methods:</p> <ul style="list-style-type: none"> <li>• In the top left cell in the State Machine table, move the cursor to the word <b>State</b> to display a + at the end of the word; click on the + to create a new State</li> <li>• Right-click in the top left cell in the State Machine table and select the <b>Add State</b> context menu option</li> <li>• Right-click on an existing State cell in the State Machine table and select the: <ul style="list-style-type: none"> <li>• <b>Insert New State Before</b> context menu option to insert a new State before the current State, or</li> <li>• <b>Insert New State After</b> context menu option to insert a new State after the current State</li> </ul> </li> <li>• Click on an existing State cell in the State Machine table, and press ( <b>Insert</b> ) to create and insert a new State above the selected State</li> <li>• In the Toolbox, on the State Elements page, click on an element and then click on: <ul style="list-style-type: none"> <li>• The diagram background to add a new State to the end of the table, or</li> <li>• An existing State cell to add the new State just above it</li> </ul> </li> </ul> <p>From the State Elements page of the Toolbox you can insert State, Initial, Final, Entry, Exit and Terminate elements.</p>
<b>Add a Substate to a selected State</b>	<p>To add a Substate to a selected State, right-click on the required State cell in the State Machine table, and select the <b>Add Substate</b> context menu option; Enterprise Architect adds the Substate to the State.</p> <p>If the selected State does not allow a Substate, the <b>Add Substate</b> menu option is grayed out.</p> <p>You can also drag one existing State over another; if the second State allows Substates, the dragged State then becomes its Substate.</p> <p>Similarly, you can change the parent State of a Substate by dragging the Substate from the original parent State to a different State.</p>
<b>Remove the parent relation of a Substate and make it a separate State</b>	<p>To remove the parent relation of a Substate and make it a separate State, right-click on the Substate in the State Machine table and select the <b>Remove Parent Relation</b> context menu option; the Substate cell becomes a State cell.</p> <p>You can also drag and drop the Substate onto the top left cell of the State Machine table; the dragged Substate again becomes a State cell.</p>


#### 6.3.4.2.4 Insert Trigger

If the State Machine table format is either *State-Trigger* or *Trigger-State*, you can use **any** of the following methods to insert a new Trigger:

Method	Action
1	In the top left cell in the State Machine table, move the cursor to the word <b>Event</b> to display a <b>+</b> at the end of the word; click on the <b>+</b> to create a new Trigger.
2	In the top left cell in the State Machine table, right-click and select the <b>Add Trigger</b> context menu option to create a new Trigger.
3	Select an existing Trigger in the State Machine table, then press ( <b>Insert</b> ) to insert a new Trigger before the existing Trigger.
4	Click on an existing Trigger in the State Machine table, right-click and select either the: <ul style="list-style-type: none"> <li>• <b>Insert New Trigger Before</b> context menu option to insert a new Trigger before the current Trigger, or</li> <li>• <b>Insert New Trigger After</b> context menu option to insert a new Trigger after the current Trigger</li> </ul>

#### 6.3.4.2.5 Insert/Change Transition

Action	Description	See also
<b>Insert a new Transition</b>	<p>You can insert a new Transition using one of the following methods:</p> <ul style="list-style-type: none"> <li>• Right-click on the cell in which to create a Transition: <ul style="list-style-type: none"> <li>• If the State Machine table format is <i>State-Trigger</i> or <i>Trigger-State</i>, the context menu lists the States you can choose as the target of the Transition; click on the required State name to create the Transition</li> <li>• If the State Machine table format is <i>State-Next State</i>, click on the <b>Insert Transition</b> context menu option to create the Transition</li> </ul> </li> <li>• In the State Relationships page of the Toolbox, select the Transition element, then click on the cell in the State Machine table in which to create the Transition; double-click on the Transition to define it in the Transition Properties dialog</li> </ul>	
<b>Change the Transition</b>	As for the State Chart diagram, to change the properties of a Transition double-click the Transition cell and edit the details on the Transition Properties dialog.	<a href="#">State Machine Diagrams</a> <sup>[1203]</sup>
<b>Change Transition</b>	You can change the source and target of the Transition by right-clicking the Transition and selecting the <b>Advanced   Set Source and</b>	

<b>States</b>	<p><b>Target</b> context menu option.</p> <p>Alternatively, you can change the Transition source, target or Trigger by clicking on the Transition and dragging it to a different cell.</p> <p>If the State Machine table format is either <i>State-Trigger</i> or <i>Trigger-State</i>, you can change the target state of a Transition by:</p> <ol style="list-style-type: none"> <li>1. Highlighting the target state name in the Transition cell and clicking on it to display a list of the states in the table.</li> <li>2. Clicking on the preferred target state name.</li> </ol>	
<b>Highlight States and Trigger Related to Transition</b>	<p>You can select options to highlight the source State, target State and Trigger cells associated with a Transition, using the Highlight Options panel on the State Machine Diagram: Options dialog.</p> <p>When you click on the Transition cell its associated State and Trigger cells are highlighted.</p> <p>Alternatively, click on the Transition cell and press and hold ( <b>L</b> ).</p>	<a href="#">State Machine Diagram: Options</a> 

#### 6.3.4.2.6 Reposition State or Trigger Cells

You can change the position of a selected State or Trigger cell in one of the following ways:

- Right-click on the State or Trigger title cell and select the appropriate **Order | Move xxx** context menu option
- Click on the cell and press ( **Shift** ) + ( **→** ), ( **←** ), ( **↑** ) or ( **↓** )

#### 6.3.4.2.7 Add Legend

You can add a simple legend to any State Machine Table cell that has no transition. The two legend symbols are:

- **I** - Ignore
- **N** - Never Happen

#### Assign a legend symbol to a State Machine Table cell

Step	Action
<b>1</b>	<p>Click on the cell to which to assign the legend and press:</p> <ul style="list-style-type: none"> <li>• ( <b>I</b> ) to insert the Ignore legend, or</li> <li>• ( <b>N</b> ) to insert the Never Happen legend</li> </ul> <p>The required symbol displays in the center of the cell.</p>

#### Alternatively

Step	Action
1	Right-click on the cell to which to assign the legend.
2	<p>Select the appropriate context menu option:</p> <ul style="list-style-type: none"> <li>• <b>Legend   Ignore</b></li> <li>• <b>Legend   Never Happen</b></li> </ul> <p>The required symbol displays in the center of the cell.</p>

#### Notes

- To remove a legend symbol from a cell, either:
  - Click on the cell and press ( **Delete** ), or
  - Right-click on the cell and select the **Legend | Remove Legend** context menu option

#### 6.3.4.2.8 Find Cell in State Machine Diagram

#### Topics

Locate	Description
<b>In State Chart</b>	<p>On the State Machine table, to locate a selected State or Trigger element in a State Machine diagram:</p> <ul style="list-style-type: none"> <li>• Select the <b>Find   Locate in State Chart</b> context menu option</li> </ul> <p>Enterprise Architect switches to the State Machine diagram and highlights the selected element.</p> <p>You can locate a Transition relationship in a similar way, by selecting the <b>Locate in State Chart</b> context menu option.</p> <p>A Trigger on a State Machine table might or might not exist on the corresponding State Machine diagram; if the Trigger does not exist on the State Machine diagram, the <b>Locate in State Chart</b> option is disabled.</p>
<b>In State Table</b>	<p>Conversely, on the State Machine diagram, to locate a selected State or Trigger element in the corresponding State Machine table:</p> <ul style="list-style-type: none"> <li>• Select the <b>Find   Locate in State Table</b> context menu option</li> </ul> <p>Enterprise Architect switches to the State Machine table and highlights the selected element.</p> <p>You can locate a Transition relationship in a similar way, by selecting the <b>Locate in State Table</b> context menu option.</p>

#### 6.3.4.2.9 State Machine Table Conventions

##### Trigger

- Deleting a Trigger removes it completely from the model, therefore you cannot UNDO a Trigger deletion
- There is a <None> column at the end of the **Event** heading row; this is for Transitions that have no Trigger information

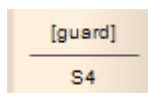
##### State

From the Toolbox you can insert the following *State* element types only (although the State Machine table might pick up and display other types, such as *Submachine State*):

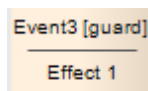
- State
- Initial
- Final
- Entry
- Exit
- Terminate

##### Transition

The Transition cell displays its properties in one of two ways, depending on the State Machine table format.; if the State Machine table format is *State - Trigger* or *Trigger - State*, the Transition cell displays the *Guard* and *Target* as shown below:



If the State Machine table format is *State - Next State*, then the Transition cell displays the *Trigger*, *Guard* and *Effect* as shown below:



In the State Machine table, you can edit the *Guard* and *Effect* in place. If the *Guard* or *Effect* is empty for your selected Transition cell, the cell displays an ellipsis ( ... ) instead; click twice (not double-click) on the ellipsis to type in the *Guard* and *Effect* names.

#### 6.3.4.2.10 Export State Table To CSV File

##### Export a State Machine Table to a CSV file

Step	Action
1	Open the required State Machine Table.
2	Right-click on the diagram background and select the <b>Export Statechart to CSV file</b> context menu option. The Save As browser dialog displays.
3	Select the appropriate directory location and type in the .CSV filename.
4	Click on the <b>Save</b> button.

#### 6.3.4.3 Example State-Trigger Table

##### State - Trigger:

The rows indicate the current states and the columns indicate trigger events (or the other way around if you prefer, in a **Trigger - State** format).

The cell at the intersection of a row and column identifies the target state in the transition if the trigger occurs, and the condition (or guard) of the transition.

State \ Trigger		Event1	Event2	Event3	Event4	<None>
		E0	E1	E2	E3	E4
Initial	S0					S1
State1	S1				S2	
State2	S2	S6	<u>[Guard]</u> S4			
	SubState1	S3	S4			
	SubState2	S4		<u>[Cond]</u> S2		
	SubState3	S5				
State3	S6					S7
Final	S7					

#### 6.3.4.4 Example State-Next State Table

##### State - Next State:

The rows and columns both indicate states, and the cell at the intersection of a row and column indicates:

- The event that triggers a transition from the current (row) state to the next (column) state
- The condition (or guard) of the event, and
- The effect of the transition

Next State State		Initial	State 1	State2				State3	Final
					SubState1	SubState2	SubState3		
		S0	S1	S2	S3	S4	S5	S6	S7
Initial		S0							
State1		S1		Event4					
State2		S2				Event2 [Guard]		Event1	
	SubState1	S3				Event2			
	SubState2	S4		Event3 [Cond]					
	SubState3	S5							
State3		S6							
Final		S7							

#### 6.3.4.5 State Machine Table Simulation

A State Machine Table is a representation of a State Machine, and can be simulated in exactly the same way as a State Machine diagram.

**Access** Select State Machine diagram in Project Browser and  
(If necessary) **Right-click diagram background | Statechart Editor | Table** (any of the three options) then  
**Analyzer | Simulator**

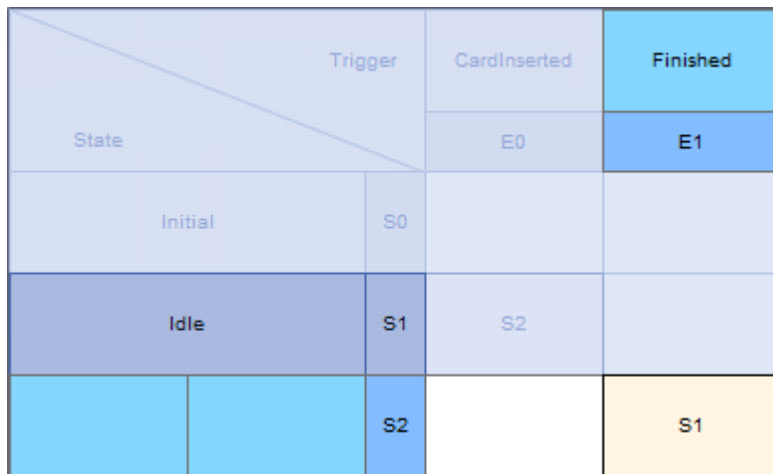
##### Highlight active cells

As the simulation executes, the table cells change color to indicate the:

- Currently active State(s) - the color set in the **Highlight Color** field of the State Machine Options: Dialog, and a dark border
- Potential next States(s) - A variant of the color in the **Highlight Color** field or, if the **Use Different Color for Target State** checkbox is selected on the State Machine Options: Dialog, the color set in the **Target Zone Color** field
- Active Transition(s) - the color set in the **Transition Cell** field of the State Machine Options: Dialog
- Trigger(s) - the color set in the **Highlight Color** field of the State Machine Options: Dialog
- Non-active States - gray



For example:



### Signal Triggers

As when running a simulation as a Diagram, the simulation will automatically traverse transitions with no guards or validated guards. Transitions with a Trigger will not be followed unless that Trigger has been fired. They may be fired automatically from the Simulation Events window or you can fire a Trigger manually by right-clicking on the Transition or Trigger cell and select the **Signal Trigger in Simulation** context menu option.

### Learn more

- [State Machine Table Options](#) <sup>1213</sup>
- [Model Simulation](#) <sup>2463</sup>

## 6.3.5 Timing Diagram

A Timing diagram defines the behavior of different objects within a time-scale. It provides a visual representation of objects changing state and interacting over time. You can use it to:





- Define hardware-driven or embedded software components; for example, those used in a fuel injection system or a microwave controller
- Specify time-driven business processes

**Example Diagram** [Example Timing Diagram](#) <sup>1227</sup>

### Tools

Select Timing diagram elements and connectors from the Timing pages of the Toolbox. Click on the elements and connectors below for more information.

Timing Diagram Elements	Timing Diagram Message
State Lifeline	Message

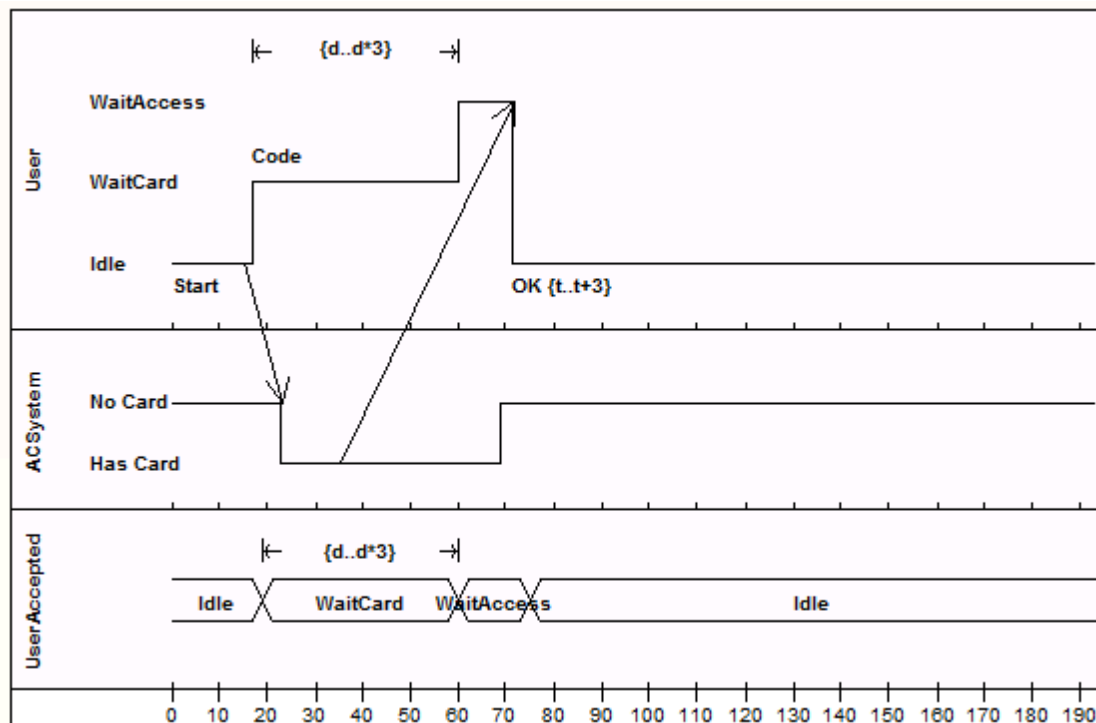
Timing Diagram Elements	Timing Diagram Message
 Value Lifeline	
 Message Label	
 Message Endpoint	
 Diagram Gate	

### Learn more

- [Create a Timing Diagram](#) <sup>f1227</sup>
- [Set a Time Range](#) <sup>f1228</sup>
- [State Lifeline](#) <sup>f1335</sup>
- [Value Lifeline](#) <sup>f1355</sup>
- [Edit a Timing Diagram](#) <sup>f1228</sup>
- [Time Intervals](#) <sup>f1241</sup>
- [Message \(Timing Diagram\)](#) <sup>f1431</sup>
- [Timing Toolbox](#) <sup>f806</sup>

### 6.3.5.1 Example Timing Diagram

An example of a Timing diagram is shown below:



### 6.3.5.2 Create a Timing Diagram

#### Create a Timing diagram

Step	Action
1	Right-click on a package in the Project Browser and select the <b>Add   Add Diagram</b> context menu option. The New Diagram dialog displays.
2	In the Select From panel, select <b>UML Behavioral</b> .
3	In the Diagram Types panel, select <b>Timing</b> .
4	Click on the <b>OK</b> button. The Diagram View displays, on which you create the Timing elements for the diagram.

[Learn more](#)

- [Set a Time Range](#)<sup>[1228]</sup>
- [Edit a Timing Diagram](#)<sup>[1228]</sup>

### 6.3.5.3 Set a Time Range

**Set a time range before adding Lifeline elements to your Timing diagram**

Step	Action
1	Right-click on the diagram and select the <b>Set Timeline Range</b> context menu option. The Set Timeline Range dialog displays.
2	In the <b>Start Time</b> and <b>End Time</b> fields, type the numeric values for the start and end points of the timeline; for example, set the range <b>0</b> to <b>100</b> . The start time must be less than the end time.
3	In the <b>Time Units</b> field, type the unit in which the time is measured; for example, seconds or minutes.
4	If it is not necessary to show the time range on the diagram, select the <b>Suppress In Diagram</b> checkbox.
5	Click on the <b>OK</b> button. If you have not suppressed it, the time range displays underneath the Lifeline elements that you create on the diagram.

### 6.3.5.4 Edit a Timing Diagram

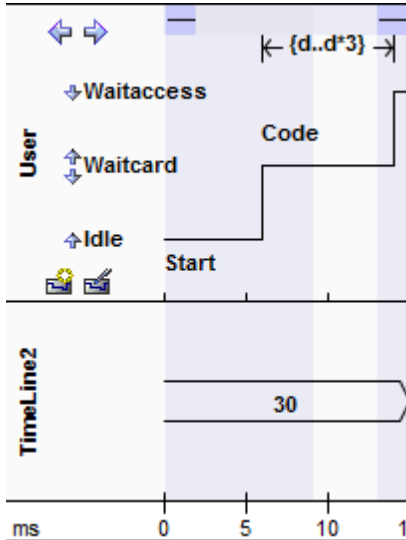
On a Timing Diagram, you can add State Lifeline elements and Value Lifeline elements. You can maintain the states and transitions on these Lifeline elements either on the diagram itself or via the Configure Timeline dialog.

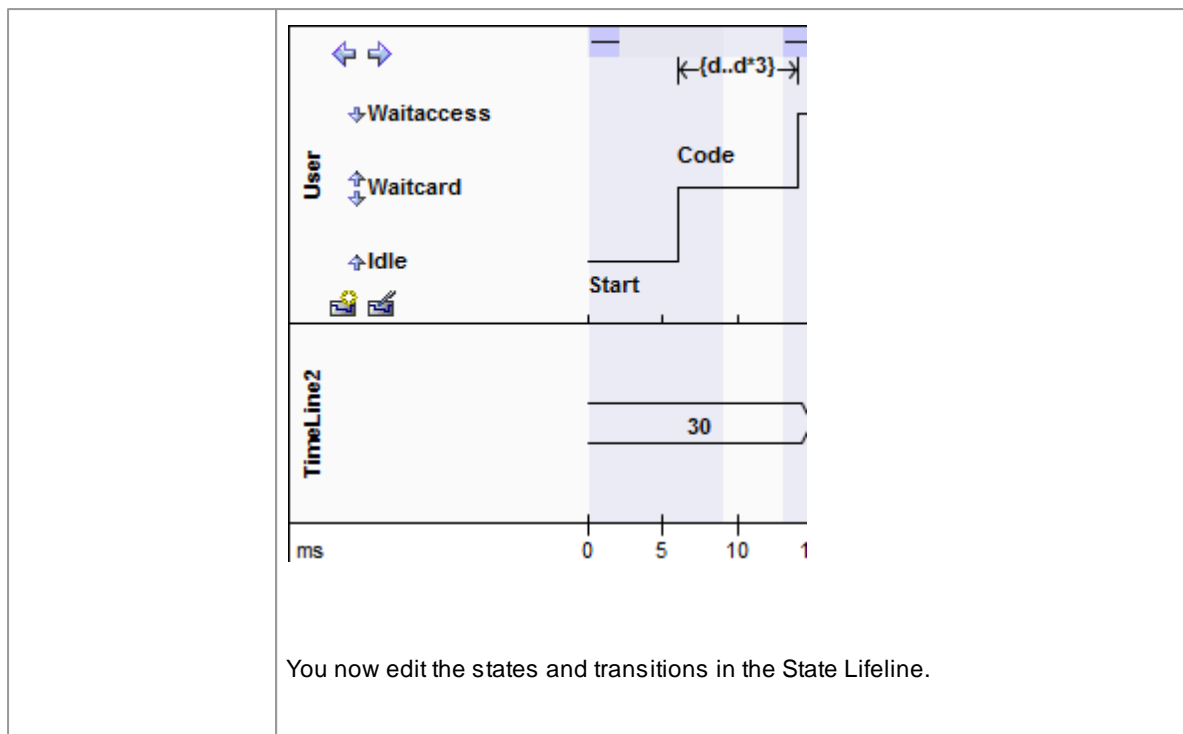
#### **Learn more**

- [Add and Edit a State Lifeline Element](#)<sup>[1229]</sup>
- [Add States to a State Lifeline](#)<sup>[1230]</sup>
- [Edit States in a State Lifeline](#)<sup>[1231]</sup>
- [Delete States in a State Lifeline](#)<sup>[1231]</sup>
- [Edit Transitions in a State Lifeline Element](#)<sup>[1232]</sup>
- [Add and Edit a Value Lifeline Element](#)<sup>[1234]</sup>
- [Add States in a Value Lifeline Element](#)<sup>[1235]</sup>
- [Edit Transitions in a Value Lifeline Element](#)<sup>[1235]</sup>
- [Configure Timeline dialog - States Tab](#)<sup>[1237]</sup>
- [Configure Timeline dialog - Transitions Tab](#)<sup>[1239]</sup>

#### 6.3.5.4.1 Add and Edit State Lifeline

From the Timing elements page of the Toolbox drag a State Lifeline icon onto your diagram. The element displays on the diagram.

Task	Action
<b>Define the name of the State Lifeline</b>	<ol style="list-style-type: none"> <li>1. Right-click on the element and select the <b>Other Properties</b> context menu option; the Timeline &lt;name&gt; dialog displays, showing the General tab.</li> <li>2. Overtyp the <b>Name</b> field.</li> <li>3. Click on the <b>Apply</b> button and the <b>OK</b> button.</li> </ol>
<b>Sizing and Scale</b>	<p>In the top left corner of a selected Lifeline element are the left and right quick sizing buttons (↔).</p> <p>These buttons increase or decrease the width of the Lifeline element, which in turn controls the scale width of each time unit; by increasing the width of the element you increase the resolution when adding transitions, which makes them easier to edit.</p> <p>In order to edit the State Lifeline element, you must click on it to select it.</p>
<b>Set Timeline Start Position</b>	<p>You might require more space at the start of your timelines; for example, to use long state names.</p> <p>To insert more space in all the timelines on a diagram</p> <ol style="list-style-type: none"> <li>1. Right-click on the diagram background and select the <b>Set Timeline Start Position</b> menu option; the Set Timeline Start Position dialog displays.</li> <li>2. The <b>Value 80 to 300</b> field defaults to <b>80</b> as the minimum distance in pixels between the start of the timeline element and the start of the timeline itself; type a new value up to 300 pixels and click on the <b>OK</b> button to increase the space at the start of the timeline. The two diagrams below have start positions of 80 pixels and 150 pixels respectively.</li> </ol> 








#### Learn more

- [Add States to a State Lifeline](#) <sup>1230</sup>
- [Edit States in a State Lifeline](#) <sup>1231</sup>
- [Delete States in a State Lifeline](#) <sup>1231</sup>
- [Edit Transitions In State Lifeline](#) <sup>1232</sup>

#### 6.3.5.4.2 Add States to a State Lifeline



##### Add States to a State Lifeline

Step	Description	See also
1	Click on the <i>State Lifeline</i> element.  The <b>New State</b> button (  ) and <b>Edit States</b> button (  ) display at the bottom left of the element.	
2	Click on the <b>New State</b> button.  The New State dialog displays.	
3	In the <b>State</b> field, type the name of the state.	
4	Click on the <b>OK</b> button.  You must add at least two states; for example, <b>On</b> and <b>Off</b> .	

5	<p>As you add states, increase the height of the element by dragging a handle-box (  ) on the edge of the element.</p> <p>You can also add states using the States tab of the Configure Timeline dialog.</p> <p>Add either:</p> <ul style="list-style-type: none"> <li>• Discrete states to the Timeline, or</li> <li>• A continuous range of numeric states</li> </ul>	<a href="#">Add a New State</a>  <a href="#">Numeric Range Generator</a> 
---	--	--

#### 6.3.5.4.3 Edit States in a State Lifeline

##### Edit States in a State Lifeline


Step	Description	See also
1	Click on the State Lifeline element and click on the required state. The Edit State dialog displays.	
2	In the <b>State</b> field, change the name as required.	
3	Click on the <b>OK</b> button.	
4	<p>If necessary, change the order of the states by either:</p> <ul style="list-style-type: none"> <li>• Clicking on the up or down arrows (  ) beside each state name, or</li> <li>• Right-clicking on the state name and selecting the <b>Move Up</b> or <b>Move Down</b> context menu options</li> </ul> <p>You can also edit the states using the States tab of the Configure Timeline dialog.</p>	<a href="#">States</a> 

#### 6.3.5.4.4 Delete States in a State Lifeline

##### Delete States in a State Lifeline

Step	Description
1	Right-click on the state name and select the <b>Delete</b> context menu option.




##### Alternatively

Step	Description
1	Click on the State Lifeline element.
2	Hold down ( <b>Ctrl</b> ) and move the cursor over the state name. The cursor changes form (  ).
3	Click the mouse button. The state name is deleted.

#### 6.3.5.4.5 Edit Transitions In State Lifeline

A Timing diagram allows you to show the transitions (changes of state) that occur within a State Machine over a fixed time period and at certain timing points. This is similar in many respects to an Interaction lifeline with State changes highlighted. As events and changes occur within the instance this Timing diagram represents, state changes occur and are mapped onto this Timeline. In that respect it is a record of how a particular aspect of the system behaves over time.

When building a Timeline it is necessary to define the States first - and then to add the explicit transitions between those states at particular timing points.

Task	Action	See also
<b>Add and Move Transitions</b>	After you have added states, you can add transitions between states directly on the timeline using the mouse.	<a href="#">Add and Move Transitions</a> <sup>[1233]</sup>
<b>Change the Transition Time</b>	Move the cursor over one or other of the vertical transition lines and drag the line left or right to change the time of the transition.  While on the line, the cursor shape changes to the horizontal movement cursor (  ).	
<b>Merge Transitions</b>	If necessary, you can 'push' a transition to merge it with the next or previous transition point on any Lifeline element on the diagram.  Position the cursor off the appropriate side of the transition line; the cursor changes form (  or  ).  Click the mouse button; the system locates the nearest transition in the required direction, on any element on the diagram, and merges the current transition with that transition.	
<b>Delete Transitions</b>	Transitions are automatically deleted when you move the transition to the same state as the previous transition state, and release the cursor.  Alternatively, right-click on the transition line and select the <b>Delete</b> context menu option.	










--	--	--

#### 6.3.5.4.5.1 Add and Move Transitions

After you have added states, you can configure state changes (transitions) directly on the Timeline using the mouse. This is a fast and effective means of building a detailed model of state changes over time.

In order to modify the Timeline, place the mouse over the existing Timeline. As you move the cursor over the Timeline, the cursor changes to one of three shapes, described below.

Task	Action	See also
<b>The move cursor</b> (  )	Displays when it is directly over the timeline.  Hold down the mouse button and drag the line to move the timeline to a state above or below the current position; you can move the transition more than one state up or down, if necessary.	
<b>The new transition up cursor</b> (  )	Displays when it is just below the timeline, and there is another state above the line.  Press and hold ( <b>Alt</b> ); the cursor changes (  ).  Click to create a new transition to the state above the line.  To push the transition up more than one state, move the cursor onto the line and drag it up.  The transition is for one interval unit; you can make it longer by changing the transition time.  If you do not hold ( <b>Alt</b> ), the cursor does not change and the whole timeline from the transition onwards moves up.	<a href="#">Edit Transitions In State Lifeline</a>  <sup>[1232]</sup>
<b>The new transition down cursor</b> (  )	Displays when it is just above the transition line, and there is another state below the line.  Press and hold ( <b>Alt</b> ); the cursor changes (  ).  Click to create a new transition to the state below the line.  To push the transition down more than one state, move the cursor onto the line and drag it down.  The transition is for one interval unit; you can make it longer by changing the transition time.  If you do not hold ( <b>Alt</b> ), the cursor does not change and the whole timeline from the transition onwards moves down.	<a href="#">Edit Transitions In State Lifeline</a>  <sup>[1232]</sup>

As you move the cursor over the vertical line of a transition, the time at which the transition occurs displays

next to the line.

**Access** Click directly on the appropriate transition line, after the transition begins, or Right-click on the transition line and select the **Edit** context menu option

Edit the transitions as required, on the Edit Transition dialog.

Field/ Option	Action
<b>At Time</b>	Type the point on the timescale at which the transition occurs.
<b>Transition To</b>	Type the name of the state to which the transition occurs.
<b>Event</b>	Type the name of the event that the transition represents. This displays on the Timeline element just above the transition line.
<b>Duration Constraint</b>	Type any constraint on the duration of the transition. This displays on the Timeline element, along the top of the element over the transition.
<b>Time Constraint</b>	Type any constraint on the start of the transition. This displays on the Timeline element at the start of the transition.
<b>OK</b>	Click on this button to save the changes.

#### Notes

- Once Event, Duration Constraint or Time Constraint are displayed on the diagram, you can edit them directly by clicking on them to display their specific dialog
- You can delete them by pressing and holding ( **Ctrl** ) as you click on them; the cursor changes form when you press ( **Ctrl** )
- You can also edit transitions using the Transitions tab of the Configure Timeline dialog

#### Learn more

- [Transitions](#) 



#### **6.3.5.4.6 Add and Edit Value Lifeline**

From the Toolbox drag a Value Lifeline element onto your diagram. The element displays on the diagram.

#### Edit the Value Lifeline name

Step	Action
1	Right-click on the element and select the <b>Other Properties</b> context menu option. The Timeline <name> dialog displays, showing the General tab.
2	Overtyping the <b>Name</b> field.
3	Click on the <b>Apply</b> button and the <b>OK</b> button.

### Sizing and Scale

In the top left corner of a selected Lifeline element are the left and right *quick sizing* buttons (   ). These buttons increase or decrease the width of the Lifeline element, which in turn controls the scale width of each time unit. By increasing the width of the element you increase the resolution when adding transitions, which makes them easier to edit.

### Learn more

- [Value Lifeline](#)<sup>[1355]</sup>
- [Add States in a Value Lifeline](#)<sup>[1235]</sup>
- [Edit Transitions in a Value Lifeline](#)<sup>[1235]</sup>

#### 6.3.5.4.7 Add States In Value Lifeline

Adding States to a Value Lifeline is similar to adding states to a State Lifeline element.

For a Value Lifeline, only the first state displays on the diagram. The other states are added to a list to access when creating transitions; they only display on the Lifeline element as you create transitions to those states.


You can only edit or delete states in a Value Lifeline element using the States tab of the Configure Timeline dialog.

### Learn more

- [Add States to a State Lifeline](#)<sup>[1230]</sup>

#### 6.3.5.4.8 Edit Transitions In Value Lifeline

##### Add Transitions to the states on a Value Lifeline element, via the diagram


Step	Action
1	Move the cursor above the transition line. The cursor changes form (  ).

2	Click the mouse button. The New Transition Event dialog displays.
3	In the <b>Transition To</b> field, click on the drop-down arrow and select a state from the list of available states; this displays on the Lifeline element within the transition box. The remaining fields on the dialog are optional.
4	In the <b>Event</b> field, type the name of the event that the transition represents; this displays on the Lifeline element just below and at the start of the transition line.
5	In the <b>Duration Constraint</b> field, type any constraint on the duration of the transition; this displays on the Lifeline element, along the top of the element over the transition.
6	In the <b>Time Constraint</b> field, type any constraint on the start of the transition. This displays on the Lifeline element at the start of the transition, just after the Event name.
7	Click on the <b>OK</b> button to create the new transition.


#### Edit a Transition

Step	Action
1	Click on the state name in the transition. Alternatively, right-click on the state name and select the <b>Edit</b> context menu option. The Edit Transition dialog displays, which is the same as the New Transition Event dialog, except that the <b>At Time</b> field is enabled.
2	If necessary, overwrite the <b>At Time</b> field to define a different start point. You cannot change the <b>At Time</b> field for the first state in the timeline; this is always <b>0</b> .
3	Edit the remaining fields as necessary.
4	Click on the <b>OK</b> button to save the changes.

#### Change the transition time


Step	Action
1	<p>To change the start or end time of a transition, click on the start or end point of the transition and drag it to the new position.</p> <p>While on the line, the cursor shape changes to the horizontal movement cursor (.</p>

### Delete Transitions

Step	Action
1	<p>To delete a transition, press and hold ( <b>Ctrl</b> ) and click on the transition state name.</p> <p>While you hold ( <b>Ctrl</b> ) on the transition state name, the cursor changes form ( .</p> <p>Alternatively, right-click on the state name and select the <b>Delete</b> context menu option.</p>

#### 6.3.5.4.9 Configure Timeline - States

You can manage states using the States tab of the Configure Timeline dialog. To display this, either:

- Double-click on the Lifeline element
- Right click on the Lifeline element and select the **Properties** context menu option, or
- On a Value Lifeline, click on the **Edit States** button (  )

The Configure Timeline dialog defaults to the States tab.

All states currently defined for the Lifeline element are listed in the States panel.

### Add a new State

Step	Action
1	In the <b>State Name</b> field, type the name of the first new state in the Lifeline element; for example, <b>WaitState</b> .
2	<p>Click on the <b>Save</b> button.</p> <p>The state is added to the States panel and (for a State Lifeline Element) to the diagram.</p>
3	Click on the <b>New</b> button.
4	In the <b>State Name</b> field, type the name of the next state in the Lifeline element.
5	Repeat steps 2 to 5 until you have added all required states (you must add at least three to the

	Lifeline element).
6	When you have added all the required states, click on the <b>OK</b> button to close the Configure Timeline dialog.



#### Edit an existing state

Step	Action
1	Click on the state in the <b>States:</b> list.
2	In the <b>State Name</b> field, change the name of the state.
3	Click on the <b>Save</b> button.

#### Delete an existing State

Step	Action
1	Click on the state in the <b>States:</b> list.
2	Click on the <b>Delete</b> button.

#### Change the order of States

Step	Action
1	Click on the state in the <b>States:</b> list.
2	Click on the  or  buttons to move the state up or down the sequence.

#### Learn more

- [Numeric Range Generator](#)<sup>[1239]</sup>

#### 6.3.5.4.10 Numeric Range Generator

You can also use the Configure Timeline dialog to create a range of states having numeric values to be applied to the Timeline.

**Important:** This operation deletes all existing states and transitions for the Timeline element.


##### Create a range of states having numeric values

Step	Action
1	Double-click on the Lifeline element. The Configure Timeline dialog displays.
2	Click on the <b>Create Continuous Numeric States</b> button. The Numeric Range Generator dialog displays.
3	In the <b>High Value</b> and <b>Low Value</b> fields, type the upper and lower values of the range.
4	In the <b>Step Value</b> field, type the increase interval. Nonsense values do not parse; <b>Low Value</b> must be less than <b>High Value</b> , and <b>Step Value</b> must be a positive value smaller than the total range.
5	In the <b>Units</b> field, type the name of the measurement unit; for example, <b>minutes</b> .
6	Click on the <b>OK</b> button. Enterprise Architect displays a warning that existing states and transitions are to be deleted.
7	Click on the <b>Yes</b> button. The Configure Timeline dialog redisplay, with the defined range of states listed in the States panel.
8	Click on the <b>OK</b> button. For a: <ul style="list-style-type: none"> <li>Value Lifeline, the first state is shown on the Timeline for the full time range of the Timeline</li> <li>State Lifeline, the range of states is displayed as the y-axis of the Timeline</li> </ul>

#### 6.3.5.4.11 Configure Timeline - Transitions

You can also manage transitions using the Transitions tab of the Configure Timeline dialog. To display this, either:

- Double-click on the Lifeline element
- Right click on the Lifeline element and select the **Properties** context menu option, or

- On a Value Lifeline, click on the **Edit States** button (  )

The Configure Timeline dialog defaults to the States tab. Click on the Transitions tab.

All transitions defined for the Timeline element are listed in the Transition Points panel.

#### Add a new transition

Step	Action
1	Click on the <b>New</b> button.
2	In the New Transition panel, type the details of the transition.
3	Click on the <b>Save</b> button.

#### Edit a transition

Step	Action
1	Click on a transition in the list.
2	In the Edit Transition panel, edit the fields for the transition as required.
3	Click on the <b>Save</b> button.

#### Delete a transition

Step	Action
1	Click on a transition in the list.
2	Click on the <b>Delete</b> button. The transition is removed from the dialog and the Lifeline.
3	Click on the <b>OK</b> button.



### 6.3.5.5 Time Intervals

You create and manage Time Intervals using the *Interval Bar* (the pale line along the top of each selected Lifeline element). Time Intervals enable you to perform various operations on transitions, such as copy and paste. They also enable you to compress sections of the timeline so that they are not visible.

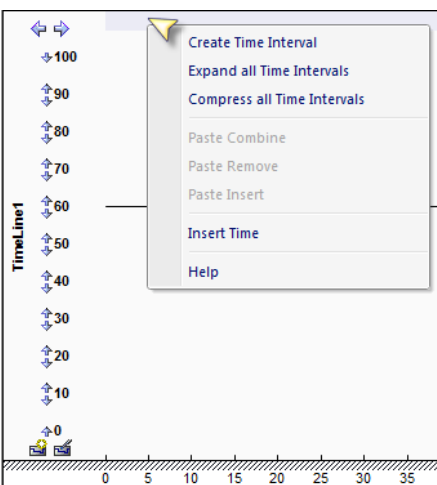
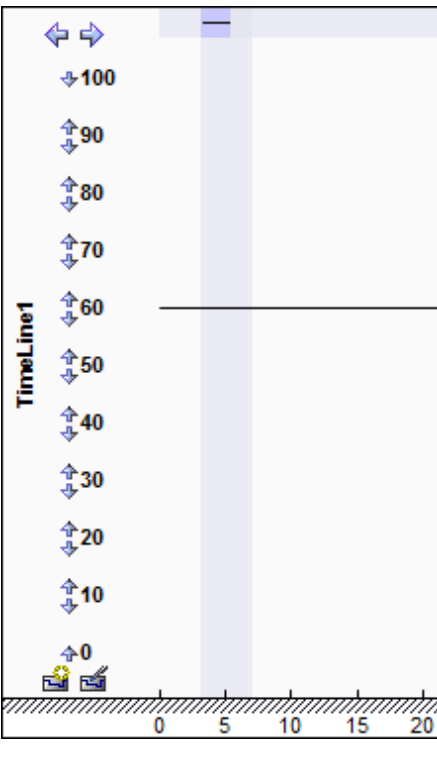
Each Time Interval displays across all Timeline elements down to the last element on the diagram.

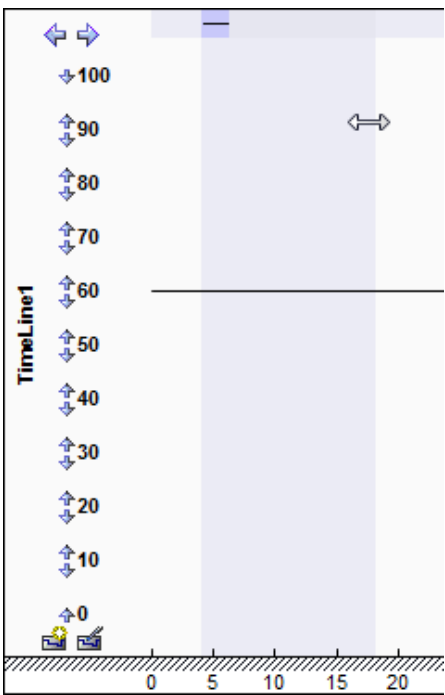
Action	Description	See also
<b>Create Time Intervals</b>	You can create a Time Interval using the: <ul style="list-style-type: none"> <li>Interval Bar - Context menu</li> <li>Interval Bar - ( <b>Shift</b> ) key, or</li> <li>Timeline - Context menu</li> </ul>	<a href="#">Create Time Intervals</a> <sup>[1241]</sup>
<b>Compress Time Intervals</b>	You can compress Time Intervals to conserve space on long timelines.	<a href="#">Compress Time Intervals</a> <sup>[1244]</sup>
<b>Select Time Intervals</b>	There are a number of ways to select Time Intervals for performing other operations.	<a href="#">Select Time Intervals</a> <sup>[1245]</sup>
<b>Move Time Intervals</b>	To move a Time Interval, move the cursor over the Interval bar within the Time Interval, hold down the mouse button and drag the interval left or right.  Time Intervals can meet, but cannot overlap.	
<b>Resize Time Intervals</b>	To resize a Time Interval, move the cursor over the Interval Bar at the start or end edge of the Time Interval, hold down the mouse button and move the edge left or right.  Time Intervals can meet, but cannot overlap.	
<b>Delete Time Intervals</b>	To delete Time Intervals, select each Time Interval to be deleted and press ( <b>Delete</b> ).  Deleting the Time Interval does not delete transitions within that interval.	

#### 6.3.5.5.1 Create Time Intervals

##### Create a Time Interval using the Interval Bar context menu

Images	Step	Action
--------	------	--------



	<p>1 Right-click on the Interval Bar at approximately the point at which to start or finish the Time Interval, and select the <b>Create Time Interval</b> context menu option.</p>
	<p>2 The Time Interval displays down all the timeline elements, as a narrow pale band with a blue compression box at the top.</p>



3

Move the cursor to the edge of the Time Interval in the Interval Bar so that the cursor changes to the drag form and drag the edge to the correct start or end point.

Create a Time Interval using the Interval Bar and ( Shift ) key

Step	Description	See also
1	Move the cursor over the Interval Bar and press ( Shift ). The cursor changes shape (  ).	
2	Click to create the Time Interval.	
3	Move the cursor to the edge of the Time Interval in the Interval Bar so that the cursor changes to the drag form (  ) and drag the edge to the correct start or end point.	

Create a Time Interval using the Timeline context menu

Step	Description	See also
1	Right-click on the timeline just after a transition. The context menu displays.	
2	Click on the <b>Select</b> menu option.	<a href="#">Compress Timeline</a> <small>1247</small>

Enterprise Architect creates a Time Interval covering the period from the selected transition up to the next transition.

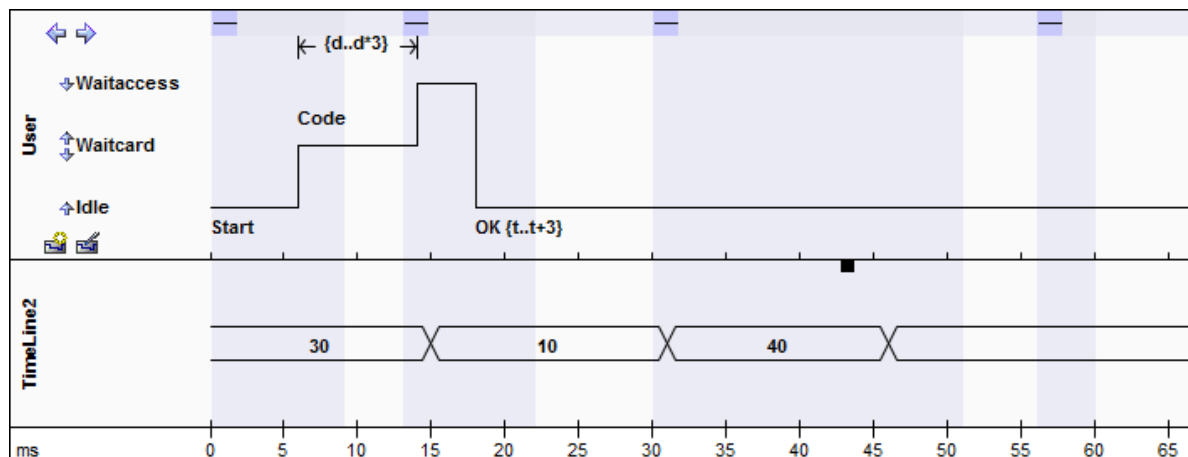
If there are other Time Intervals in this period, Enterprise Architect replaces them with the single Time Interval for the transition state; you should consider this when creating the Time Interval, as it extends across the other Timeline elements in the diagram.

A value of this method is that it creates a Time Interval for a period in which no transitions occur, which could be lengthy; you can then compress this Time Interval to hide the period of inactivity.

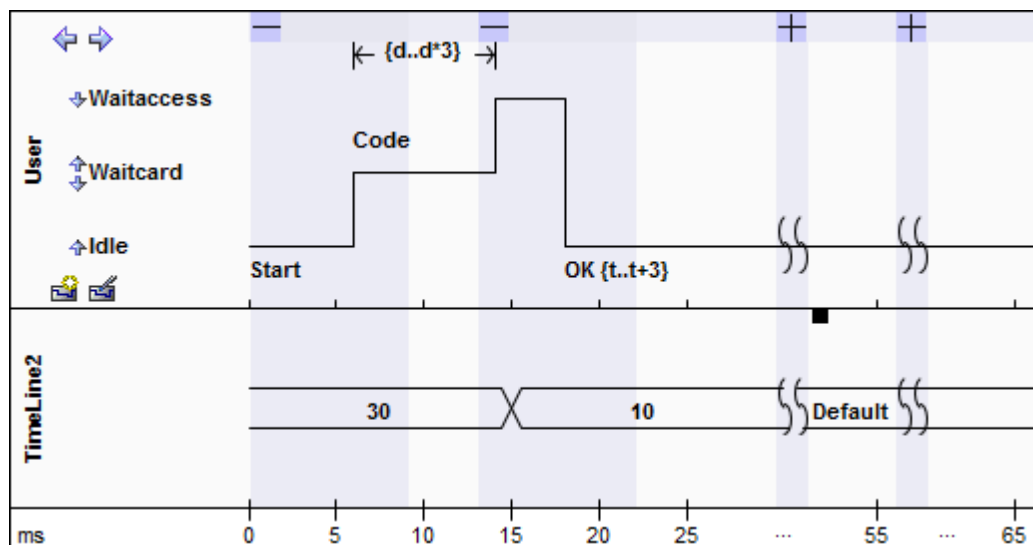
#### 6.3.5.5.2 Compress Time Intervals

You can compress Time Intervals to conserve space on long timelines.

##### Uncompressed Time Intervals



##### Compressed Time Intervals



Notice:

Item	Description
	<p>The compression toggle boxes:</p> <ul style="list-style-type: none"> <li> is expanded, click on this to compress the selected time interval</li> <li> is compressed, click on this to expand the selected time interval again</li> </ul>
	<p>The compressed sections of the timelines themselves, in all elements.</p> <p>If there is space between the paired symbols, there are transitions within the compressed section.</p> <p>If the timeline continues through the paired symbols there are no transitions in the compressed section.</p>
25 ... 55	The compressed sections in the time range underneath the elements.

You can also compress and expand Time Intervals using context menu options,

#### Learn more

- [Time Interval Operations on Transitions](#) <sup>[1246]</sup>

#### 6.3.5.5.3 Select Time Intervals

Task	Action	See also
<b>Select a Time Interval across all elements on the diagram</b>	Click on the Interval Bar within the Time Interval.	

<b>Select a number of individual Time Intervals</b>	Press and hold ( <b>Ctrl</b> ) while clicking on the Interval Bar within each Time Interval.	
<b>Select all Time Intervals in a range</b>	Click on the Interval Bar within the first Time Interval in the range, then press and hold ( <b>Shift</b> ) and click on the Interval Bar within the last Time Interval in the range.  All Time Intervals between the two are selected.	

After you have selected one or more Time Intervals, you can modify the selection in the following ways:

Task	Action	See also
<b>Exclude Lifeline elements from the selection</b>	Press and hold ( <b>Ctrl</b> ) and click on any part of the selection within that element.  Repeat the step to toggle the selection and re-include the element.	<a href="#">Toggle Interval Selection</a> <sup>[1246]</sup>
<b>Select only one Lifeline element and exclude all others</b>	Press and hold ( <b>Shift</b> ) and click on any part of the selection within that element.	

#### 6.3.5.5.4 Time Interval Operations

You can select and update specific Time Intervals.

Right-click on the Interval Bar within an interval. A context menu displays providing the following options:

Option	Action	See also
<b>Select Interval</b> <b>Deselect Interval</b>	Select the Time Interval or, if the interval is already selected, deselect it.  You can select several Time Intervals in this way, accessing the menu separately on each interval.	<a href="#">Select Time Intervals</a> <sup>[1245]</sup>
<b>Toggle Interval Selection</b>	Switch the selection or deselection of the Time Interval within the selected Timeline element.  You select or deselect a Time Interval across all Timeline elements, but the Toggle option acts only on the element in which you access the menu.	<a href="#">Select Time Intervals</a> <sup>[1245]</sup>
<b>Compress Interval</b>	Compress the Time Interval, and hide all transitions within that Time Interval.  This is also useful for hiding long sections of inactivity on the time	<a href="#">Compress Timeline</a> <sup>[1247]</sup>

Option	Action	See also
	line.	
<b>Remove Interval</b>	Delete the Time Interval.	
<b>Copy</b>	Copy the transitions for all selected Time Intervals.	
<b>Cut</b>	Copy and delete the selected transitions from the diagram.	
<b>Cut and Remove Time</b>	Copy and delete the transitions that lie in the selected Time Intervals from the diagram.  This option also removes time from the timeline, the amount being the duration of the Time Interval.  All transitions and Time Intervals to the right of the selected time interval are moved left.	
<b>Delete</b>	Delete the selected transitions from the diagram.	
<b>Delete and Remove Time</b>	Delete the transitions that lie in the selected Time Intervals from the diagram.  This option also removes time from the timeline, the amount being the duration of the Time Interval.  All transitions and Time Intervals to the right of the current Time Interval are moved left.	
<b>Insert Time</b>	Add time to the timeline and move all transitions and time intervals to the right.  Also expand the duration of the current Time Interval.	

### Compress Timeline

The Compression toggle boxes and **Compress Interval** menu option operate on the Time Interval and compress the timeline and all transitions within the Interval. You have an alternative option that operates on the timeline and compresses a single transition state.

1. Right-click on the timeline (rather than the Interval Bar) just after a transition, and select the **Compress** context menu option.
2. Enterprise Architect creates a new Time Interval covering the period from the selected transition up to the next transition, and then compresses that Time Interval.

If there are other Time Intervals in this period, Enterprise Architect replaces them with the single Time Interval for the transition state. You should consider this when creating and compressing the Time Interval, as it extends across the other Timeline elements in the diagram.

A value of this method is that it creates a Time Interval for a period in which no transitions occur, which could be lengthy, and then compresses this Time Interval to hide the period of inactivity.

### All Time Intervals in the Diagram

To create a new Time Interval or work across all Time Intervals in the diagram, right-click on the Interval Bar between Time Intervals. A context menu displays, providing the following options (The **Paste ...** menu options become active after transitions have been copied).

Menu Option	Action	See also
<b>Create Time Interval</b>	Create a single Time Interval.	<a href="#">Create Time Intervals</a> <sup>[1248]</sup>
<b>Expand all Time Intervals</b>	Expand all Time Intervals over the whole diagram.	
<b>Compress all Time Intervals</b>	Compress all Time Intervals over the whole diagram.	
<b>Paste Combine</b>	Paste copied transitions over any existing transitions within the copied time frame.  The diagram does not allow two consecutive transitions to the same state, and removes the second transition automatically.	
<b>Paste Remove</b>	Delete all the transitions and then pastes the copied transition within the copied time frame.	
<b>Paste Insert</b>	Insert time, moving all transitions and Time Intervals to the right to make room to paste in the copied transitions.	
<b>Insert Time</b>	Add time to the timeline and move all transitions and Time Intervals to the right.  This option does not change the duration of any Time Interval.	

### Copy and paste transitions from one timeline element to another

Step	Action	See also
<b>1</b>	Press and hold ( <b>Shift</b> ) and select the Timeline element within a Time Interval to copy or cut.	
<b>2</b>	Right-click on the Interval Bar (it doesn't matter which element you select).  The context menu displays.	



Step	Action	See also
3	Copy or cut the transitions. You can also cut and remove time.	<a href="#">Copy or cut</a> <sup>[1247]</sup>
4	Select the timeline to paste transitions to and right-click on the Interval Bar. The context menu displays.	
5	Select one of the paste operations. Note that states are created if they don't already exist in the timeline. Any states that don't exist in the Timeline element you are pasting to are created. Any new states created might be in the wrong order; you can change the order via the diagram quick buttons.	<a href="#">Quick Buttons</a> <sup>[1229]</sup>

#### **Shift transitions within a selected Time Interval or multiple selected Time Intervals**

Step	Action	See also
1	Select all the Time Intervals containing the transitions to be shifted.	<a href="#">Select Time Intervals</a> <sup>[1245]</sup>
2	Press and hold ( <b>Shift</b> ) and click on the Interval Bar (it doesn't matter which Timeline element you select), and move the transition left or right.  You cannot drag transitions over other transitions; the move stops when the moved transition collides with a stationary transition.  If you have collision problems, use ( <b>Shift</b> ) + <b>select</b> to shift transitions for a single Timeline element.	

### **6.3.6 Sequence Diagram**

A Sequence diagram is a structured representation of behavior as a series of sequential steps over time. You can use it to:

- Depict work flow, message passing and how elements in general cooperate over time to achieve a result
- Capture the flow of information and responsibility throughout the system, early in analysis; messages between elements eventually become method calls in the Class model
- Make explanatory models for Use Case scenarios; by creating a Sequence diagram with an Actor and elements involved in the Use Case, you can model the sequence of steps the user and the system undertake to complete the required tasks

#### **Construction**

- Each Sequence element is arranged in a horizontal sequence, with messages passing back and forward between elements















- Messages on a Sequence diagram can be of several types; the Messages can also be configured to reflect the operations and properties of the source and target elements (see the Notes in the *Message* topic)
- An Actor element can be used to represent the user initiating the flow of events
- Stereotyped elements, such as Boundary, Control and Entity, can be used to illustrate screens, controllers and database items, respectively
- Each element has a dashed stem called a Lifeline, where that element exists and potentially takes part in the interactions

To toggle the numbering of messages on a Sequence diagram, select or deselect the **Show Sequence Numbering** checkbox on the Options dialog.

**Example Diagram** [Example Sequence Diagram](#) <sup>[1251]</sup>

### Tools

Select Sequence diagram elements and connectors from the [Interaction pages](#) <sup>[806]</sup> of the Toolbox. Click on the following elements and connectors for more information.

Sequence Diagram Elements	Sequence Diagram Connectors
 Actor	 Message
 Lifeline	 Self-Message
 Boundary	 Recursion
 Control	 Call
 Entity	
 Fragment	
 Endpoint	
 Diagram Gate	
 State/Continuation	
 Interaction	

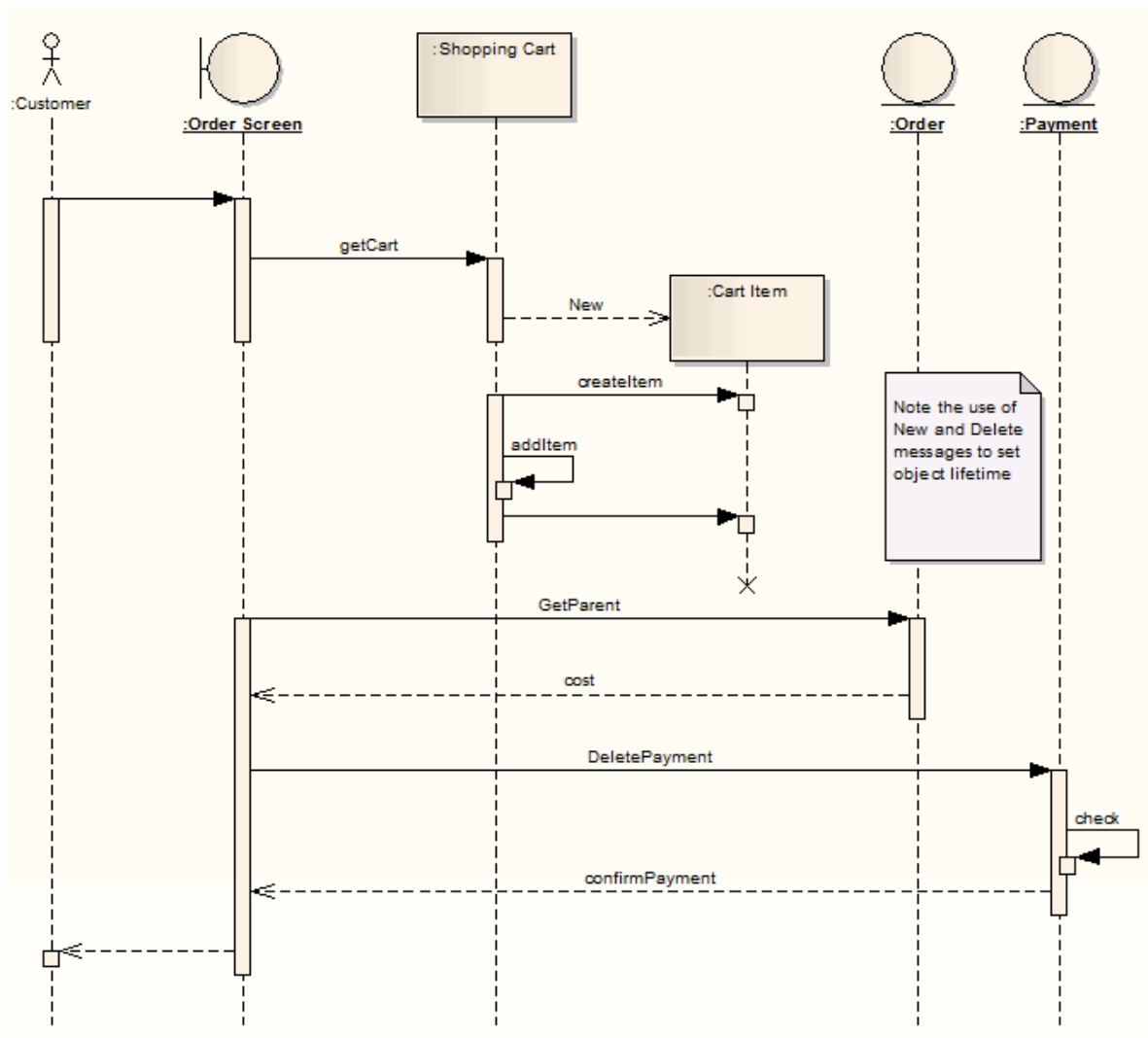
### Learn more

- [Denote the Lifecycle of an Element](#) <sup>[1251]</sup>
- [Layout of Sequence Diagrams](#) <sup>[1252]</sup>
- [Sequence Element Activation](#) <sup>[1255]</sup>
- [Lifeline Activation Levels](#) <sup>[1256]</sup>
- [Message Label Visibility](#) <sup>[1258]</sup>

- [Change the Top Margin](#)<sup>[1258]</sup> (create usable space at the top of the diagram)
- [Change the Timing Details](#)<sup>[1424]</sup>
- [Business Modeling/Interaction](#)<sup>[1805]</sup>
- [Sequence Diagrams and Version Control](#)<sup>[1254]</sup>
- [Show Sequence Numbering](#)<sup>[629]</sup>

### 6.3.6.1 Example Sequence Diagram

The following example Sequence diagram demonstrates several different elements:



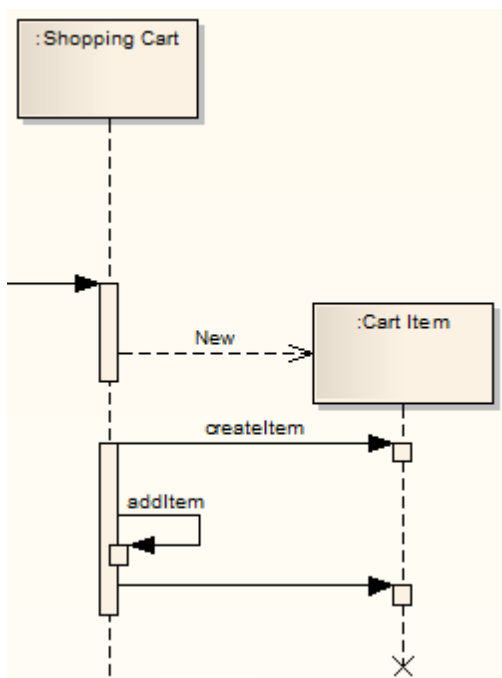
### 6.3.6.2 Denote Lifecycle of an Element

**Capture element lifetimes using messages denoted as New or Delete message types**

Step	Action
1	Double-click on a message within a Sequence diagram to display the Message Properties dialog.

2	In the <b>Lifecycle</b> field, click on the drop-down arrow and select <b>New</b> or <b>Delete</b> .
3	Click on the <b>OK</b> button to save the changes.

**Example Diagram** The example below shows two elements that have specific creation and deletion times



#### Notes

- To show the termination **X** on the lifeline in the example diagram, you must switch on garbage collection: **Tools | Options | Diagram | Sequence: Garbage Collect**

### 6.3.6.3 Layout of Sequence Diagrams

#### Offset the vertical separation of Sequence messages

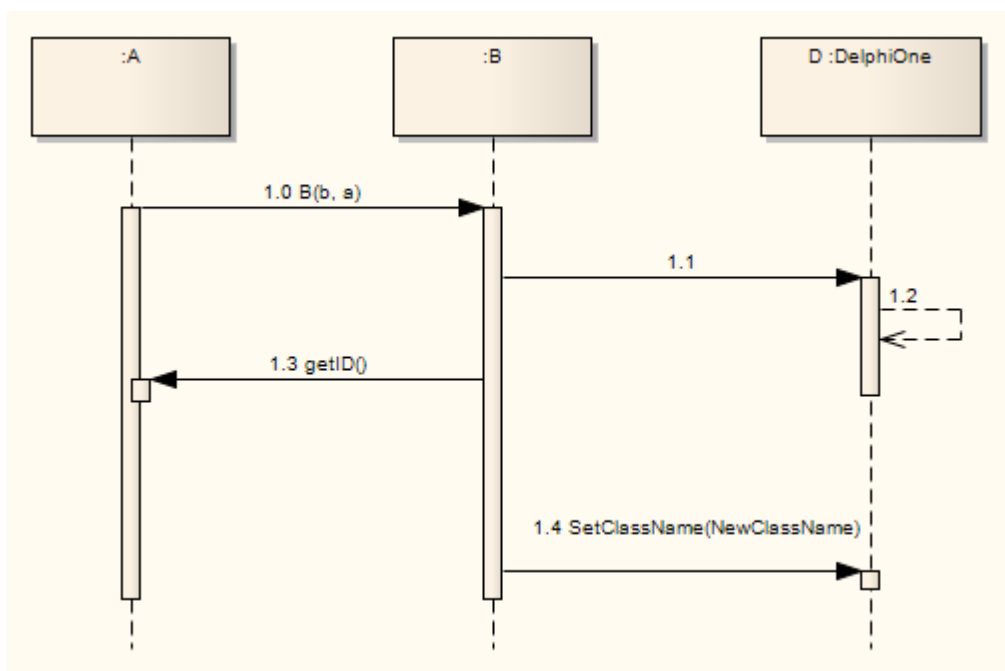
Step	Action	See also
1	Select the appropriate message in a Sequence diagram.	
2	Use the mouse to drag the message up or down as required.  As you drag a message up or down a lifeline, any messages or fragments below that message are shifted up or down the same amount.	

If you press and hold ( **Alt** ) as you drag a message up or down **past** the next or previous message, the messages **swap** positions, rather than simply move position.

As you move one Message past another, a tooltip displays to remind you to press and hold **[Alt]**, regardless of whether you are actually pressing the key. You can hide this tooltip by deselecting a checkbox on the **Diagram | Sequence** page of the Options dialog.

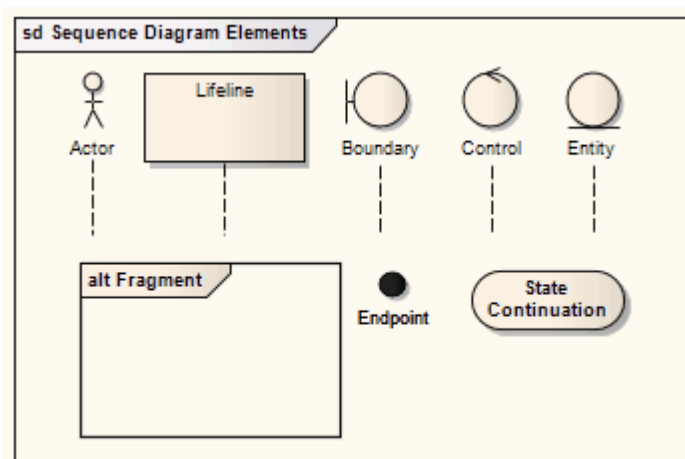
[Sequence Diagram Options](#)  
629

**Example Diagram** The example below shows an economical use of space in a Sequence diagram.



#### 6.3.6.4 Sequence Elements

The example below shows some possible elements of Sequence diagrams and their stereotyped display.



Element	Description	See also
<b>Actor</b>	An instance of an actor at runtime; this can be depicted either as the human figure or in rectangle notation.	<a href="#">Actor</a> <sup>[1284]</sup>
<b>Lifeline</b>	An Object element with the stereotype Lifeline.	<a href="#">Lifeline</a> <sup>[1321]</sup>
<b>Boundary</b>	Represents a user interface screen or input/output device.	<a href="#">Boundary</a> <sup>[1997]</sup>
<b>Entity</b>	A persistent element - typically implemented as a database table or element.	<a href="#">Control</a> <sup>[1999]</sup>
<b>Control</b>	The active component that controls what work gets done, when and how.	<a href="#">Entity</a> <sup>[2000]</sup>

### 6.3.6.5 Sequence Diagrams and Version Control

You might create Sequence diagrams that use elements from other packages as the Lifelines within the diagram. In such cases, the diagrams could be corrupted when the element packages are checked in and out under version control. This is because during checkout the elements are first deleted from the model and then re-imported, and although they are reinstated in the diagrams, any Messages connecting them are not.

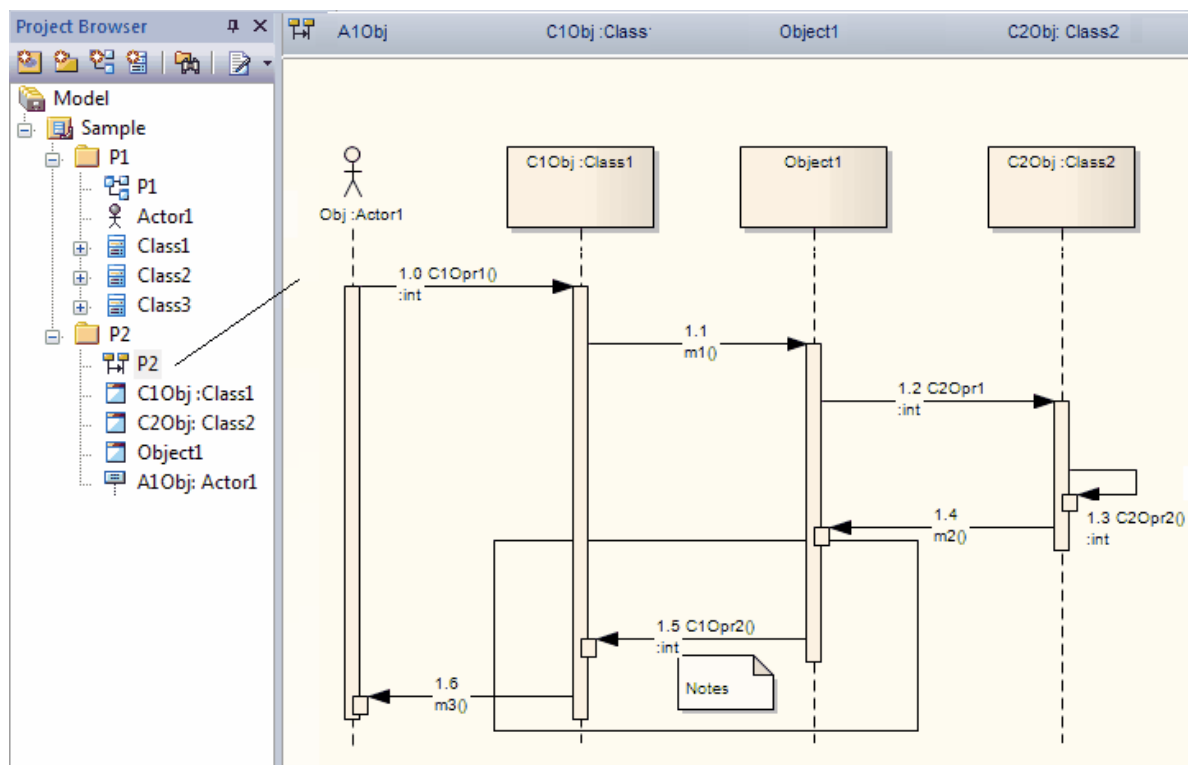
So, if the diagram and its elements reside in different packages, a round-trip of the element package through version control might damage the Sequence diagram.

The solution is to drag-and-drop each Class onto the Sequence diagram as an *object* - when you drop the Class onto the Sequence diagram, in the Paste Element dialog select the **as Instance of Element (Object)** option. This creates a new object in the *diagram*'s parent package, based on the selected Class element. You then create the Messages between the objects.

Therefore, to ensure that a Sequence diagram is not damaged by round-trips of other packages through version control, remember that:

- The Lifelines must be objects (even though you can drop elements as Lifelines onto a Sequence diagram, it is not a strictly UML compliant construct)
- The Lifelines must be in the same package as the diagram.

The following illustration shows the Project Browser with two packages: *P1*, containing the elements, and *P2*, containing a Sequence diagram that uses those elements. The diagram itself is also shown.



This diagram is not damaged, because all the Lifelines are objects and these objects reside in the same package as the Sequence diagram.

### 6.3.6.6 Sequence Element Activation

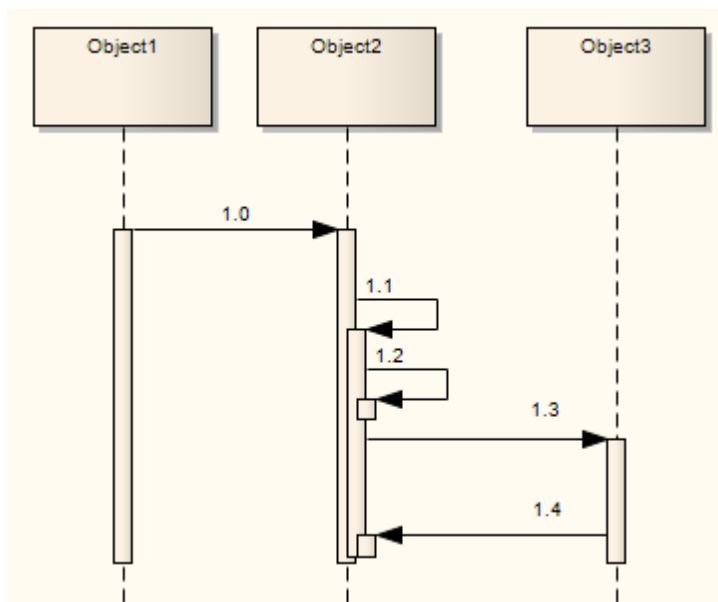
Sequence elements in a Sequence diagram have Activation rectangles drawn along their lifelines. These rectangles describe the time the element is active during the overall period of processing. This visual representation can be suppressed by right-clicking the Sequence diagram, and selecting the **Suppress Activations** context menu option.

In general, Enterprise Architect calculates the period of activation for you, but in some cases you might want to fine tune the rectangle length. There are several context menu options on a sequence message that you can use to accomplish this. To access the following context menu, right-click on the message and select the **Activations** context menu option.

Command	Description
<b>Start New Message Group</b>	Starts off a new round of processing in the current diagram.  This enables you to describe more than one processing scenario in a single diagram.
<b>Extend Source Activation Down</b>	Forces an element to stay active beyond the normal processing period.  This could be used to express an element that continues its own processing concurrently with other processes.

<b>Extend Source Activation Up</b>	Forces an element's activation upwards.
<b>End Source Activation</b>	Truncates the activation of the source element after the current message. This is useful for expressing an asynchronous message after which the source element becomes idle.
<b>End Target Activation</b>	Ends a Forced Activation started by the <b>Extend Source Activation</b> options.
<b>Raise Activation Level</b>	Displays on the context menu only where its use is appropriate. For example, after a self-message the next message starts by default at a lower activation level but the <b>Raise Activation Level</b> command displays on the context menu to enable you to raise its level.
<b>Lower Activation Level</b>	Displays on the context menu only where its use is appropriate.

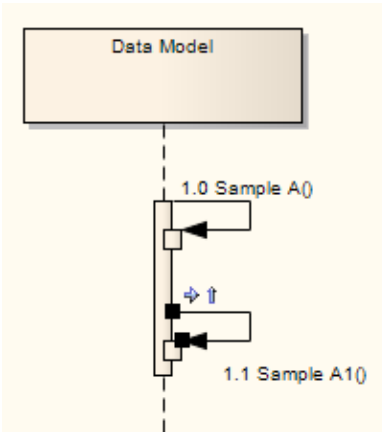
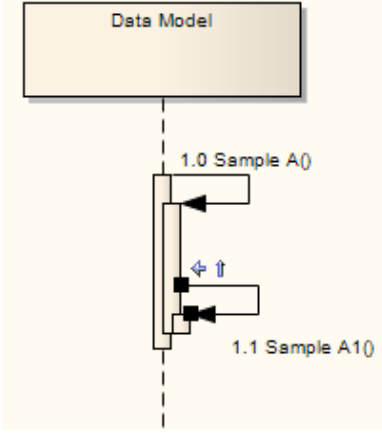
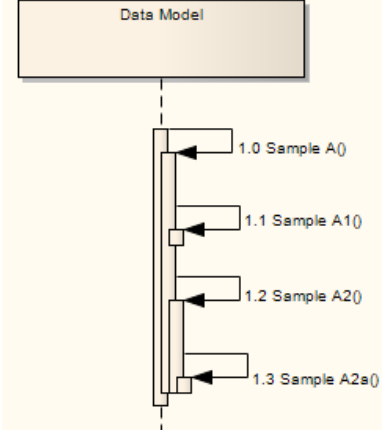
A more convenient way to change activation levels is directly on the diagram. Whenever appropriate, left and/or right arrows display on specific connectors. In the following diagram, see connector 1.3. Click on the arrow to raise or lower the activation level.



### 6.3.6.7 Lifeline Activation Levels

Complicated processing systems can be easily negotiated and reflected in Sequence diagrams, by adding activation layers on a single lifeline.



Example	Description
 <p>The diagram shows a lifeline for 'Data Model'. It starts with a message '1.0 Sample A()' which is a self-call. Below this, there is a blue double-headed arrow indicating a connector. Then, there is a message '1.1 Sample A1()' which is also a self-call. The lifeline is represented by a vertical dashed line with rectangular activation bars.</p>	<p>A Class invokes the method <i>Sample A</i>, which in turn calls <i>Sample A1</i>.</p> <p>To produce the arrangement in the diagram:</p> <ol style="list-style-type: none"> <li>1. Select the <b>More tools   Interaction</b> menu option.</li> <li>2. Click on the <b>Self-message</b> icon in the Interaction Relationships panel.</li> <li>3. Click on the lifeline.</li> </ol>
 <p>This diagram is identical to the previous one, but the blue double-headed arrow (connector) is now pointing upwards, indicating that the activation level of the 'Sample A1' message is being raised to encompass the duration of the 'Sample A' message.</p>	<p>In order to raise the Activation level of <i>Sample A1</i>, click on the raise arrow of the selected connector.</p> <p>The lifeline now visually depicts that method <i>Sample A1</i> is called during the processing of <i>Sample A</i>.</p>
 <p>The diagram shows a single lifeline with four sequential self-calls: '1.0 Sample A()', '1.1 Sample A1()', '1.2 Sample A2()', and '1.3 Sample A2a()'. Each call is represented by a message arrow pointing to the lifeline, with corresponding activation bars.</p>	<p>In the example below, a few more self-messages have been added.</p> <p>The message <i>Sample A2a</i> is called from <i>Sample A2</i>, which in turn is called from <i>Sample A</i> (not <i>Sample A1</i>).</p> <p><i>Sample A1</i> is called from <i>Sample A</i>.</p>

### 6.3.6.8 Sequence Message Label Visibility

#### Hide and show labels used in Sequence messages

Step	Action
1	Right-click on the message within the Sequence diagram and select the <b>Set Label Visibility</b> context menu option.  The Label Visibility dialog displays.
2	Select or clear the checkbox against each message label to display or hide, respectively.
3	Click on the <b>OK</b> button to save the settings.

### 6.3.6.9 Change the Top Margin

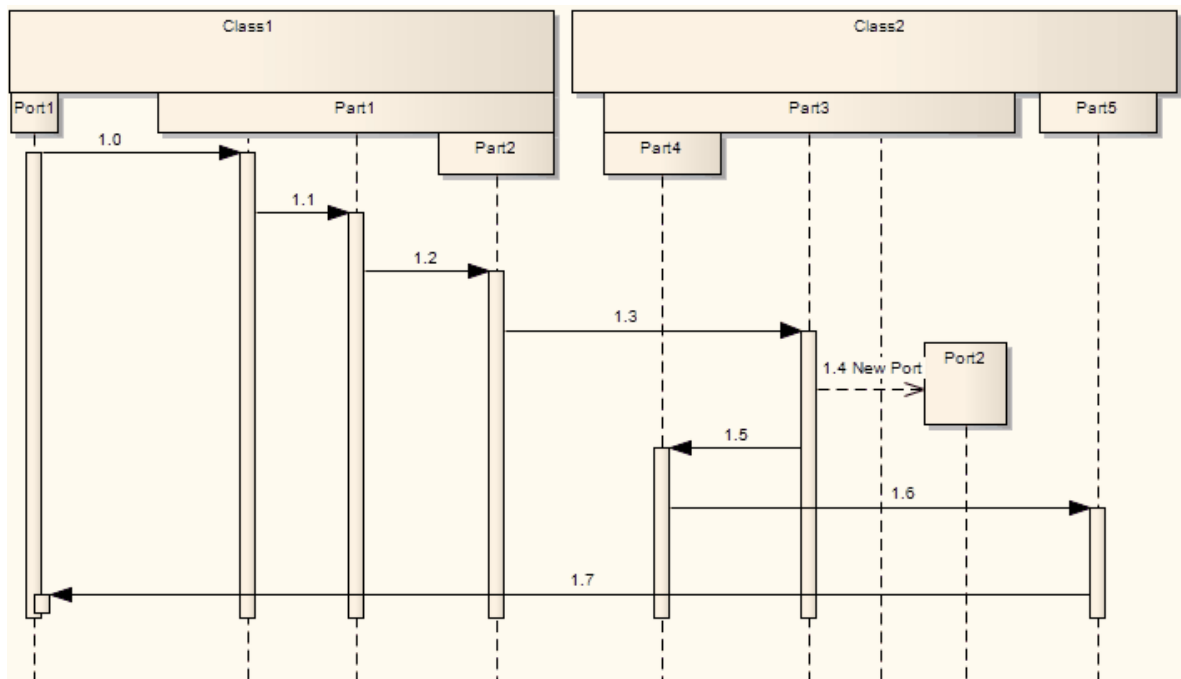
In order to change the top margin of a Sequence diagram from the default 50 units, right-click on the diagram and select the **Set Top Margin** context menu option. You can set the top margin to any value between 30 and 250 units. You can then use this space to, for example, add Note or Text elements to provide documentation on the diagram.

### 6.3.6.10 Inline Sequence Elements

On a Sequence diagram it is possible to represent existing child Part and Port elements, which render as inline sequence elements under their parent Class sequence element.

#### Represent Part and Port elements on a Sequence diagram

Step	Action
1	Right-click on the sequence elements containing the child Ports or Parts, and select the <b>Structural Elements</b> context menu option.  The Structural Elements dialog displays.
2	Select the checkbox against each Part or Port to show, and click on the <b>Close</b> button.



#### Learn more

- [Manage Structural Elements](#) <sup>[935]</sup>
- [Part](#) <sup>[1383]</sup>
- [Port](#) <sup>[1384]</sup>

### 6.3.7 Communication Diagram

A Communication diagram is a diagram that shows the interactions between elements at run-time in much the same manner as a Sequence diagram. However, Communication diagrams are used to visualize inter-object relationships, while Sequence diagrams are more effective at visualizing processing over time.

Communication diagrams employ ordered, labeled associations to illustrate processing. Numbering is important to indicate the order and nesting of processing. A numbering scheme could be:










1  
1.1  
1.1.1  
1.1.2  
1.2, and so on.

A new number segment begins for a new layer of processing, and would be equivalent to a method invocation.

**Example Diagram** [Example Communication Diagram](#) <sup>[1260]</sup>

#### Tools

Select Communication diagram elements and connectors from the Communication pages of the Toolbox. Click on the following elements and connectors for more information.

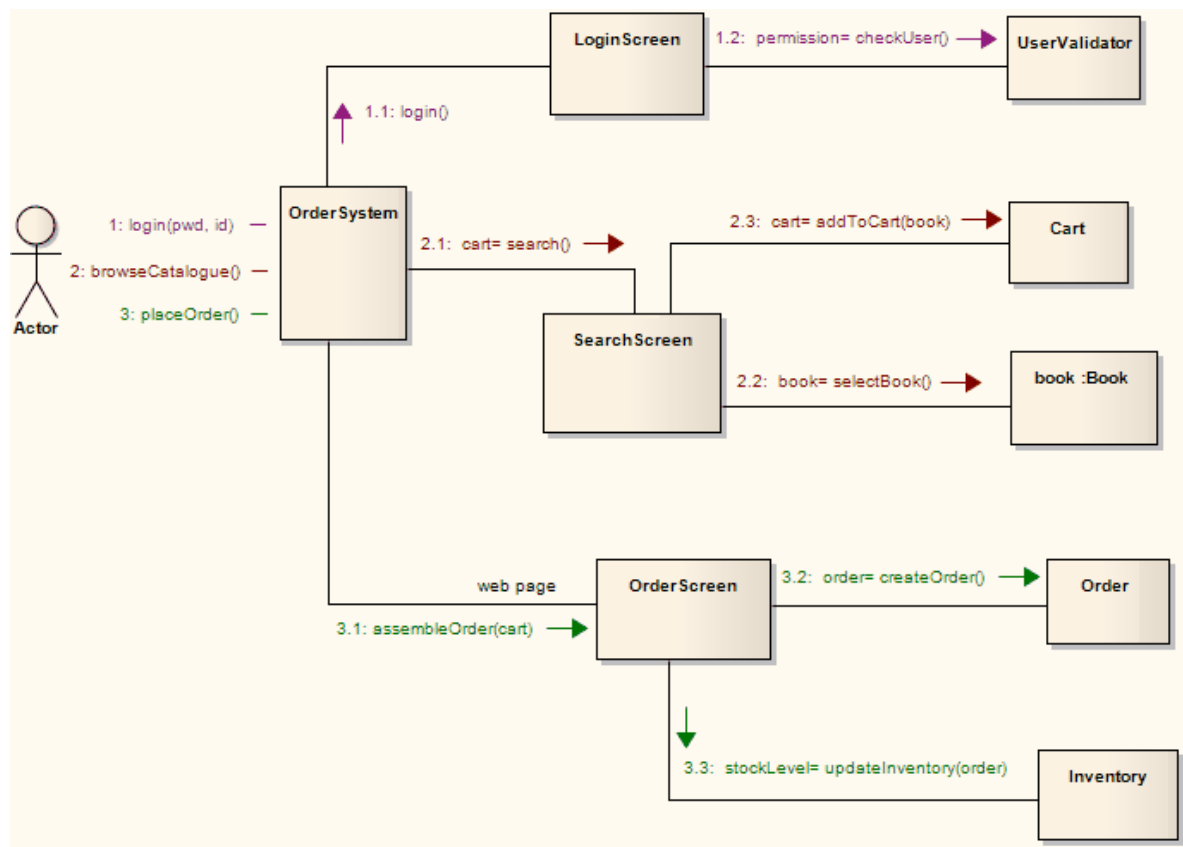
Communication Diagram Elements	Communication Diagram Connectors
 Actor	 Associate
 Object	 Nesting
 Boundary	 Realize
 Control	
 Entity	
 Package	

### Learn more

- [Communication Diagrams in Color](#)<sup>[126]</sup>

#### 6.3.7.1 Example Communication Diagram

This example illustrates a Communication diagram among cooperating object instances. Note the use of message levels to capture related flows, and the different colors of the messages.

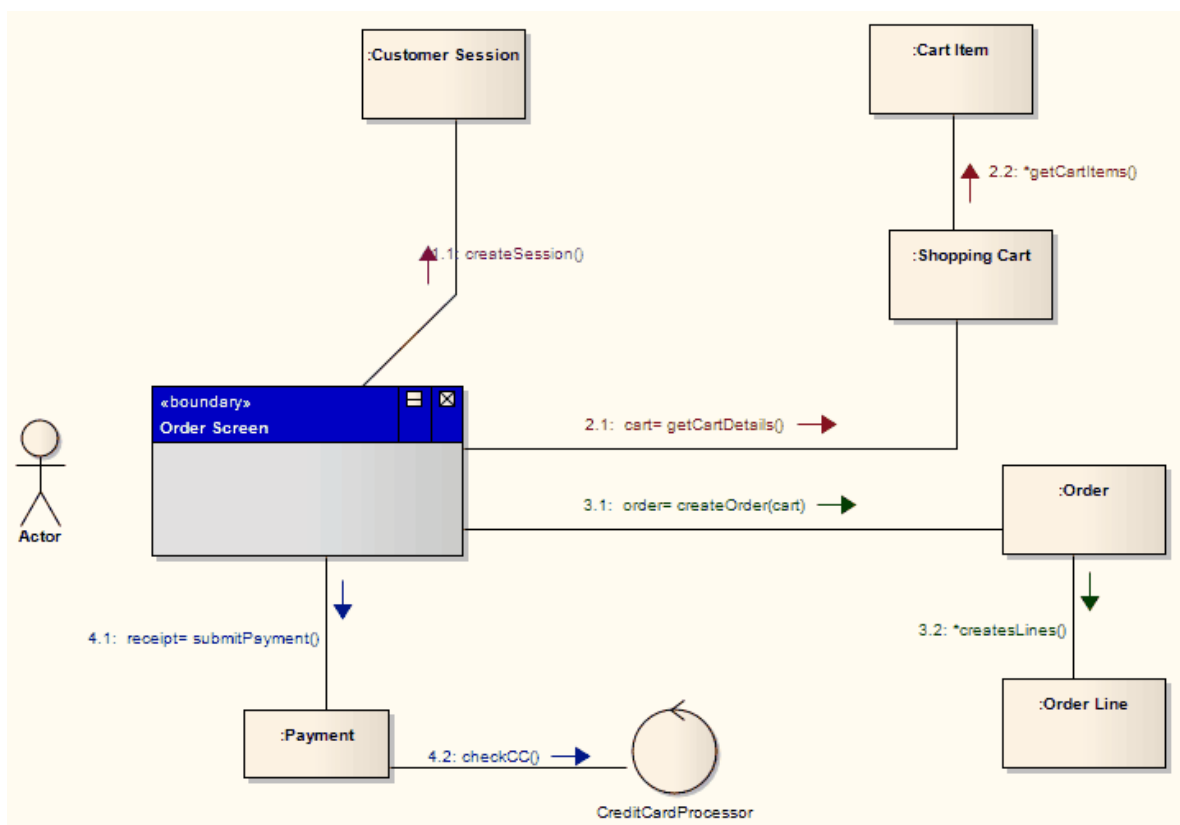


### 6.3.7.2 Communication Diagrams in Color

It is possible to highlight particular message flows in a Communication diagram using different colors for each message set.

#### Highlight the colors in a Communication diagram

Step	Action
1	Select the <b>Tools   Options   Communication Colors</b> menu option. The Communication Message Coloring page of the Options dialog displays.
2	Select the <b>Use Communication Color</b> checkbox.
3	Click on the drop-down arrow of each <b>Message n</b> field, and select the required color for each message group.
4	Click on the <b>Close</b> button. On your Communication diagram, each sequence group of messages displays in a different color, as shown.



### 6.3.8 Interaction Overview Diagram

Interaction Overview diagrams visualize the cooperation between other interaction diagrams to illustrate a control flow serving an encompassing purpose. As Interaction Overview diagrams are a variant of Activity diagrams, most of the diagram notation is the same, as is the process of constructing the diagram.

Decision points, Forks, Joins, Start points and End points are the same. Instead of Activity elements, however, rectangular elements are used. There are two types of these elements:
















- Interaction elements display an inline Interaction diagram, which can be any one of the four types (Sequence, Timing, Communication or Interaction Overview)
- Interaction Occurrence elements are references to an existing Interaction diagram: they are visually represented by a frame, with **ref** in the frame's title space; the diagram name is indicated in the frame contents


To create an Interaction Occurrence, simply drag an Interaction diagram from the Project Browser onto your Interaction Overview diagram. The **ref** frame displays, encapsulating an instance of the Interaction diagram.

**Example Diagram** [Example Interaction Overview Diagram](#) 




#### Tools

Select Interaction Overview diagram elements and connectors from the Activity pages of the Toolbox. Click on the following elements and connectors for more information.

Interaction Overview Diagram Elements	Interaction Overview Diagram Connectors
 Partition	 Control Flow
 Decision	 Object Flow
 Send	 Interrupt Flow
 Receive	
 Synch	
 Initial	
 Final	
 Flow Final	
 Region	
 Exception	
 Merge	
 Fork/Join	

Interaction Overview Diagram Elements	Interaction Overview Diagram Connectors
 Fork/Join	

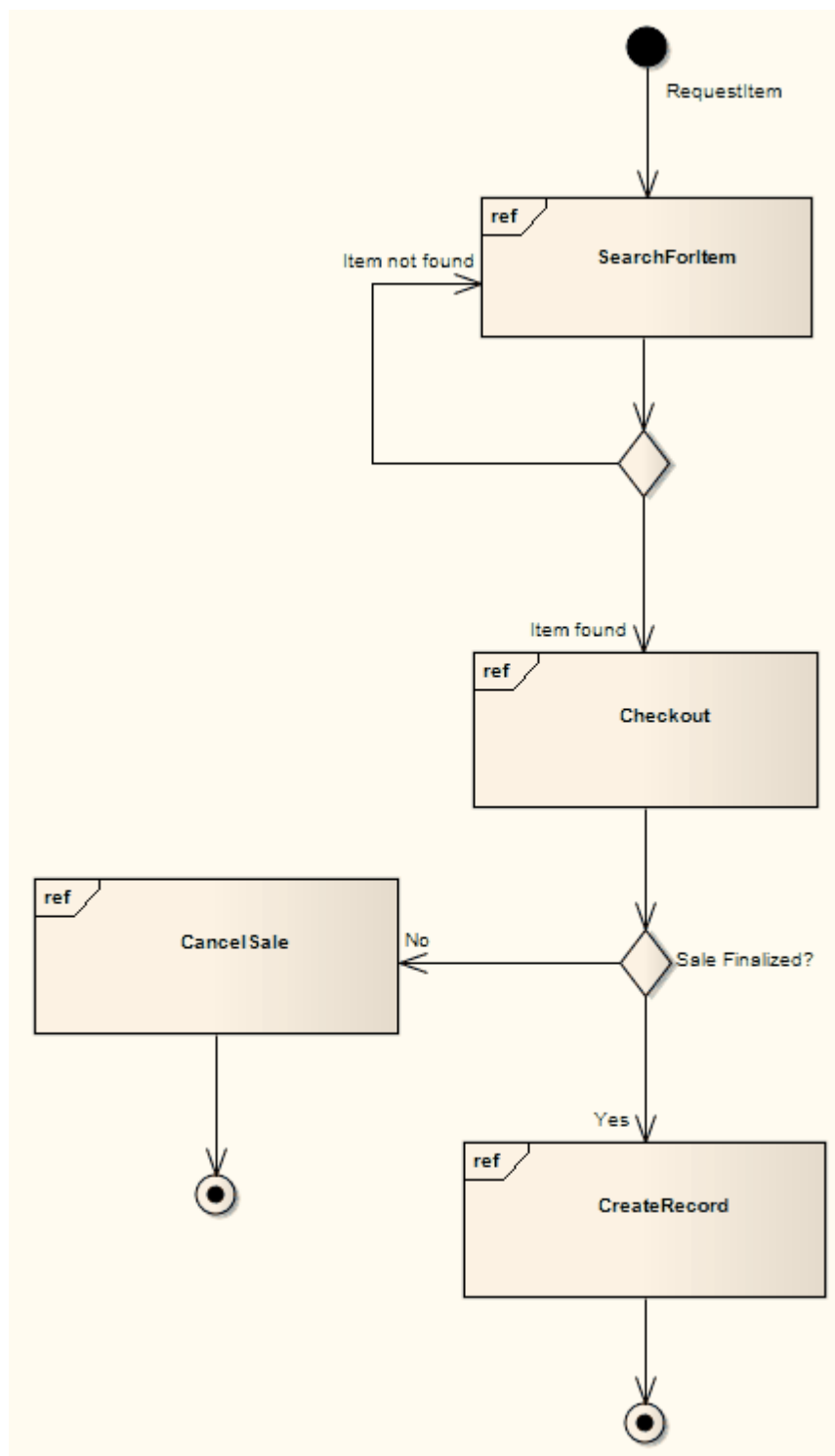
**Learn more**

- [Activity diagrams](#)  f1199
- [Interaction](#)  f1314
- [Interaction Occurrence](#)  f1317

**6.3.8.1 Example Interaction Overview Diagram**

This diagram depicts a sample sale process, shown in an Interaction Overview diagram, with sub-processes abstracted within Interaction Occurrences.

The diagram appears very similar to an Activity diagram, and is conceptualized the same way; as the flow moves into an interaction, the respective interaction's process must be followed before the Interaction Overview's flow can advance.





## 6.4 UML Elements

Models in UML are constructed from elements such as *Classes*, *Objects*, *Interfaces*, *Use Cases*, *Components* and *Nodes*, each of which has a different purpose, different rules and different notation. Model elements are used at different stages of the design process for different purposes. For example,

- During early analysis, Use Cases, Activities, Business Processes, Objects and Collaborations are used to capture the problem domain
- During elaboration, Sequence diagrams, Objects, Classes and State Machines are used to refine the system specification
- Components and Nodes are used to model larger parts of the system as well as the physical entities that are created and deployed into a production environment

### Behavioral Diagram Elements

Behavioral diagrams depict the behavioral features of a system or business process. Elements that can appear on Behavioral diagrams include Activity, Interaction, Lifeline, State Machine and Use Case.

### Structural Diagram Elements

Structural diagrams depict the structural elements composing a system or function. Elements that can appear on Structural diagrams include Class, Component, Interface, Node and Package.

#### Learn more

- [Behavioral Diagram Elements](#)<sup>[1265]</sup>
- [Structural Diagram Elements](#)<sup>[1357]</sup>

### 6.4.1 Behavioral Diagram Elements

The following elements are used in UML Behavioral Diagrams. For more information on using each element, click on the element name in this list:

- [Action](#)<sup>[1266]</sup>, [Activity](#)<sup>[1279]</sup>, [Actor](#)<sup>[1284]</sup>
- [Central Buffer Node](#)<sup>[1285]</sup>, [Choice](#)<sup>[1286]</sup>, [Collaboration](#)<sup>[1367]</sup>, [Collaboration Use](#)<sup>[1368]</sup>, [Combined Fragment](#)<sup>[1287]</sup>
- [Datastore](#)<sup>[1293]</sup>, [Decision](#)<sup>[1294]</sup>, [Diagram Frame](#)<sup>[1296]</sup>, [Diagram Gate](#)<sup>[1297]</sup>
- [Endpoint](#)<sup>[1298]</sup>, [Entry Point](#)<sup>[1300]</sup>, [Exception](#)<sup>[1300]</sup>, [Expansion Region](#)<sup>[1301]</sup>, [Exit Point](#)<sup>[1304]</sup>
- [Final](#)<sup>[1305]</sup>, [Flow Final](#)<sup>[1306]</sup>, [Fork](#)<sup>[1307]</sup>
- [History](#)<sup>[1311]</sup>
- [Initial](#)<sup>[1313]</sup>, [Interaction](#)<sup>[1314]</sup>, [Interaction Occurrence](#)<sup>[1317]</sup>, [Interruptible Activity Region](#)<sup>[1316]</sup>
- [Join](#)<sup>[1307]</sup>, [Junction](#)<sup>[1319]</sup>
- [Lifeline](#)<sup>[1321]</sup>
- [Merge](#)<sup>[1322]</sup>, [Message Endpoint](#)<sup>[1322]</sup>, [Message Label](#)<sup>[1323]</sup>
- [Note](#)<sup>[1324]</sup>
- [Object](#)<sup>[1379]</sup>
- [Package](#)<sup>[1382]</sup>, [Partition](#)<sup>[1325]</sup>
- [Receive](#)<sup>[1327]</sup>, [Region](#)<sup>[1328]</sup>

- [Send](#)<sup>[1328]</sup>, [State](#)<sup>[1329]</sup>, [State/Continuation](#)<sup>[1332]</sup>, [State Lifeline](#)<sup>[1335]</sup>, [State Machine](#)<sup>[1338]</sup>, [Structured Activity](#)<sup>[1338]</sup>, [Synch](#)<sup>[1346]</sup>, [System Boundary](#)<sup>[1347]</sup>
- [Terminate](#)<sup>[1350]</sup>, [Trigger](#)<sup>[1350]</sup>
- [Use Case](#)<sup>[1352]</sup>
- [Value Lifeline](#)<sup>[1355]</sup>

#### Learn more

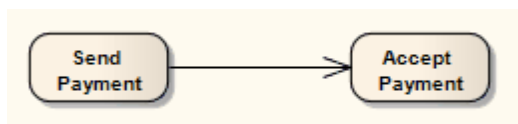
- [Behavioral Diagrams](#)<sup>[1198]</sup>

### 6.4.1.1 Action



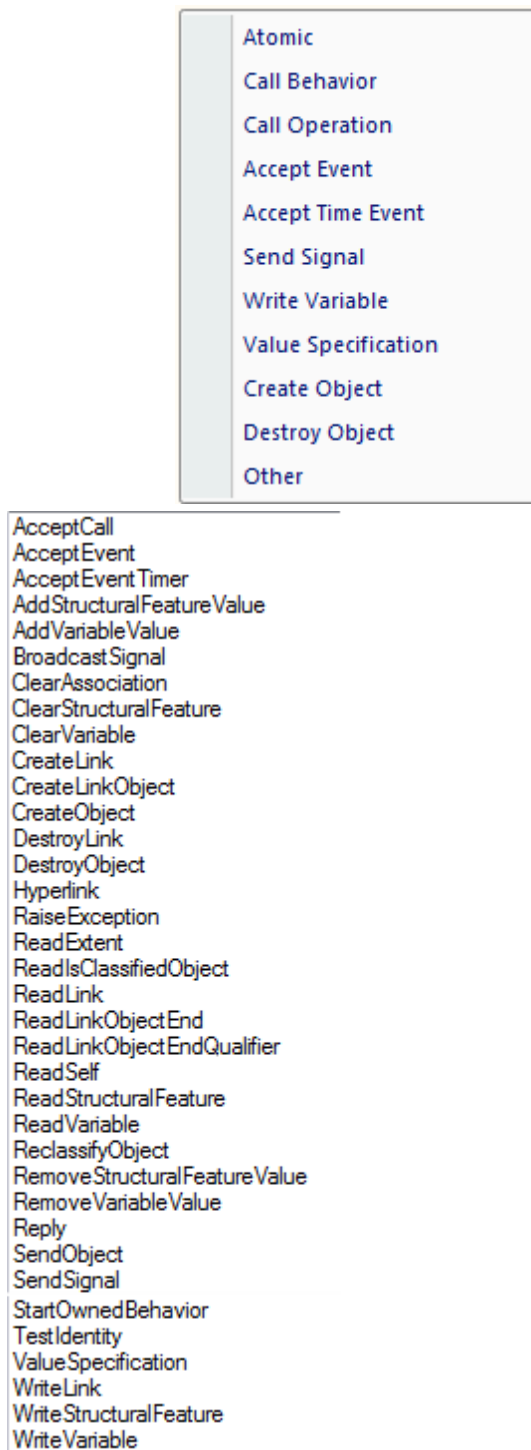
#### Description

An Action element describes a basic process or transformation that occurs within a system, and is the basic functional unit within an Activity diagram. Actions can be thought of as children of Activities; both represent processes, but Activities can contain multiple steps or decomposable processes, each of which can be embodied in an Action. An Action cannot be further broken down or decomposed.



For the purposes of simulation, you can define the effect of a basic (Atomic) Action on the Effect page of the element Properties dialog, using a Javascript expression to define the duration of the effect and selecting to display the effect on the diagram. An Action can be further defined with pre-condition and post-condition notes.

Certain properties can be graphically depicted on the Action. When you first drag the Action icon from the Toolbox onto a diagram, the system prompts you to select from a list of the more common types of Action to create. If you select the **Other** option on this list, the New Action dialog displays; the **Other** drop-down list on this dialog enables you to select a more specialized type of Action from a complete list of Action types.



If you later decide that the Action type is not appropriate, you can change it on the Advanced page of the Action element Properties dialog - select the required new type from the **Kind** drop-down list. For a Value Specification Action, you can also set the value on this page.

The data values passed out of and into an Action can be represented by Action Pins. For an Action type other than a basic Action, you can also assign Action Pins to represent specific properties.

An Action can also be depicted as an Expansion Node to indicate that the Action comprises an Expansion Region.

Toolbox iconLearn more

- [Activity Element](#) <sup>[1278]</sup>
- [Activity Diagram](#) <sup>[1199]</sup>
- [Local Pre/Post Conditions](#) <sup>[1274]</sup>
- [Action Types](#) <sup>[1268]</sup>
- [Action Pin](#) <sup>[1277]</sup>
- [Assign Action Pins](#) <sup>[1278]</sup>
- [Expansion Node](#) <sup>[1301]</sup>
- [Expansion Region](#) <sup>[1301]</sup>
- [Class Operations in Diagrams](#) <sup>[1275]</sup>
- [Code Generation - Activity Diagrams](#) <sup>[2134]</sup>
- [Add Action As Hyperlink](#) <sup>[2004]</sup>
- [Action Behavior By Type](#) <sup>[2497]</sup> (Simulation)

OMG UML Specification

The OMG UML specification (UML Superstructure Specification, v2.1.1, p. 241) states:

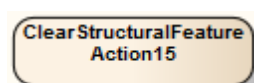
*An action is a named element that is the fundamental unit of executable functionality. The execution of an action represents some transformation or processing in the modeled system, be it a computer system or otherwise.*

The OMG UML specification (UML Superstructure Specification, v2.1.1, p. 313) also states:

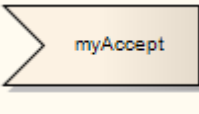

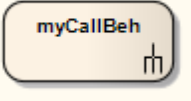
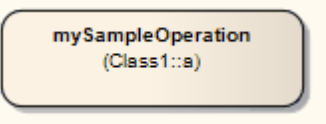

*An action may have sets of incoming and outgoing activity edges that specify control flow and data flow from and to other nodes. An action will not begin execution until all of its input conditions are satisfied. The completion of the execution of an action may enable the execution of a set of successor nodes and actions that take their inputs from the outputs of the action.*

**6.4.1.1.1 Action Types**

Action elements are extremely versatile. Enterprise Architect supports a wide range of specific Action types that you can use to represent or enact a discrete object, operation or behavior. Actions of most types are depicted as a round-cornered rectangle containing the Action type and Action name, as shown.



Certain types have their own specific **notation**; for example:

Action Kind	Notation	See also
<b>AcceptEvent</b>		
<b>AcceptEventTimer</b>		<a href="#">AcceptEvent Actions</a> <sup>[1269]</sup>
<b>CallBehavior</b>		<a href="#">CallBehavior Action</a> <sup>[1271]</sup>
<b>CallOperation</b>		<a href="#">Class Operations in Diagrams</a> <sup>[1275]</sup>
<b>SendSignal</b>		<a href="#">SendSignal Action &amp; BroadcastSignal Action</a> <sup>[1270]</sup>

### AcceptEvent Actions

An *Accept Event* Action element has a selectable output result Action Pin assigned to it, and one or more Triggers to denote the type of events accepted by the Action. You define the Triggers on the Triggers page of the Properties dialog. In a simulation, an AcceptEvent Action without a Trigger will block the simulation at the Action element.

Field	Action	See also
<b>Name</b>	Type in the name of the trigger.	
<b>Type</b>	<p>Click on the drop-down arrow and select the type of trigger: <b>Call</b>, <b>Change</b>, <b>Signal</b> or <b>Time</b>:</p> <ul style="list-style-type: none"> <li>• <b>Call</b> - specifies that the event is a CallEvent, which sends a message to the associated object by invoking an operation</li> <li>• <b>Change</b> - specifies that the event is a ChangeEvent, which indicates that the transition is the result of a change in value of an attribute</li> </ul>	

Field	Action	See also
	<ul style="list-style-type: none"> <li>• <b>Signal</b> - specifies that the event is a SignalEvent, which corresponds to the receipt of an asynchronous signal instance</li> <li>• <b>Time</b> - corresponds to a TimeEvent; which specifies a moment in time</li> </ul> <p>Code generation for State Machines currently supports Change and Time trigger events only, and expects a specification value.</p> <p>In simulation, each Trigger should have a Signal. The result will be the Accept signal.</p>	
<b>Specification</b>	<p>Specify the event instigating the Transition.</p> <p>For an AcceptEvent<b>Timer</b> Action, you can type a JavaScript expression in this field evaluating to the number of ticks to wait for.</p>	

#### SendSignal Action & BroadcastSignal Action

A **SendSignal** Action has an assigned target ActionPin and a Signal. The Signal can have input ActionPins that bind its attribute parameters as arguments. For example:

```
:: Sender : sig. binding_To_s1: Integer
```

In a model simulation, a SendSignal Action will transfer the values of the arguments into the attributes of the created Signal instance. The target ActionPin can have an empty object, to send the Signal into the root of the simulation space. If there is no target ActionPin, simulation will stop at the Action. If the target has an Object, the Signal will be sent to the Object. You must specify the Pin type of the target ActionPin in the classifier of the Object.

A **BroadcastSignal** Action is similar to a SendSignal Action, except that it does not have a target ActionPin. In a simulation, it always sends its Signal to the root of the simulation data.

You can model the Signal to be sent and the associated arguments to be conveyed, using the Signal tab of the element Properties dialog.

Field/Button	Action
<b>Signal</b>	Click on the ( ... ) button and select the required signal from the Select Signal dialog.
<b>Attribute</b>	Click on the drop-down arrow and select the attribute (as previously created in the Signal element) with which the arguments are to be associated.
<b>Value</b>	Type the appropriate value for the attribute.
<b>Add</b>	Click on this button and select the appropriate ActionPins from the Select Pin dialog, to identify the arguments for the Signal.

Field/Button	Action
	To assign more than one ActionPin, press ( <b>Ctrl</b> ) whilst you select each one.
<b>Save</b>	Click on this button to save your changes.

### CallBehavior

A **CallBehavior Action** has a behavior such as an Activity, and a selectable ActionPin result that will put the return value. The CallBehavior Action can also transfer the values of its argument ActionPins into its behavior, if they are bound together. In a simulation, if the Action has no behavior, the simulation is blocked.

### SendObject Action

A **SendObject Action** sends a copy of an Object from the requesting ActionPin to the target ActionPin. In a simulation, the SendObject Action must have both ActionPins, otherwise the simulation is blocked at the Action.

### Structural Feature Actions

A **StructuralFeature Action** acts upon a modeling structural feature, such as a Port, Part or attribute of an Activity or of the classifier of an Object, which you identify within the Action element. Enterprise Architect supports the following types of Structural Feature Actions.

Action	Description
<b>AddStructuralFeatureValue</b>	<p>Requires an object input ActionPin where the target object will be entered, and a result output ActionPin to hold the read result. If the object Port has no value at run time, the process will pause at the Action.</p> <p>In your model design, the Port should be connected to the Port of an Object or to an Object Node of an Activity. The properties of the Port or Object Node must be correctly set, and the value Port must be set up to capture the input value when the Action takes effect.</p> <p>The result ActionPin can be connected to an input consume Port or ActionPin. For example, it can be used at the next WriteStructuralFeature Action as the input value.</p>
<b>ClearStructuralFeature</b>	Clears the single value of a structural attribute or a structural Port of an Object or an Activity, and sets the value of the structural feature to null.
<b>ReadStructuralFeature</b>	<p>Resembles AddStructuralFeatureValue, except that the value Port is not necessary.</p> <p>In a simulation, if the Object's Port has no value at run time, the simulation will pause at the Action.</p>
<b>RemoveStructuralFeature</b>	Similar to ClearStructuralFeature except that it invokes a value ActionPin to

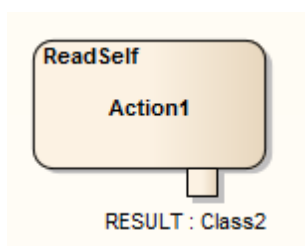
Action	Description
<b>eValue</b>	input a value and, if that value matches the value of the specified structural feature, it sets the value to null.  If the values do not match, the Action does not clear the structural feature value.
<b>WriteStructuralFeature</b>	Identical to AddStructuralFeatureValue. In a simulation, the value Port must be set up to capture the input value when the simulation runs the Action.

Access **Right-click Action element | Advanced | Set Structural Feature: Add**

Step	Action
<b>1</b>	On the Select Property dialog (a variant of the Select <Item> dialog), browse or search for the appropriate structural feature, and double-click on it.  The feature name and location displays in the <b>structuralFeature</b> field of the Set Structural Feature dialog.
<b>2</b>	Click on the <b>OK</b> button to save the setting.

### ReadSelf

A **ReadSelf Action** reads its own host object name into its result Port. You must set an output ActionPin for the result.






The Action must be within a Class, which is instantiated during run time. When a simulation passes the Action, it puts the name of the instance of the Class into the result Port.

ReadSelf is one of a group of Object Actions, with **CreateObject** and **DestroyObject**.

### Learn more

- [Select <Item> Dialog](#) <sup>[994]</sup>
- [Variable Actions](#) <sup>[1273]</sup>
- [Assign Action Pins](#) <sup>[1278]</sup>
- [Model Simulation](#) <sup>[2463]</sup>



- [Objects and Instances in Simulation](#)  <sup>[2479]</sup>
- [Create Objects in a Simulation](#)  <sup>[2480]</sup>
- [Destroy Objects in a Simulation](#)  <sup>[2483]</sup>

#### 6.4.1.1.1.1 Variable Actions

**Variable** Actions are closely concerned with the simulation of the behavior of and actions on Objects in a process. They have an **association variable** in the form of the Tagged Value **variable** with, as its value, the name of an Object in run-time. That is:

```
sim. Object Name
```

Variable Actions provide the variable not only as an Object but also as a property (such as an attribute or Port) of an Object. For example:

```
sim.a.a1
```

The parameter **a.a1** can have an integer value.

Variable Actions include:

- **ReadVariable**
- **WriteVariable**
- **ClearVariable**
- **AddVariableValue**
- **RemoveVariable**

##### ReadVariable

A **ReadVariable** Action has a Result Action Pin as an output Port. The value of the Port will be the result to be read, this being a copy of the variable read. Therefore, it is not affected by changes to the value of the variable. For example, if the variable is **sim.Object.a** that has the value **3**, and its value has been changed into **5** after it is read, the value read is still **3**.

Before reading:

```
sim.Object.a = 3;  
sim.Action1.result = null;
```

After reading:

```
sim.Object.a = 3;  
sim.Action1.result = 3;
```

After a change in the value of the variable:

```
sim.Object.a = 5;  
sim.Action2.value = 3;
```

In that example, the value is a Port of **Action2** that is connected to the result Port of **Action1** by an Object Flow connector.

##### WriteVariable

This Action has a Value Action Pin as an input Port. The value of the Port will be written into its variable. The result value is a copy of the variable from the Value Port.

#### ClearVariable

This Action clears all values of a variable, the variable being either an Object or a value.

#### AddVariableValue

This Action is effectively the same as a **WriteVariable** Action, because the current variables of the simulation do not support multiple values.

#### RemoveVariableValue

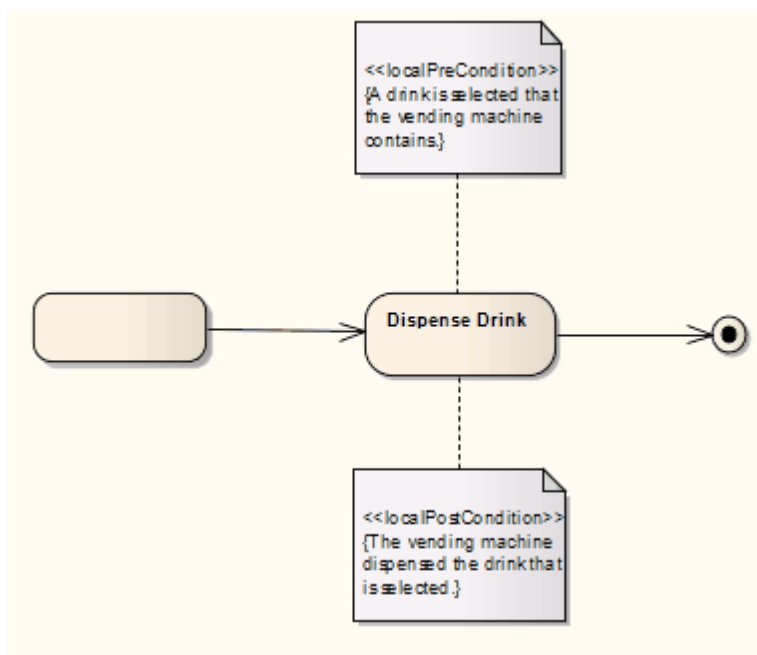
This Action is effectively the same as a **ClearVariable** Action because the current variables of the simulation do not support multiple values.

#### Learn more

- [Object Flow](#) <sup>[1435]</sup>

#### 6.4.1.1.2 Local Pre/Post Conditions

Actions can be further defined with pre-condition and post-condition notes, which constrain an Action's entry and exit. These notes can be added to an Action as defined below.



#### Attach a constraint to an Action

Step	Action
1	Right-click on the Action and select the <b>Add   Constraint</b> context menu option. A <i>Note</i> is created on the diagram, connected to the Action.
2	Right-click on the <i>Note</i> and select the <b>View Properties</b> context menu option. The Constraint dialog displays.
3	In the <b>Constraint Type</b> field, click on the drop-down arrow and select the required constraint type.
4	In the <b>Constraint</b> field, type the text for the constraint.
5	Click on the <b>OK</b> button to save the constraint.

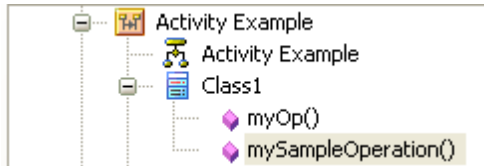
#### Learn more

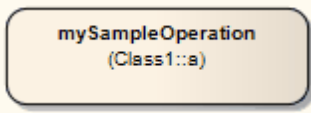
- [Action](#) 

#### 6.4.1.1.3 Class Operations in Diagrams

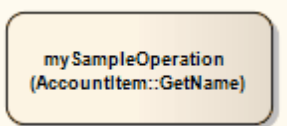
Operations from Classes can be represented by **CallOperation Action** elements on any diagram (such as an Activity, Custom or Analysis diagram). When an operation is shown as an Action, the notation of the element displays the name of the operation prefixed by the name of the Class from which it comes.

#### Add an operation to a diagram

Step	Action	See also
1	Open the target diagram.	
2	From the Project Browser open a Class and locate the operation to be added to the diagram.	
3	Drag the operation on to the diagram. 	
4	When the operation has been added to the diagram, the CallOperation Action	

Step	Action	See also
	<p>resembles the following:</p> 	

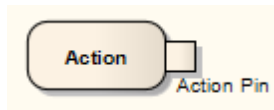
#### Change the operation that an Action refers to

Step	Action	See also
1	<p>Right-click on the Action and select the <b>Advanced   Set Operation</b> menu option.</p> <p>The Set Operation dialog displays.</p>	<a href="#">Set Operation dialog</a> [998]
2	<p>If necessary, in the <b>Go To Namespace</b> field, select the model that contains the operation.</p> <p>Browse for the operation.</p>	
3	<p>When you have located the operation, double-click on it.</p> <p>The Action updates to show the new classifier and operation names.</p> 	

#### Notes

- If you want to locate, in the Project Browser, the operation that an Action was created from, right-click on the Action in the diagram and select the **Find | Locate Operation in Project Browser** context menu option
- If you want to display the previously-generated code for the Class containing the operation, click on the Action in the diagram and press either **Ctrl+E** or **F12**; the Code Editor view displays, with the code generated for the Class (if no code has been generated for the Class, the Code Editor does not display)
- In a simulation, the CallOperation Action must have a calling operation and a target object ActionPin, the operation belonging to the object that comes from the target ActionPin; if you don't set these properties, simulation will be blocked at the Action

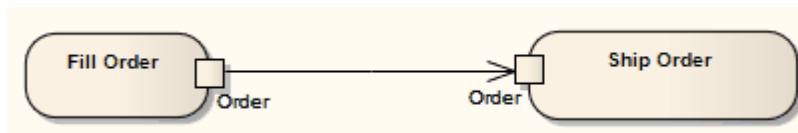
#### 6.4.1.1.4 Action Pin



#### Description

An *Action Pin* is used to define the data values passed out of and into an Action. An *input pin* provides values to the Action, whereas an *output pin* contains the results from that Action.

Action Pins are used below to connect two Actions:



Action Pins can be further characterized as defining exception parameters, streams, or states. Associating a state with a Pin defines the state of input or output values. For instance, the Pin could be called *Orders*, but the state could be *Validated* or *Canceled*.

To **add** an Action Pin to an Action, right-click on the Action to display the context menu and select the **New Child Element | Action Pin** menu option. (You can also **assign** Action Pins, to define specific properties of the Action.)

The Properties dialog of an Action Pin has a Pin page on which you define the specific actions of the Pin.

A Pin serves as an argument for Call Behavior Actions and Call Operation Actions - the Pin name and parameters are shown on the Arguments page of the Action Properties dialog. When an Action is associated with a valid behavior in the model, the associated behavior's parameters are listed in the **Parameter** field drop-down list to facilitate a one-to-one mapping between the argument and the parameter. The fields in the Argument panel of the Pin page are enabled only for Pins belonging to Call Actions, and only when the Action is associated with a valid behavior with valid parameters. To observe this:

1. Create an Activity element and give it an Activity Parameter (right-click on it and select **New Child Element | Activity Parameter**).
2. Create an Action and set the **kind** property to **CallBehavior** (on the Advanced page of the element's Properties dialog).
3. Make the Activity element the classifier for the Action (right-click on the Action element and select **Advanced | Set Behavioral Classifier**; then locate and select the Activity on the Select <Item> dialog).
4. The Structural Elements dialog immediately displays. Select the **Show Owned/Inherited** checkbox; when this is selected, the Activity Parameter is listed in the Defined Elements panel. Select the checkbox against the Activity Parameter, and click on the **Close** button.
5. The Action element now has an Action Pin representing an argument, with the Activity Parameter as the parameter of the argument.

You can also change the *objectState* property of an Action Pin on the Advanced page of the element Properties dialog.

Learn more

- [Action](#) <sup>[1266]</sup>
- [Assign Action Pins](#) <sup>[1278]</sup>
- [Activity Parameter Nodes](#) <sup>[1281]</sup>
- [Object Node](#) <sup>[1325]</sup>

**6.4.1.1.5 Assign Action Pins**

Apart from *adding* Action Pins to any Action, you can *assign* specialized input or output Action Pins to Actions that have a specific type (that is, those that are not Basic or Atomic Actions). These input/output Pins signify various *properties* of the Action - they are not visible as structures on the diagram unless they have previously been added, but are listed in the Project Browser as properties of the Action.

You can only assign Pins that have already been added or assigned to the Action, or that are being created specifically to be assigned to the Action.

**Assign Action Pins to an Action**

Step	Action	See also
<b>1</b>	<p>Right-click on the Action in the diagram, and select the <b>Advanced   Assign Action Pins</b> context menu option.</p> <p>The Assign Action Pins to &lt;ActionName&gt; dialog displays.</p> <p>The format of this dialog depends on the type of Action; for a:</p> <ul style="list-style-type: none"> <li>• <i>SendObject</i> Action the dialog has two fields (<b>request</b> and <b>target</b>)</li> <li>• <i>TestIdentity</i> Action, three</li> <li>• <i>CallBehavior</i> Action, one (<b>result</b>)</li> </ul> <p>The fields are populated in exactly the same way.</p>	
<b>2</b>	<p>The mandatory number and type of Pins are automatically selected (if they exist) or created.</p> <p>To change or add a Pin in a field, click on the corresponding <b>Add</b> button.</p> <p>The Select Pins dialog displays, showing the selected Action and listing all the input Pins currently owned by the Action.</p>	<a href="#">Select &lt;Item&gt; Dialog</a> <sup>[994]</sup>
<b>3</b>	<p>Double-click on one of the Pins (or, depending on the multiplicity of the Pin, <b>Ctrl+click</b> on several Pins).</p> <p>Alternatively, if no suitable Pin exists, click on the <b>Add New</b> button and then click on the newly-created Pin.</p> <p>The selected Pin is identified in the field on the Assign Action Pins to &lt;ActionName&gt; dialog.</p>	
<b>4</b>	Click on the <b>OK</b> button.	

Step	Action	See also

### Notes

- To check the exact location of an assigned Action Pin, you can right-click on the Pin name in the dialog and select the **Find in Project Browser** context menu option

### Learn more

- [Action Pin](#)<sup>[1277]</sup>

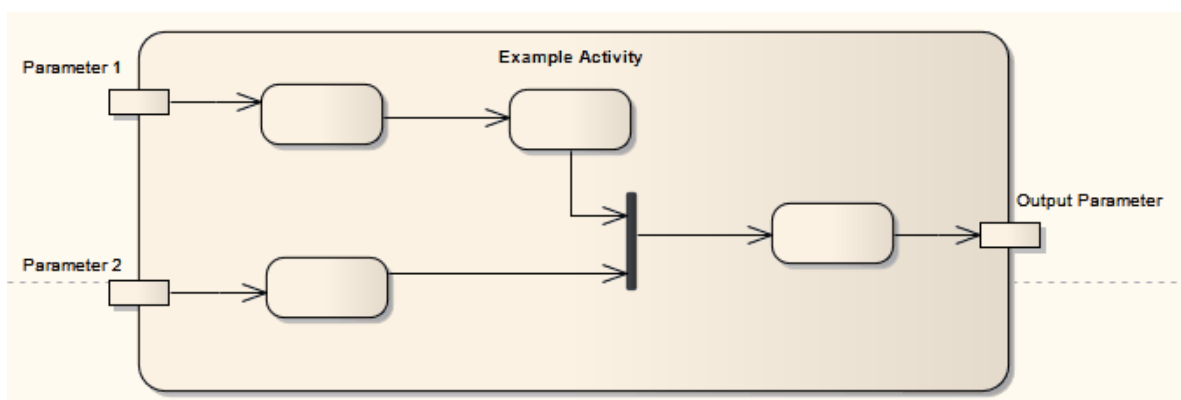
## 6.4.1.2 Activity



### Description

An Activity organizes and specifies the participation of subordinate behaviors, such as sub-Activities or Actions, to reflect the control and data flow of a process. Activities are used in Activity diagrams for various modeling purposes, from procedural-type application development for system design, to business process modeling of organizational structures or work flow.

The following simple diagram of an Activity contains Action elements and includes input parameters and output parameters.



You can define an Activity as a composite element, either during creation or during later edits. When creating a composite Activity element, it is simpler to apply the mechanism for creating Structured Activity elements,

which reduces the number of steps to work through. If converting an existing Activity element, right-click on the element and select the **Advanced | Make Composite** context menu option.

Certain properties can be graphically depicted on an Activity. The Actions in an Activity can be further organized by Activity Partitions.

An Activity can also be depicted as an Expansion Node to indicate that the Activity comprises an Expansion Region.

#### Toolbox icon



#### Learn more

- [Activity Notation](#) <sup>[1281]</sup>
- [Activity Parameter Nodes](#) <sup>[1281]</sup>
- [Activity Partition](#) <sup>[1283]</sup>
- [Action](#) <sup>[1266]</sup>
- [Object Node](#) <sup>[1325]</sup>
- [Activity Diagram](#) <sup>[1199]</sup>
- [Composite Element](#) <sup>[936]</sup>
- [Structured Activity](#) <sup>[1338]</sup>
- [Expansion Node](#) <sup>[1301]</sup>
- [Expansion Region](#) <sup>[1301]</sup>

#### OMG UML Specification

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 318) states:

*An activity specifies the coordination of executions of subordinate behaviors, using a control and data flow model. The subordinate behaviors coordinated by these models may be initiated because other behaviors in the model finish executing, because objects and data become available, or because events occur external to the flow. The flow of execution is modeled as activity nodes connected by activity edges. A node can be the execution of a subordinate behavior, such as an arithmetic computation, a call to an operation, or manipulation of object contents. Activity nodes also include flow-of-control constructs, such as synchronization, decision, and concurrency control. Activities may form invocation hierarchies invoking other activities, ultimately resolving to individual actions. In an object-oriented model, activities are usually invoked indirectly as methods bound to operations that are directly invoked.*

*Activities may describe procedural computation. In this context, they are the methods corresponding to operations on classes. Activities may be applied to organizational modeling for business process engineering and workflow. In this context, events often originate from inside the system, such as the finishing of a task, but also from outside the system, such as a customer call. Activities can also be used for information system modeling to specify system level processes. Activities may contain actions of various kinds:*

- *Occurrences of primitive functions, such as arithmetic functions.*
- *Invocations of behavior, such as activities.*
- *Communication actions, such as sending of signals.*

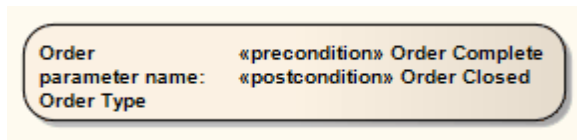


- Manipulations of objects, such as reading or writing attributes or associations.

Actions have no further decomposition in the activity containing them. However, the execution of a single action may induce the execution of many other actions. For example, a call action invokes an operation that is implemented by an activity containing actions that execute before the call action completes.

#### 6.4.1.2.1 Activity Notation

Certain properties can be graphically depicted on an Activity element, as shown:



To define these properties, right-click on the Activity and select the **Properties** context menu option, then select the Advanced page of the Properties dialog.

You can also define the duration (the number of ticks to wait for) of the Activity, using a JavaScript expression. Open the Behavior page of the Properties dialog, and type the JavaScript expression in the **Specification** field.

#### Learn more

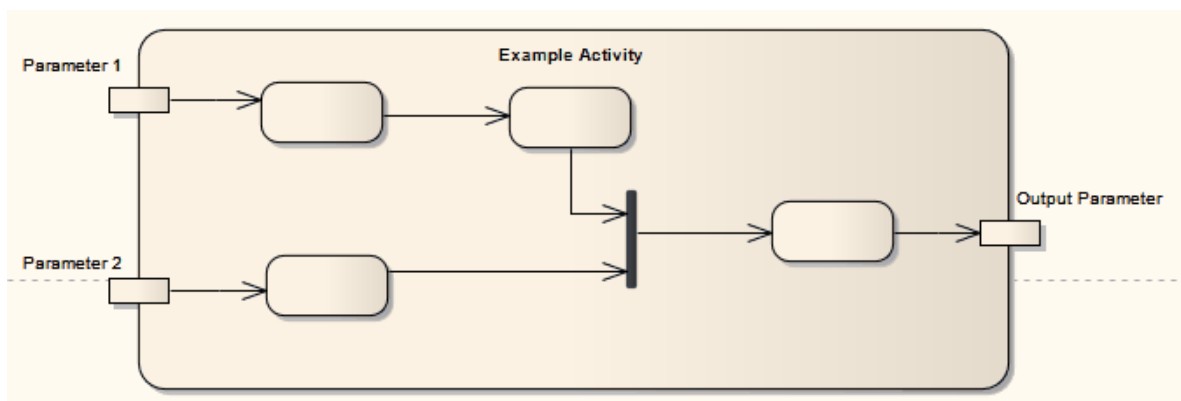
- [Activity](#)<sup>[1279]</sup>

#### 6.4.1.2.2 Activity Parameter Nodes

##### Description

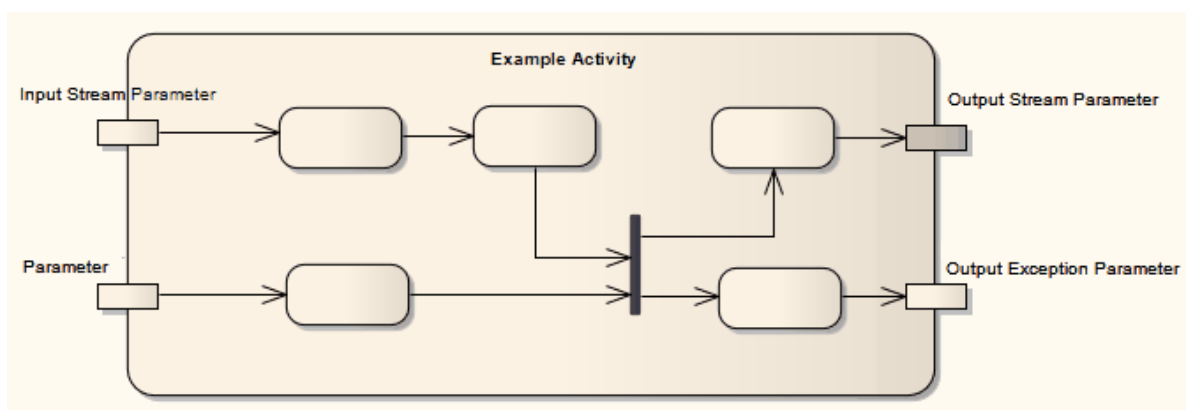
An Activity Parameter Node accepts input to an Activity or provides output from an Activity.

The following example depicts two entry parameters and one output parameter defined for the Activity.



**Define an Activity Parameter Node for an Activity**

Step	Action
1	Right-click on the element and select the <b>New Element   Activity Parameter</b> context menu option.
2	The Properties dialog displays, which prompts for the <b>Name</b> and other properties of the embedded element.
3	<p>To further define the new Activity Parameter, select the Parameter page of the Properties dialog and define:</p> <ul style="list-style-type: none"> <li>• Type</li> <li>• Default Value</li> <li>• Direction</li> <li>• Whether this is a fixed value</li> <li>• Multiplicity upper and lower bounds</li> <li>• Whether to allow duplicates and</li> <li>• Whether multiplicity is ordered</li> </ul> <p>Activity Parameter Nodes also have the <b>Exception</b> and <b>Stream</b> options:</p> <ul style="list-style-type: none"> <li>• <b>Exception</b> indicates that a parameter can emit a value at the exclusion of other outputs, usually because of some error</li> <li>• <b>Stream</b> indicates whether or not a parameter can accept or post values during the execution of the Activity</li> </ul>



**Learn more**

- [Activity Element](#)<sup>[1278]</sup>
- [Action Pin](#)<sup>[1277]</sup>

**OMG UML Specification:**

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 338) states:

*An activity parameter node is an object node for inputs and outputs to activities.*

*... Activity parameter nodes are object nodes at the beginning and end of flows that provide a means to accept inputs to an activity and provide outputs from the activity, through the activity parameters.*

*Activity parameters inherit support for streaming and exceptions from Parameter.*

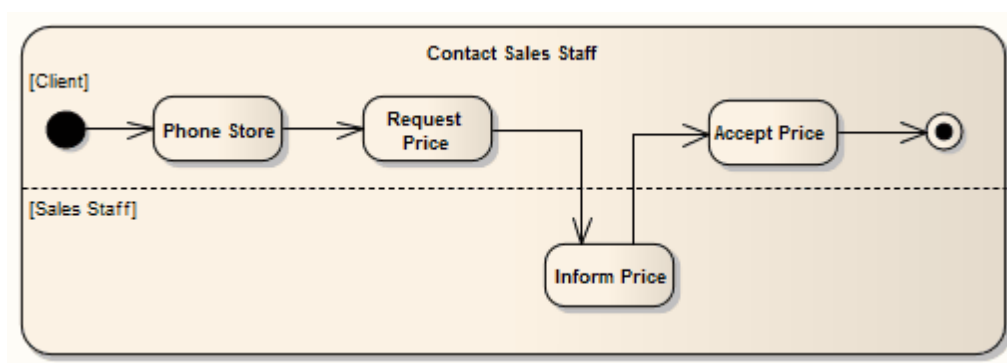
**6.4.1.2.3 Activity Partition****Description**

Enterprise Architect supports two types of *Activity Partition*:

- The Activity Partition *feature*, described in this topic, which is used to logically organize an Activity *element*
- The Activity Partition *element*, which is used to logically organize an Activity *diagram*

In effect, these are the same. They partition the Actions of the Activity without affecting the token flow, helping to structure the view or parts of the Activity.

An example of a feature-partitioned Activity is shown below:

**Define Partitions**

Step	Action
1	Right-click on the Activity element and select the <b>Advanced   Partition Activity</b> context menu option.

Step	Action
	The Activity Partitions dialog displays.
2	In the <b>Name</b> field, type the name of a partition. Click on the <b>Save</b> button.
3	Repeat step 2 for each partition to be created.

### Learn more

- [Activity Diagram](#)<sup>[1199]</sup>
- [Activity Element](#)<sup>[1279]</sup>
- [Activity Partition Element](#)<sup>[1325]</sup>

### OMG UML Specification

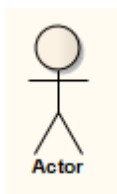
The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 341) states:

*Partitions divide the nodes and edges to constrain and show a view of the contained nodes. Partitions can share contents. They often correspond to organizational units in a business model. They may be used to allocate characteristics or resources among the nodes of an activity.*

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 341) also states:

*An activity partition is a kind of activity group for identifying actions that have some characteristic in common.*

### 6.4.1.3 Actor



### Description

An Actor is a user of the system; user can mean a human user, a machine, or even another system or subsystem in the model. Anything that interacts with the system from the outside or system boundary is termed an Actor. Actors are typically associated with Use Cases.

Actors can use the system through a graphical user interface, through a batch interface or through some other media. An Actor's interaction with a Use Case is documented in a Use Case scenario, which details the functions a system must provide to satisfy the user requirements.

Actors also represent the role of a user in Sequence Diagrams, where you can display them using rectangle

notation. Enterprise Architect supports a stereotyped Actor element for business modeling. The business modeling elements also represent Actors as stereotyped Objects.

#### Toolbox icon



#### Learn more

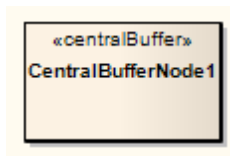
- [Sequence Diagrams](#)<sup>[1249]</sup>
- [Business Modeling](#)<sup>[1805]</sup>
- [Use Case](#)<sup>[1352]</sup>

#### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 584) states:

*An actor models a type of role played by an entity that interacts with the subject (e.g. by exchanging signals and data), but which is external to the subject. ... Actors may represent roles played by human users, external hardware, or other subjects. Note that an actor does not necessarily represent a specific physical entity but merely a particular facet (i.e., "role") of some entity that is relevant to the specification of its associated Use Cases. Thus, a single physical instance may play the role of several different actors and, conversely, a given actor may be played by multiple different instances.*

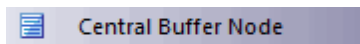
### 6.4.1.4 Central Buffer Node



#### Description

A Central Buffer Node is an object node for managing flows from multiple sources and destinations, represented in an Activity diagram. It acts as a buffer for multiple in-flows and out-flows from other object nodes, but does not connect directly to Actions.

#### Toolbox icon



#### Learn more

- [Activity Diagram](#)<sup>[1199]</sup>

#### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 352) states:

*A central buffer node is an object node for managing flows from multiple sources and destinations. ... A central buffer node accepts tokens from upstream object nodes and passes them along to downstream object nodes.*

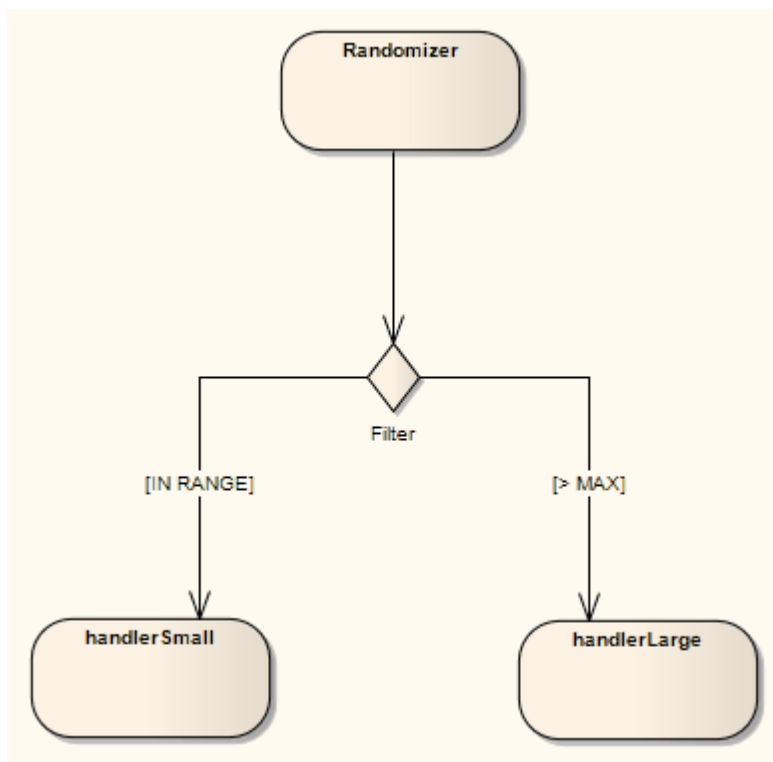
#### 6.4.1.5 Choice



##### Description

The Choice pseudo-state is used to compose complex transitional paths in, for example, a State Machine diagram, where the outgoing transition path is decided by dynamic, run-time conditions. The run-time conditions are determined by the actions performed by the State Machine on the path leading to the choice.

The following example depicts the Choice element. Upon reaching the Filter pseudo-state, a transition fires to the appropriate state based on the run-time value passed to the Filter. Very similar in form to a Junction pseudo-state, the Choice pseudo-state's distinction is in deciding transition paths at run-time.



##### Toolbox icon



### Learn more

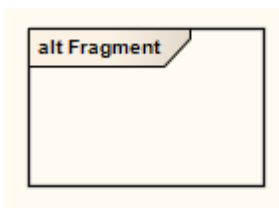
- [State Machine Diagram](#) <sup>[1203]</sup>
- [State Machine](#) <sup>[1329]</sup>
- [Pseudo-states](#) <sup>[1208]</sup>
- [Junction](#) <sup>[1319]</sup>

### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 538) states:

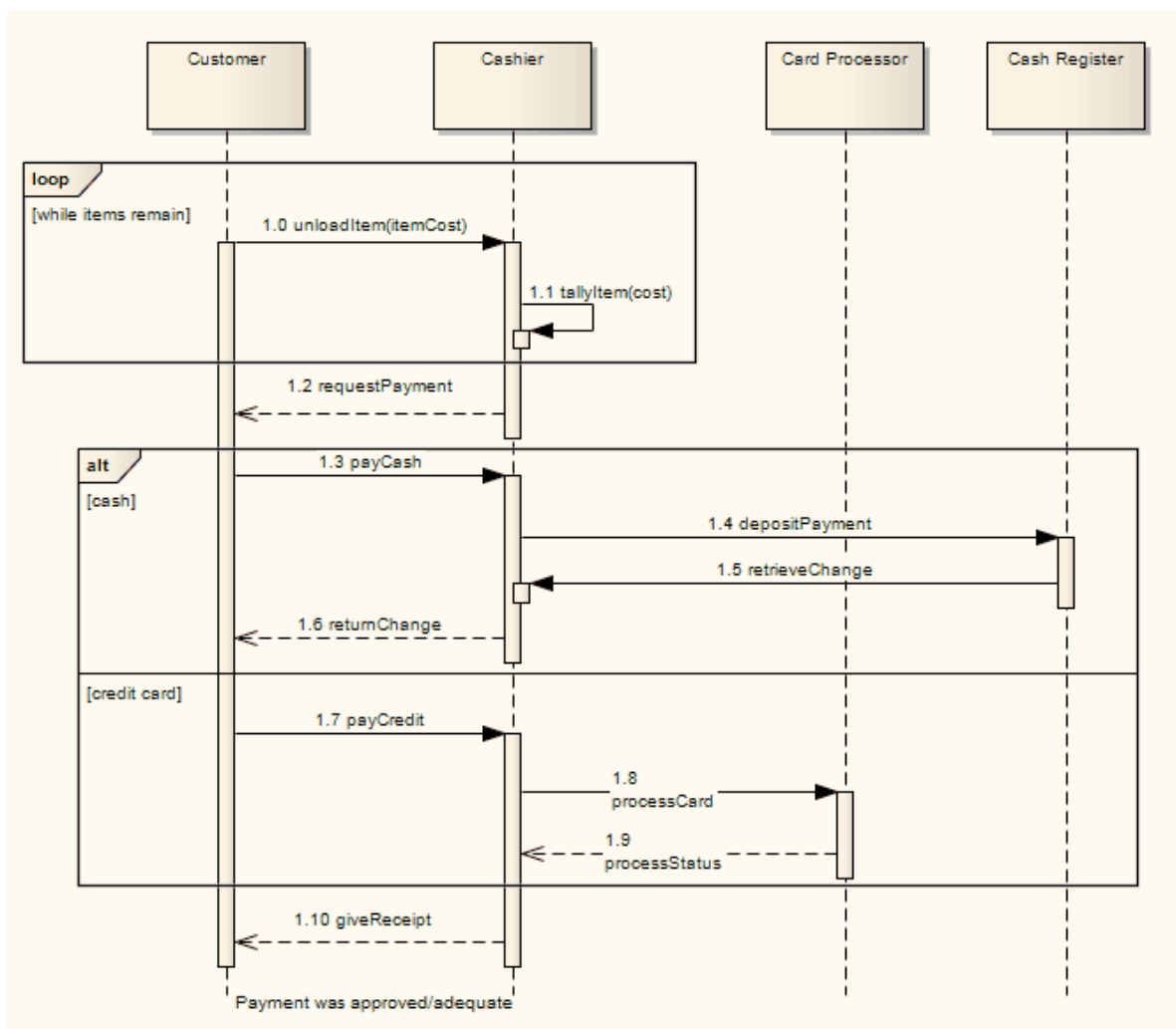
*...choice vertices which, when reached, result in the dynamic evaluation of the guards of the triggers of its outgoing transitions. This realizes a dynamic conditional branch. It enables splitting of transitions into multiple outgoing paths such that the decision on which path to take may be a function of the results of prior actions performed in the same run-to-completion step. If more than one of the guards evaluates to true, an arbitrary one is selected. If none of the guards evaluates to true, then the model is considered ill-formed. (To avoid this, it is recommended to define one outgoing transition with the predefined "else" guard for every choice vertex.) Choice vertices should be distinguished from static branch points that are based on junction points.*

### 6.4.1.6 Combined Fragment



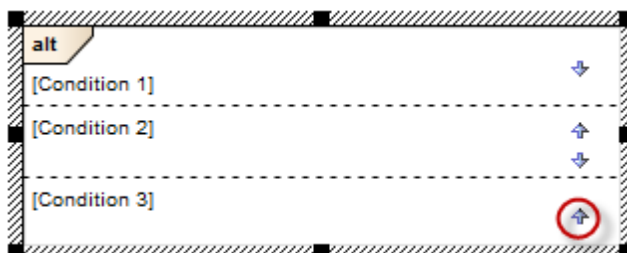
A **Combined Fragment** reflects one or more aspects of interaction (called **interaction operands**) controlled by an **interaction operator**, with corresponding boolean conditions known as **interaction constraints**. The Fragment displays as a transparent window, divided by horizontal lines for each operand.

This Sequence diagram illustrates the use of Combined Fragments in modeling a simplified purchasing process. A **loop fragment** represents iteration through an unknown number of items for purchase, after which the cashier requests payment. An **alternative fragment** represents the payment options, the fragment being divided to show the two operands *cash* and *credit card*. After the fragment completes its trace, the cashier gives a receipt to the customer, under the fulfilled condition that payment requirements were met.



The order of interaction fragment conditions can be changed directly on the diagram:


1. Select an interaction fragment with more than one condition defined; up and down arrows appear on the right hand side of the each condition.
2. Click on the appropriate arrow to change the order.



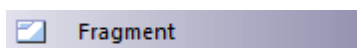




### Notes

- In order to select a Combined Fragment, you must click near the inside edge or drag a selection rectangle around the Fragment; this prevents accidental selection when moving connectors inside the Fragment
- Once contained within a Fragment or a Fragment Operand, Messages continue to be contained by it as they are moved up and down the diagram
- To move a Message out of a Fragment, or to a different position in the message sequence within the Fragment, press and hold ( **Alt** ) as you move it
- A Fragment on a Sequence diagram will resize when a Message within it is moved up or down, to continue to contain that message
- When you select an Interaction Fragment on a diagram, it shows one of two **element icons** (off the top right corner) that indicate whether or not the fragment contains Messages and that control how freely you can move the fragment and any Messages within and below the fragment
- To move a Combined Fragment **independently** of its contents, display the 'move freely' element icon (  ) and drag the element border
- Interaction Fragments inside a Combined Fragment operand cannot be moved outside the operand unless the fragment is in 'move freely' mode
- Moving an operand line moves any objects and Messages below that line down or up by the amount the operand line is moved
- Fragments containing other fragments resize when the contained fragment is resized, unless the fragment is in 'move freely' mode

### Toolbox icon



### Learn more

- [Sequence Diagram](#) <sup>[1249]</sup>
- [Create a Combined Fragment](#) <sup>[1290]</sup>
- [Interaction Operator](#) <sup>[1290]</sup>
- [Interaction Operand Condition and Message Behavior](#) <sup>[2490]</sup> (in Model Simulation)
- [Element Icons](#) <sup>[954]</sup>

### OMG UML Specification

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 467) states:

*A combined fragment defines an expression of interaction fragments. A combined fragment is defined by an interaction operator and corresponding interaction operands. Through the use of CombinedFragments the user will be able to describe a number of traces in a compact and concise manner.*

#### 6.4.1.6.1 Create a Combined Fragment

##### Create a Combined Fragment

Step	Action	See also
1	Drag the <i>Fragment</i> element onto the diagram from the Interaction Elements page of the Toolbox.	<a href="#">Combined Fragment</a> <sup>[1287]</sup>
2	In the <b>Type</b> field, click on the drop-down arrow and select one of the various types of interaction operator.	<a href="#">Interaction Operators</a> <sup>[1290]</sup>
3	In the <b>Condition</b> field, specify a condition or interaction constraint for each operand.	
4	A rectangular frame displays, partitioned by lines into segments for each operand.	
5	Adjust the frame to encompass the required event occurrences for each operand.	

##### Notes

- A message will always be contained within a fragment or a fragment operand when it is moved within it
- Fragments on Sequence diagrams will resize when a message is moved down to ensure that messages, once within a fragment, always remain within the fragment

#### 6.4.1.6.2 Interaction Operators

When creating **Combined Fragments**, you must apply an appropriate **interaction operator** to characterize the fragment. The following table provides guidance on the various operators, and their corresponding descriptions.

##### Interaction Operator

Operator	Action
<b>alt</b>	Divide up interaction fragments based on Boolean conditions.
<b>opt</b>	Enclose an optional fragment of interaction.

Operator	Action
<b>par</b>	Indicate that operands operate in parallel.
<b>loop</b>	Indicate that the operand repeats a number of times, as specified by interaction constraints.
<b>critical</b>	Indicate a sequence that cannot be interrupted by other processing.
<b>neg</b>	Assert that a fragment is invalid, and implies that all other interaction is valid.
<b>assert</b>	Specify the only valid fragment to occur. This operator is often enclosed within a <i>consider</i> or <i>ignore</i> operand.
<b>strict</b>	Indicate that the behaviors of the operands must be processed in strict sequence.
<b>seq</b>	Indicate that the Combined Fragment is weakly sequenced. This means that the ordering within operands is maintained, but the ordering between operands is undefined, so long as an event occurrence of the first operand precedes that of the second operand, if the event occurrences are on the same lifeline.
<b>ignore</b>	Indicate which messages should be ignored during execution, or can appear anywhere in the execution trace.
<b>consider</b>	Specify which messages should be considered in the trace. This is often used to specify the resulting event occurrences with the use of an <b>assert</b> operator.
<b>ref</b>	<p>Provide a reference to another diagram.</p> <p>The ref fragment is not created using the method described in the <i>Create a Combined Fragment</i> topic. To create a ref fragment, simply drag an existing diagram from the Project Browser onto the current diagram.</p>

#### Learn more

- [Combined Fragments](#)<sup>[1287]</sup>
- [Create a Combined Fragment](#)<sup>[1290]</sup>

#### OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 468-471*) states:

*The semantics of a CombinedFragment is dependent upon the interactionOperator as explained below.*

## Alternatives

The interactionOperator **alt** designates that the CombinedFragment represents a choice of behavior. At most one of the operands will be chosen. The chosen operand must have an explicit or implicit guard expression that evaluates to true at this point in the interaction. An implicit true guard is implied if the operand has no guard.

The set of traces that defines a choice is the union of the (guarded) traces of the operands.

An operand guarded by **else** designates a guard that is the negation of the disjunction of all other guards in the enclosing CombinedFragment.

If none of the operands has a guard that evaluates to true, none of the operands are executed and the remainder of the enclosing InteractionFragment is executed.

## Option

The interactionOperator **opt** designates that the CombinedFragment represents a choice of behavior where either the (sole) operand happens or nothing happens. An option is semantically equivalent to an alternative CombinedFragment where there is one operand with non-empty content and the second operand is empty.

## Break

The interactionOperator **break** designates that the CombinedFragment represents a breaking scenario in the sense that the operand is a scenario that is performed instead of the remainder of the enclosing InteractionFragment. A **break** operator with a guard is chosen when the guard is true and the rest of the enclosing Interaction Fragment is ignored. When the guard of the **break** operand is false, the **break** operand is ignored and the rest of the enclosing InteractionFragment is chosen. The choice between a **break** operand without a guard and the rest of the enclosing InteractionFragment is done non-deterministically.

A CombinedFragment with interactionOperator **break** should cover all Lifelines of the enclosing InteractionFragment.

## Parallel

The interactionOperator **par** designates that the CombinedFragment represents a parallel merge between the behaviors of the operands. The OccurrenceSpecifications of the different operands can be interleaved in any way as long as the ordering imposed by each operand as such is preserved.

A parallel merge defines a set of traces that describes all the ways that OccurrenceSpecifications of the operands may be interleaved without obstructing the order of the OccurrenceSpecifications within the operand.

## Weak Sequencing

The interactionOperator **seq** designates that the CombinedFragment represents a weak sequencing between the behaviors of the operands.

Weak sequencing is defined by the set of traces with these properties:

1. The ordering of OccurrenceSpecifications within each of the operands is maintained in the result.
2. OccurrenceSpecifications on different lifelines from different operands may come in any order.
3. OccurrenceSpecifications on the same lifeline from different operands are ordered such that an OccurrenceSpecification of the first operand comes before that of the second operand.

Thus weak sequencing reduces to a parallel merge when the operands are on disjunct sets of participants. Weak sequencing reduces to strict sequencing when the operands work on only one participant.

## Strict Sequencing

The interactionOperator **strict** designates that the CombinedFragment represents a strict sequencing between the behaviors of the operands. The semantics of strict sequencing defines a strict ordering of the operands on the first level within the CombinedIFragment with interactionOperator **strict**. Therefore OccurrenceSpecifications within contained CombinedFragment will not directly be compared with other OccurrenceSpecifications of the enclosing CombinedFragment.

## Negative

The interactionOperator **neg** designates that the CombinedFragment represents traces that are defined to be invalid.

The set of traces that defined a CombinedFragment with interactionOperator negative is equal to the set of traces given by its (sole) operand, only that this set is a set of invalid rather than valid traces. All InteractionFragments that are different from Negative are considered positive meaning that they describe traces that are valid and should be possible.

### Critical Region

The interactionOperator **critical** designates that the CombinedFragment represents a critical region. A critical region means that the traces of the region cannot be interleaved by other OccurrenceSpecifications (on those Lifelines covered by the region). This means that the region is treated atomically by the enclosing fragment when determining the set of valid traces. Even though enclosing CombinedFragments may imply that some OccurrenceSpecifications may interleave into the region, such as with **par**-operator, this is prevented by defining a region.

Thus the set of traces of enclosing constructs are restricted by critical regions.

### Ignore / Consider

(p. 473) The interactionOperator **ignore** designates that there are some message types that are not shown within this combined fragment. These message types can be considered insignificant and are implicitly ignored if they appear in a corresponding execution. Alternatively one can understand **ignore** to mean that the messages that are ignored can appear anywhere in the traces.

Conversely the interactionOperator **consider** designates which messages should be considered within this CombinedFragment. This is equivalent to defining every other message to be ignored.

### Assertion

The interactionOperator **assert** designates that the CombinedFragment represents an assertion. The sequences of the operand of the assertion are the only valid continuations. All other continuations result in an invalid trace. Assertions are often combined with Ignore or Consider.

### Loop

The interactionOperator **loop** designates that the CombinedFragment represents a loop. The **loop** operand will be repeated a number of times.

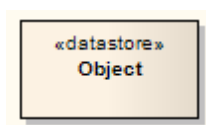
The Guard may include a lower and an upper number of iterations of the loop as well as a Boolean expression. The semantics is such that a loop will iterate minimum the 'minint' number of times (given by the iteration expression in the guard) and at most the 'maxint' number of times. After the minimum number of iterations have executed, and the boolean expression is false the loop will terminate. The loop construct represent a recursive application of the **seq** operator where the **loop** operand is sequenced after the result of earlier iterations.

### The Semantics of Gates

The gates of a CombinedFragment represent the syntactic interface between the CombinedFragment and its surroundings, which means the interface towards other InteractionFragments.

The only purpose of gates is to define the source and the target of messages.

#### 6.4.1.7 Datastore



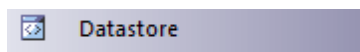
#### Description

A Datastore is an element used to define permanently stored data. A token of data that enters into a Datastore is stored permanently, updating tokens for data that already exists. A token of data that comes out of a Datastore is a copy of the original data.

Use Object Flow connectors to connect elements (such as Activities) to Datastores, as values and information are being passed between nodes. Selection and transformation behavior, together composing a sort of query, can be specified as to the nature of data access. For instance, selection behavior determines which objects are affected by the connection to the Datastore. Transformation behavior might then further specify the value of an attribute pertaining to a selected object.

To define the behavior of access to a Datastore, attach a note to the Object Flow connector. To do this, right-click on the Object Flow and select the **Attach Note or Constraint** context menu option. A dialog indicates other flows in the Activity diagram, to which you can attach the note (if the behavior applies to multiple flows). To comply with UML 2, preface behavior with the notation «selection» or «transformation».

#### Toolbox icon



#### Learn more

- [Activity Diagram](#) <sup>[1199]</sup>
- [Activity](#) <sup>[1279]</sup>
- [SysML Activity Toolbox](#) <sup>[2295]</sup>
- [Object Flow](#) <sup>[1435]</sup>

#### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 360) states:

*A data store node is a central buffer node for non-transient information... A data store keeps all tokens that enter it, copying them when they are chosen to move downstream. Incoming tokens containing a particular object replace any tokens in the object node containing that object.*

### 6.4.1.8 Decision

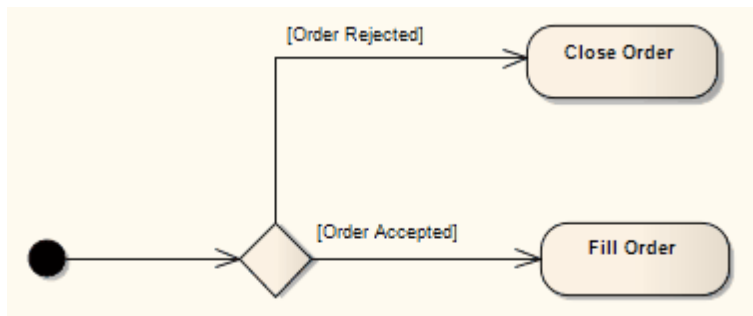


#### Description

A Decision is an element of an Activity diagram or Interaction Overview diagram that indicates a point of conditional progression: if a condition is true, then processing continues one way; if not, then another.

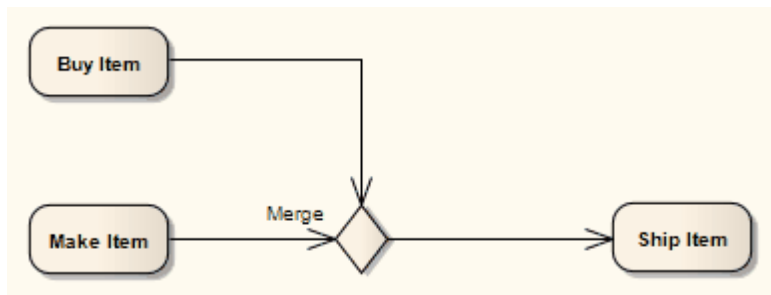
This can also be used as a Merge node in that multiple alternative flows can be merged (but not synchronized) to form one flow. The following examples show both of these modes of using the decision element.

*Used as a decision:*



See *UML Superstructure Specification, v2.1.1, figure 12.77, p. 363.*

*Used as a merge:*



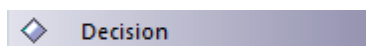
See *UML Superstructure Specification, v2.1.1, figure 12.106, p. 388.*

### Notes

- Moving a diagram generally does not affect the location of elements in packages; if you move a diagram out of one package into another, all the elements in the diagram remain in the original package

However, Decision elements are used only within one diagram, have no meaning outside that diagram, and are never re-used in any other diagram; therefore, if you move a diagram containing these elements, they are moved to the new parent package with the diagram

### Toolbox icon



### Learn more

- [Activity Diagram](#) <sup>[1199]</sup>
- [Interaction Overview Diagram](#) <sup>[1262]</sup>
- [Merge Node](#) <sup>[1322]</sup>

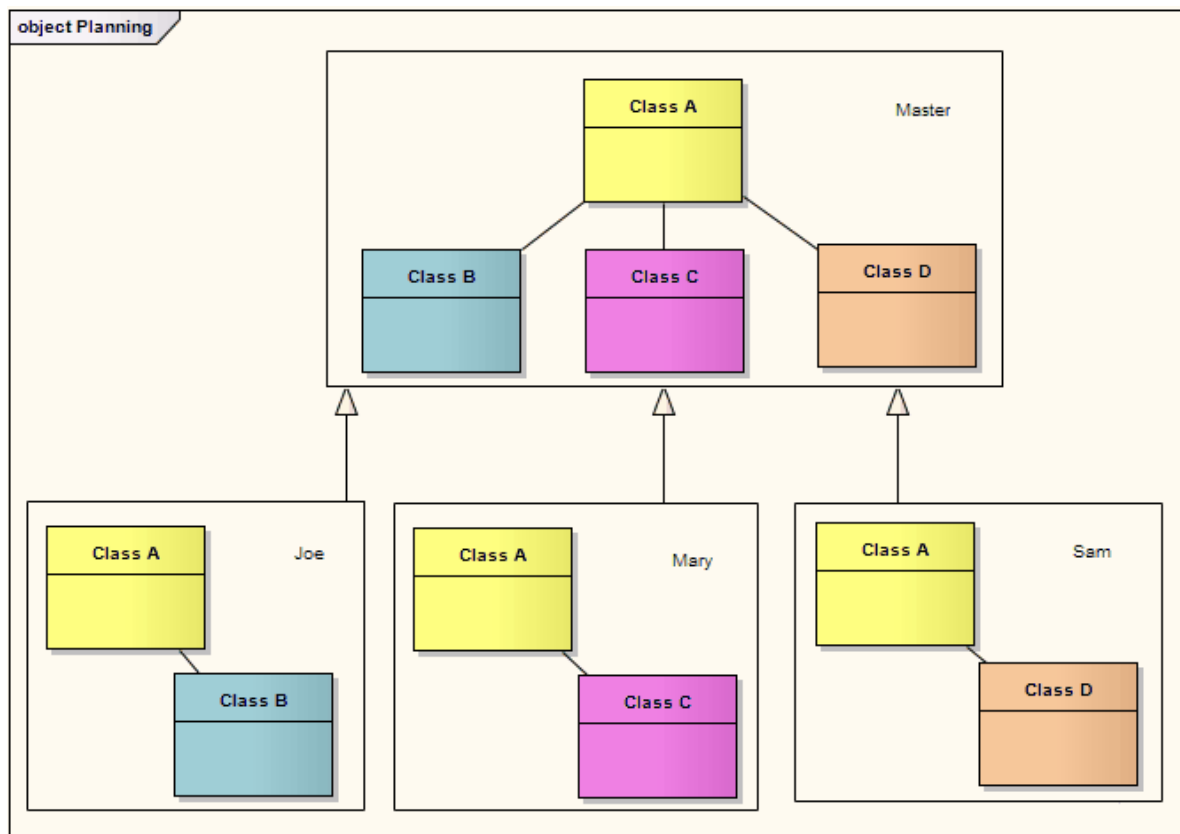
**OMG UML Specification:**

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 361 (Decision symbol)*) states:

*A decision node is a control node that chooses between outgoing flows. A decision node has one incoming edge and multiple outgoing activity edges.*

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 387 (Merge symbol)*) also states:

*A merge node is a control node that brings together multiple alternate flows. It is not used to synchronize concurrent flows but to accept one among several alternate flows... A merge node has multiple incoming edges and a single outgoing edge.*

**6.4.1.9 Diagram Frame**

A **Diagram Frame** element is a rendition of a diagram dropped from the Project Browser into another diagram. It is a type of **Combined Fragment** with the Interaction Operator **ref**. However, it can be created on any type of diagram, and is not created in the same way as other Combined Fragments.

When you drop the diagram from the Project Browser onto the open diagram, a dialog shows providing the following options:

- **Diagram Frame** - a Diagram Frame is inserted into the diagram, containing an image of the dropped diagram
- **Diagram Reference** - an empty frame is inserted with the name of the dropped diagram in the frame label
- **Hyperlink** - a diagram icon is inserted with no frame, and with the parent package and diagram name



next to it

In all three cases, the object acts as a hyperlink to the real referenced diagram. You can also define properties for the objects, as for other elements, by right-clicking on the object and selecting the element **Properties** context menu option.

### Diagram Frame Appearance

You can change the appearance of a Diagram **Frame**, as for other elements, but the available options are tailored for this element type. If you right-click on the frame and select the **Appearance | Diagram Frame Appearance** option, a sub-menu displays with these options:

- **Normal** - the default appearance of a visible rectangular frame with a visible frame label; you can use this option to reset the appearance after using one of the other options
- **Boundary** - hides the frame label of the Diagram Frame
- **Boundary With Name** - hides the border of the frame **label**
- **Name Only** - hides the border of the Diagram Frame and frame label, leaving the text only
- **Hidden** - hides the border and text of the Diagram Frame

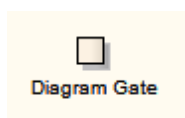
### Notes

- You can change the size of all three objects, but you cannot reduce a Diagram Frame to less than the size of the enclosed diagram
- You cannot change the diagram within a Diagram Frame; to edit the diagram, double-click within the frame and edit the original diagram
- The Diagram Frame **element** is not the same as the diagram frame **border** that you can set (using the Diagram Frames panel on the Diagram page of the Options dialog) on images of diagrams that you print out, copy to file or paste into other tools; it is possible, but not usual, to paste the diagram image from the clipboard into another Enterprise Architect diagram, in which case the image initially looks the same as the Diagram Frame element, but element options do not function on this image

### Learn more

- [Combined Fragment](#)<sup>[1287]</sup>
- [Interaction Operator: ref](#)<sup>[1291]</sup>
- [Diagram Options](#)<sup>[608]</sup>

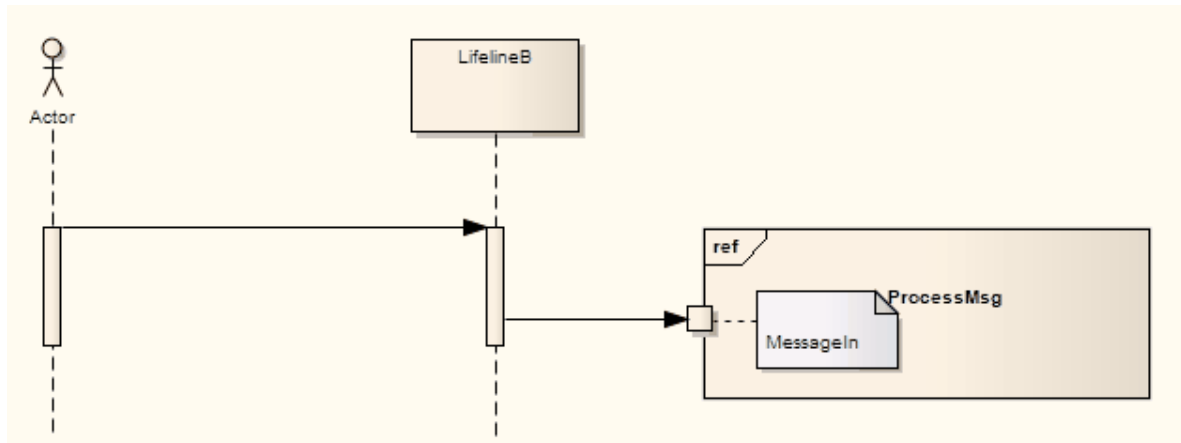
## 6.4.1.10 Diagram Gate



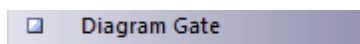
### Description

A Diagram Gate is a simple graphical way to indicate the point at which messages can be transmitted into and out of interaction fragments. A fragment might be required to receive or deliver a message; internally, an ordered message reflects this requirement, with a gate indicated on the boundary of the fragment's frame.

Any external messages 'synching' with this internal message must correspond appropriately. Gates can appear on Interaction diagrams (Sequence, Timing, Communication or Interaction Overview), interaction occurrences and combined fragments (to specify the expression).



#### Toolbox icon



#### Learn more

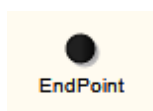
- [Sequence Diagram](#) <sup>[1249]</sup>
- [Timing Diagram](#) <sup>[1225]</sup>
- [Communication Diagram](#) <sup>[1259]</sup>
- [Interaction Overview Diagram](#) <sup>[1262]</sup>
- [Interaction Occurrence](#) <sup>[1317]</sup>
- [Combined Fragment](#) <sup>[1287]</sup>

#### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 480) states:

*A Gate is a connection point for relating a Message outside an InteractionFragment with a Message inside the InteractionFragment ... Gates are connected through Messages. A Gate is actually a representative of an OccurrenceSpecification that is not in the same scope as the Gate. Gates play different roles: we have formal gates on Interactions, actual gates on InteractionUses, expression gates on CombinedFragments.*

#### 6.4.1.11 Endpoint

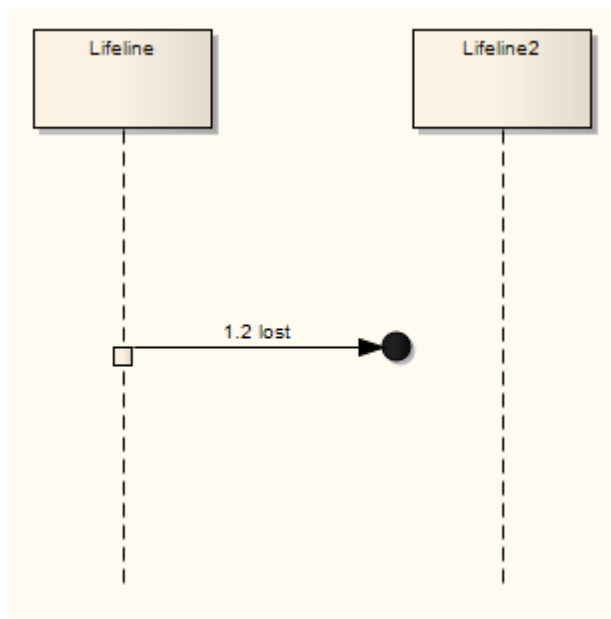


### Description

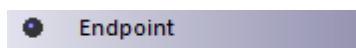
An Endpoint is used in Interaction diagrams (Sequence, Timing, Communication or Interaction Overview) to reflect a lost or found Message in sequence. To model this, drag an Endpoint element onto the workspace.

With Sequence diagrams, drag a Message from the appropriate Lifeline to the Endpoint. With Timing diagrams, the Message connecting the Lifeline to the Endpoint requires some timing specifications to draw the connection.

The following example depicts a lost Message in a Sequence diagram.



### Toolbox icon



### Learn more

- [Sequence Diagram](#) <sup>[1249]</sup>
- [Timing Diagram](#) <sup>[1225]</sup>
- [Communication Diagram](#) <sup>[1259]</sup>
- [Interaction Overview Diagram](#) <sup>[1262]</sup>

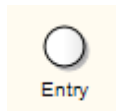
### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 492) states:

*A lost message is a message where the sending event occurrence is known, but there is no receiving event occurrence. We interpret this to be because the message never reached its destination.*

*A found message is a message where the receiving event occurrence is known, but there is no (known) sending event occurrence. We interpret this to be because the origin of the message is outside the scope of the description. This may, for example, be noise or other activity that we do not want to describe in detail.*

#### 6.4.1.12 Entry Point



##### Description

Entry Point pseudo-states are used to define the beginning of a State Machine. An Entry Point exists for each region, directing the initial concurrent state configuration.

##### Toolbox icon



##### Learn more

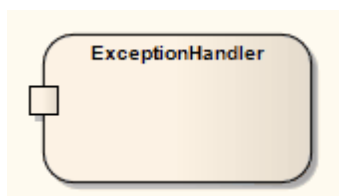
- [State Machine Diagram](#) <sup>[1203]</sup>
- [Pseudo-states](#) <sup>[1208]</sup>

##### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 471) states:

*An entry point pseudostate is an entry point of a state machine or composite state. In each region of the state machine or composite state it has a single transition to a vertex within the same region.*

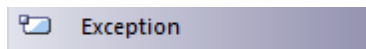
#### 6.4.1.13 Exception



##### Description

The Exception Handler element defines the group of operations to carry out when an exception occurs. In an Activity diagram, the protected element can contain a set of operations and is connected to the exception handler via an Interrupt Flow connector. Any defined error contained within an element's parts can trigger the flow to move to an exception.

### Toolbox icon



### Learn more

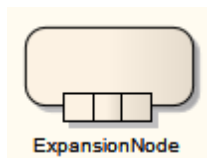
- [Activity Diagram](#) <sup>[1199]</sup>
- [Interrupt Flow](#) <sup>[1415]</sup>

### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 364) states:

*An exception handler is an element that specifies a body to execute in case the specified exception occurs during the execution of the protected node.*

#### 6.4.1.14 Expansion Node



### Description

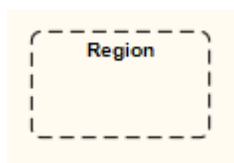
Representing an Action or an Activity as an *Expansion Node* is a shorthand notation to indicate that the Action/Activity comprises an Expansion Region.

To specify an Action or Activity as an Expansion Node, right-click on the Action and select the **New Element | Expansion Node** context menu option.

### Learn more

- [Action](#) <sup>[1266]</sup>
- [Activity](#) <sup>[1279]</sup>
- [Expansion Region](#) <sup>[1301]</sup>

#### 6.4.1.15 Expansion Region



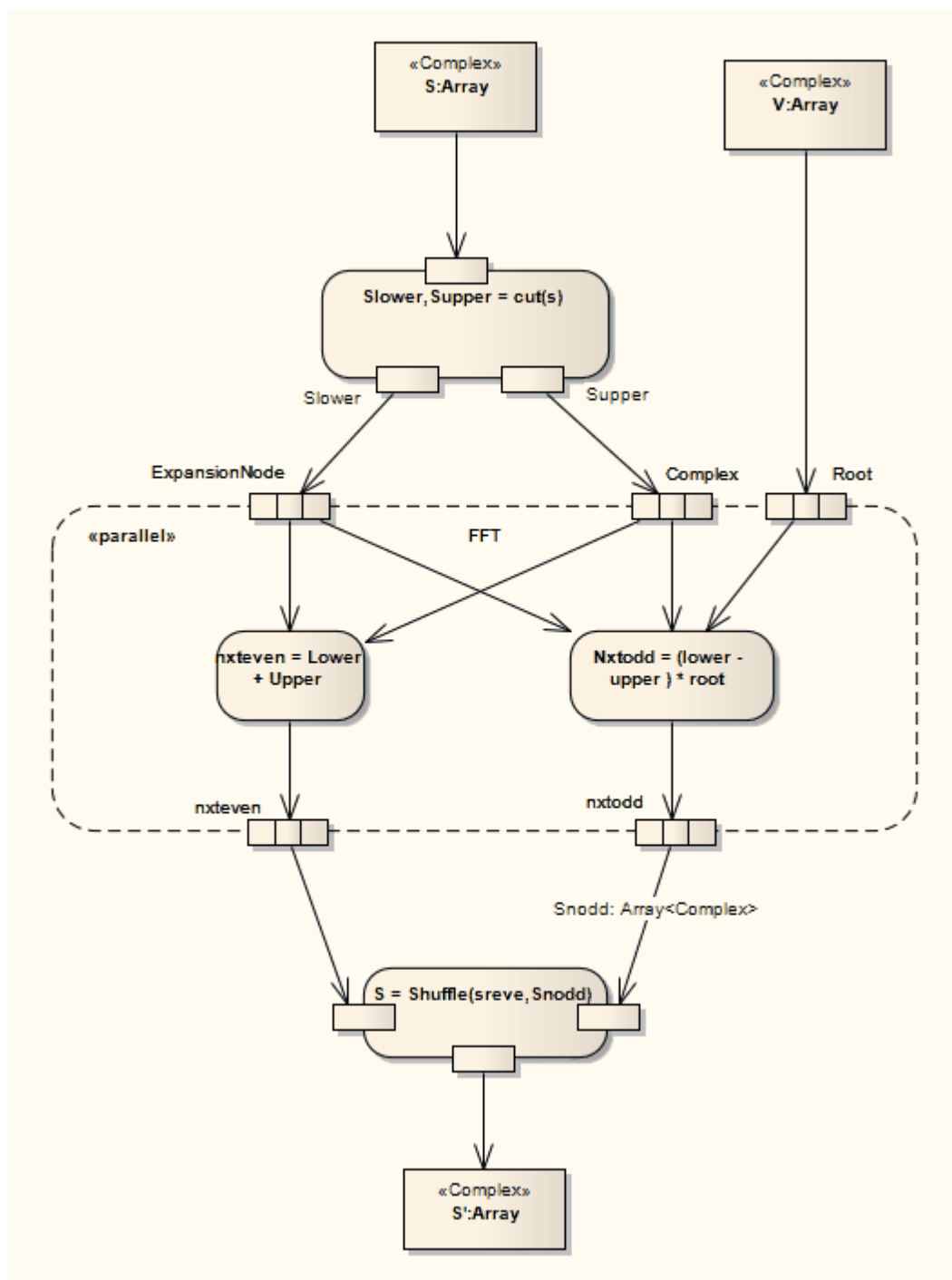
### Description

You create an Expansion Region as one variant of a Region (the other is an Interruptible Activity Region).

On an Activity diagram, an Expansion Region surrounds a process to be imposed multiple times on the incoming data, once for every element in the input collection. If there are multiple inputs, the collection sizes must match, and the elements within each collection must be of the same type. Similarly, any outputs must be in the form of a collection matching the size of the inputs.

The concurrency of the Expansion Region's multiple executions can be specified as type parallel, iterative, or stream. Parallel reflects that the elements in the incoming collections can be processed at the same time or overlapping, whereas an iterative concurrency type specifies that execution must occur sequentially. A stream-type Expansion Region indicates that the input and output come in and exit as streams, and that the Expansion Region's process must have some method to support streams.

To modify the mode of an Expansion Region, right-click on it and select the **Properties** context menu option, then select the Advanced page.



See *UML Superstructure Specification, v2.1.1*, figure 12.87, p. 372.

#### Toolbox icon



### Learn more

- [Activity Diagram](#) <sup>[1199]</sup>
- [Region](#) <sup>[1328]</sup>
- [Interruptible Activity Region](#) <sup>[1316]</sup>
- [Expansion Node](#) <sup>[1301]</sup>

### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 367) states:

*An expansion region is a structured activity region that executes multiple times corresponding to elements of an input collection.*

## 6.4.1.16 Exit Point



### Description

Exit Points are used in State Machine elements and State Machine diagrams to denote the point where the machine is exited and the transition sourcing this exit point, for State Machine elements, is triggered. Exit points are a type of pseudo-state used in the State Machine diagram.

### Toolbox icon



### Learn more

- [State Machines](#) <sup>[1203]</sup> (Diagrams)
- [State](#) <sup>[1329]</sup>
- [State Machine](#) <sup>[1338]</sup>
- [Pseudo-states](#) <sup>[1208]</sup>

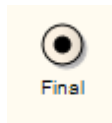
### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification*, v2.1.1 p. 538) states:

*An exit point pseudostate is an exit point of a state machine or composite state. Entering an exit point within any region of the composite state or state machine referenced by a submachine state implies the exit of this composite state or submachine state and the triggering of the transition that has this exit point as source in the state machine enclosing the submachine or composite state.*



### 6.4.1.17 Final

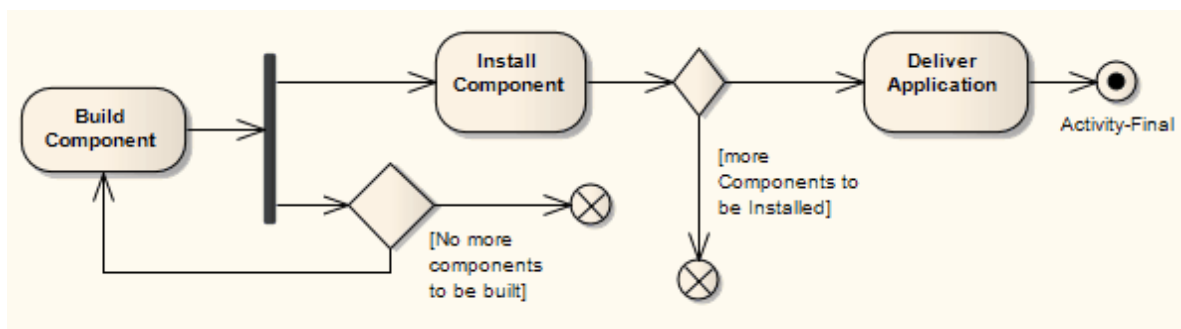


#### Description

There are two nodes used to define a Final state in an Activity, both defined in UML 2.4.1 as of type Final Node. The Activity Final element, shown above, indicates the completion of an Activity; upon reaching the Final, all execution in the Activity diagram is aborted. The other type of final node, Flow Final, depicts an exit from the system that has no effect on other executing flows in the Activity.

The following example illustrates the development of an application. The process comes to a Flow Final node when there are no more components to be built; note that the Fork element indicates a concurrent process with the building of new components and installation of completed components. The Flow Final terminates only the sub-process building components. Similarly, only those tokens entering the decision branch for the installation of further components terminate with the connecting Flow Final (that is, stop installing this component, but keep on installing other components). It is only after the Deliver Application activity is completed, after the control flow reaches the Final node, that all flows stop.

The node that initiates a flow is the Initial node.



See *UML Superstructure Specification, v2.1.1, figure 12.91, p. 374.*

#### Notes

- Moving a diagram generally does not affect the location of elements in packages; if you move a diagram out of one package into another, all the elements in the diagram remain in the original package

However, Final elements are used only within one diagram, have no meaning outside that diagram, and are never re-used in any other diagram; therefore, if you move a diagram containing these elements, they are moved to the new parent package with the diagram

#### Toolbox icon



Final

### Learn more

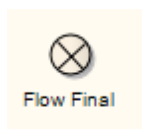
- [Activity Diagram](#) <sup>[1199]</sup>
- [Activity](#) <sup>[1279]</sup>
- [Flow Final](#) <sup>[1306]</sup>
- [Fork](#) <sup>[1309]</sup>
- [Initial](#) <sup>[1313]</sup>

### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 332*) states:

*An activity may have more than one activity final node. The first one reached stops all flows in the activity.*

#### 6.4.1.18 Flow Final

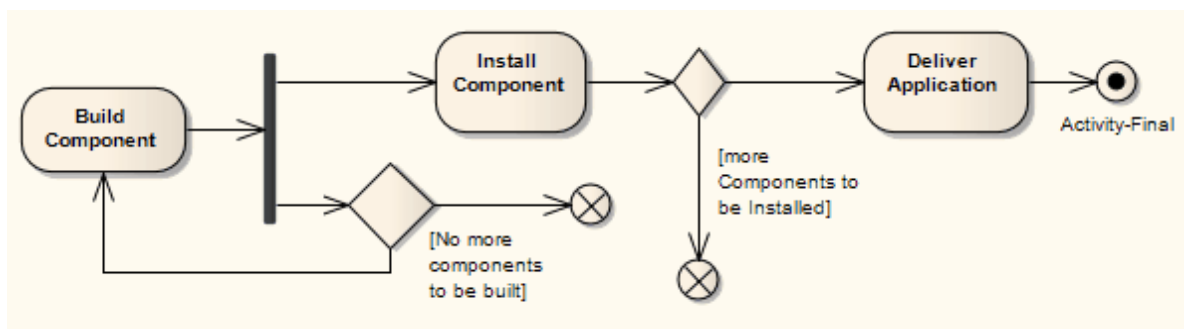


Flow Final

### Description

There are two nodes used to define a final state in an Activity, both defined in UML 2.4.1 as of type *Final Node*. The Flow Final element depicts an exit from the system, as opposed to the Activity Final, which represents the completion of the Activity. Only the flow entering the Flow Final node exits the Activity; other flows continue undisturbed.

The following example Activity Diagram illustrates the development of an application. The process comes to a Flow Final node when there are no more components to be built; note that the Fork element indicates a concurrent process with the building of new components and installation of completed components. The Flow Final terminates only the sub-process building components. Similarly, only those tokens entering the decision branch for the installation of further components terminate with the connecting Flow Final (that is, stop installing this component, but keep on installing other components). It is only after the Deliver Application activity is completed, after the control flow reaches the Final node, that all flows stop.



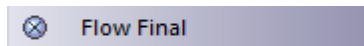
See *UML Superstructure Specification, v2.1.1, figure 12.91, p. 374*.

### Notes

- Moving a diagram generally does not affect the location of elements in packages: if you move a diagram out of one package into another, all the elements in the diagram remain in the original package

However, Flow Final elements are used only within one diagram, have no meaning outside that diagram, and are never re-used in any other diagram; therefore, if you move a diagram containing these elements, they are moved to the new parent package with the diagram

### Toolbox icon



### Learn more

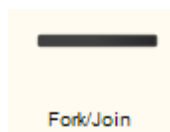
- [Activity Diagram](#) <sup>[1199]</sup>
- [Activity Final](#) <sup>[1305]</sup>
- [Fork](#) <sup>[1309]</sup>

### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 375*) states:

*A flow final destroys all tokens that arrive at it. It has no effect on other flows in the activity.*

#### 6.4.1.19 Fork/Join



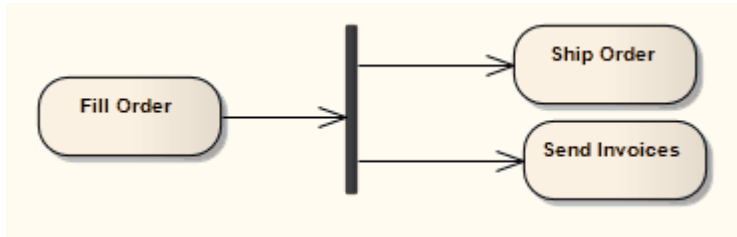
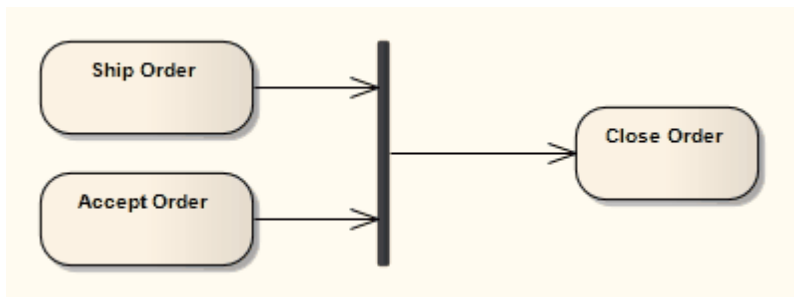
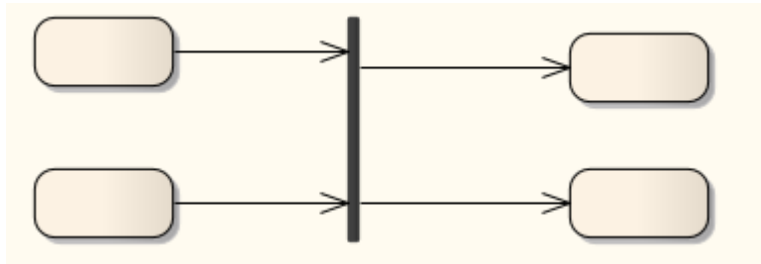
### Description

The *Fork/Join* elements have different modes of use, as follows:

- To fork or split the flow into a number of concurrent flows
- To join the flow of a number of concurrent flows
- To both join and fork a number of incoming flows to a number of outgoing flows

These elements are used in both Activity and State Machine diagrams. With respect to State Machine diagrams, Forks and Joins are used as pseudo-states. Other pseudo-states include history states, entry points and exit points. Forks are used to split an incoming transition into concurrent multiple transitions leading to different target states. Joins are used to merge concurrent multiple transitions into a single transition leading to a single target. They are semantic inverses. To learn more about these individual elements see their specific topics.

#### Example Diagrams:

Description	Diagram
<b>Fork or split the flow into a number of concurrent flows</b>	 A diagram illustrating a fork. A rounded rectangle labeled 'Fill Order' has an arrow pointing to a thick vertical line (the fork). From this line, two arrows branch out to two separate rounded rectangles labeled 'Ship Order' and 'Send Invoices'.
<b>Join the flow of a number of concurrent flows</b>	 A diagram illustrating a join. Two rounded rectangles labeled 'Ship Order' and 'Accept Order' have arrows pointing to a thick vertical line (the join). From this line, a single arrow points to a rounded rectangle labeled 'Close Order'.
<b>Join and Fork a number of incoming flows to a number of outgoing flows</b>	 A diagram illustrating a join and fork. Two rounded rectangles on the left have arrows pointing to a thick vertical line (the join). From this line, two arrows branch out to two separate rounded rectangles on the right.

#### Toolbox icon



#### Learn more

- [Activity Diagram](#) <sup>[1199]</sup>
- [State Machine Diagram](#) <sup>[1203]</sup>
- [Pseudo-state](#) <sup>[1208]</sup>
- [Fork](#) <sup>[1309]</sup>
- [Join](#) <sup>[1310]</sup>

#### **OMG UML Specification:**

##### **Fork**

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 376) states:

*A fork node is a control node that splits a flow into multiple concurrent flows... A fork node has one incoming edge and multiple outgoing edges.*

##### **Join**

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 381-382) states:

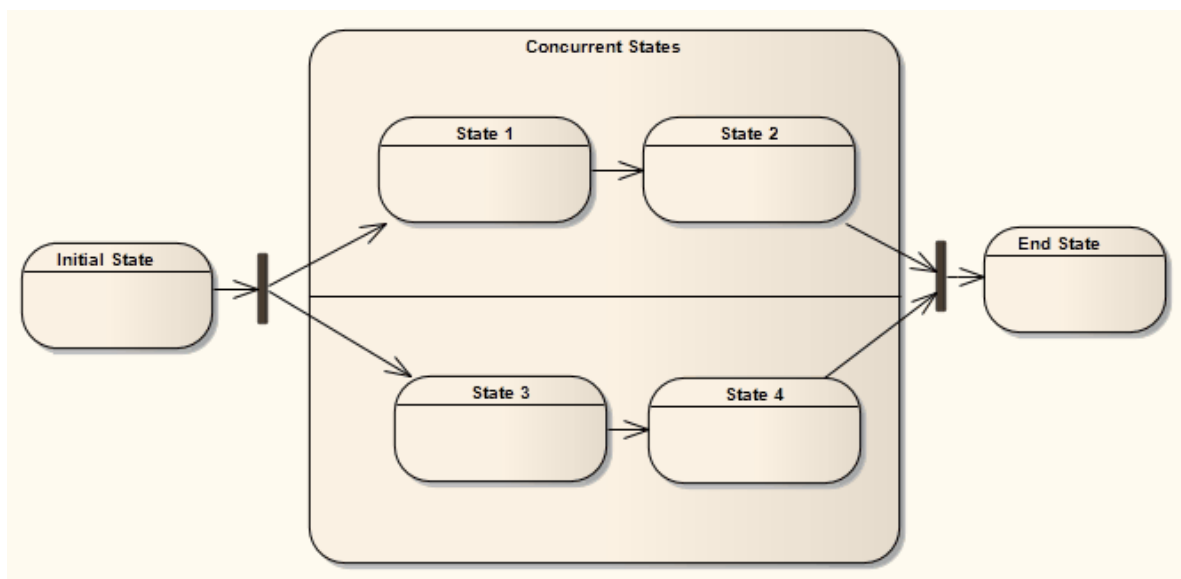
*A join node is a control node that synchronizes multiple flows... A join node has multiple incoming edges and one outgoing edge.*

#### **6.4.1.19.1 Fork**



#### **Description**

The Fork element is used in both Activity and State Machine diagrams. With respect to State Machine diagrams, a Fork pseudo-state signifies that its incoming transition comes from a single state, and it has multiple outgoing transitions. These transitions must occur concurrently, requiring the use of concurrent regions, as depicted here in the Composite State. Unlike Choice or Junction pseudo-states, Forks must not have triggers or guards. This diagram demonstrates a Fork pseudo-state dividing into two concurrent regions, which then return to the End State via the Join pseudo-state.



#### Learn more

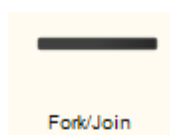
- [Activity Diagram](#) <sup>[1199]</sup>
- [State Machine Diagram](#) <sup>[1203]</sup>
- [Pseudo-state](#) <sup>[1208]</sup>
- [States](#) <sup>[1329]</sup>
- [Choice](#) <sup>[1286]</sup>
- [Junction](#) <sup>[1319]</sup>
- [Join](#) <sup>[1310]</sup>
- [Region](#) <sup>[1328]</sup>

#### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 538*) states:

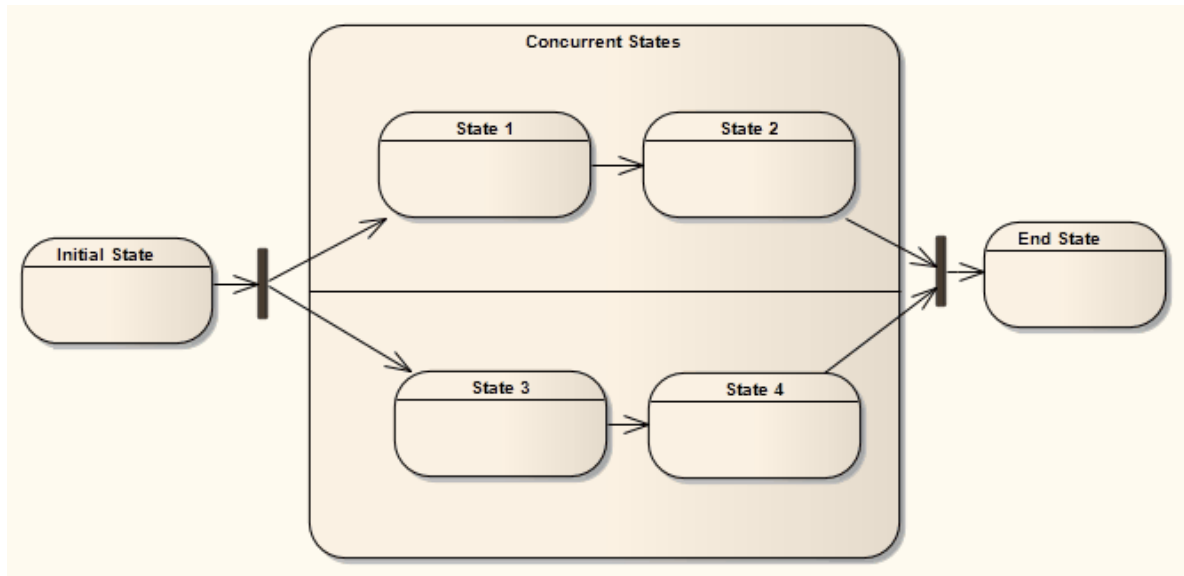
*Fork vertices serve to split an incoming transition into two or more transitions terminating on orthogonal target vertices (i.e. vertices in different regions of a composite state). The segments outgoing from a fork vertex must not have guards or triggers.*

#### 6.4.1.19.2 Join



#### Description

The Join element is used by Activity and State Machine diagrams. The example illustrates a Join transition between Activities. With respect to State Machine diagrams, a Join pseudo-state indicates multiple States concurrently transitioning into the Join and onto a single State. Unlike Choice or Junction pseudo-states, Joins must not have triggers or guards. The following diagram demonstrates a Fork pseudo-state dividing into two concurrent Regions, which then return to the End State via the Join.



#### Learn more

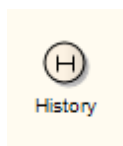
- [Activity Diagram](#) <sup>[1199]</sup>
- [State Machine Diagram](#) <sup>[1203]</sup>
- [Pseudo-state](#) <sup>[1208]</sup>
- [States](#) <sup>[1329]</sup>
- [Choice](#) <sup>[1286]</sup>
- [Junction](#) <sup>[1319]</sup>
- [Fork](#) <sup>[1309]</sup>
- [Region](#) <sup>[1328]</sup>

#### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 538*) states:

*Join vertices serve to merge several transitions emanating from source vertices in different orthogonal regions. The transitions entering a join vertex cannot have guards or triggers.*

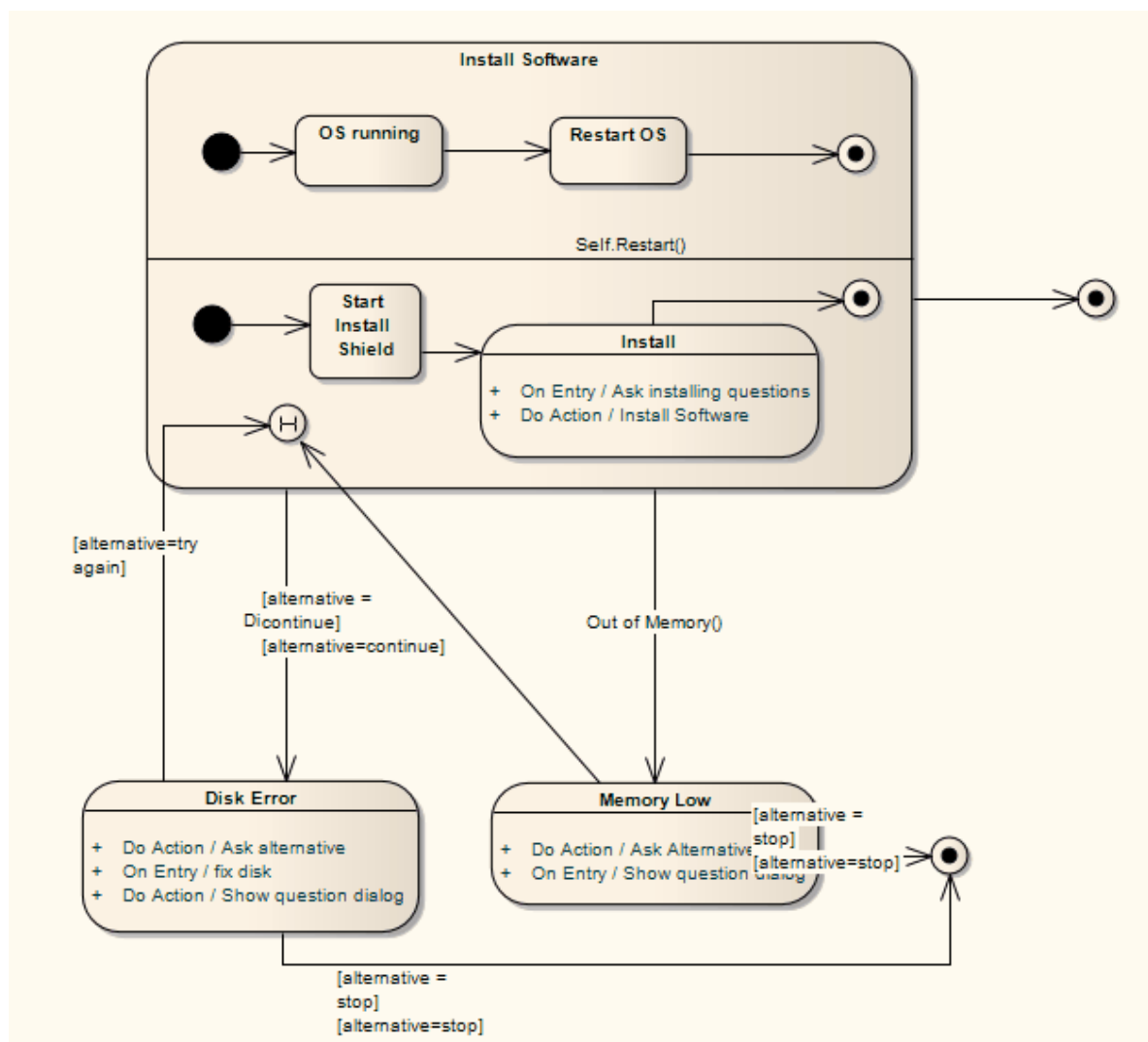
#### 6.4.1.20 History



**Description**

There are two types of History pseudo-state defined in UML: shallow and deep history. A shallow History sub-state is used to represent the most recently active sub-state of a Composite State; this pseudo-state does not recurse into this sub-state's active configuration, should one exist. A single connector can be used to depict the default shallow History state, in case the Composite State has never been entered.

A deep History sub-state, in contrast, reflects the most recent active configuration of the Composite State. This includes active sub-states of all regions, and recurses into those sub-states' active sub-states, should they exist. At most one deep history and one shallow history can dwell within a composite state. You can reassign a shallow History substate as a deep History substate using the Advanced element context menu.

**Toolbox icon**

History



### Learn more

- [Pseudo-state](#)<sup>[1208]</sup>
- [Composite State](#)<sup>[1330]</sup>

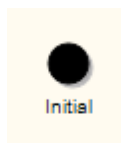
### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 537) states:

... *deepHistory* represents the most recent active configuration of the composite state that directly contains this pseudostate (e.g., the state configuration that was active when the composite state was last exited). A composite state can have at most one deep history vertex. At most one transition may originate from the history connector to the default deep history state. This transition is taken in case the composite state had never been active before. Entry actions of states entered on the path to the state represented by a deep history are performed.

... *shallowHistory* represents the most recent active substate of its containing state (but not the sub states of that substate). A composite state can have at most one shallow history vertex. A transition coming into the shallow history vertex is equivalent to a transition coming into the most recent active substate of a state. At most one transition may originate from the history connector to the default shallow history state. This transition is taken in case the composite state had never been active before. Entry actions of states entered on the path to the state represented by a shallow history are performed.

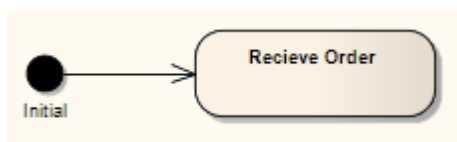
#### 6.4.1.21 Initial



### Description

The Initial element is used by Activity and State Machine diagrams. In Activity diagrams, it defines the start of a flow when an Activity is invoked. With State Machines, the Initial element is a pseudo-state used to denote the default state of a Composite State; there can be one Initial vertex in each Region of the Composite State.

This simple example shows the start of a flow to receive an order.



See *UML Superstructure Specification*, v2.1.1, Figure 12.97, p. 378.

The activity flow is completed by a Final or Flow Final node.

### Notes

- Moving a diagram generally does not affect the location of elements in packages; if you move a diagram out of one package into another, all the elements in the diagram remain in the original package

However, Initial elements are used only within one diagram, have no meaning outside that diagram, and are never re-used in any other diagram; therefore, if you move a diagram containing these elements, they are moved to the new parent package with the diagram

#### Toolbox icon



#### Learn more

- [Activity Diagram](#) <sup>[1199]</sup>
- [State Machine Diagram](#) <sup>[1203]</sup>
- [Activity](#) <sup>[1279]</sup>
- [Pseudo-state](#) <sup>[1208]</sup>
- [Composite State](#) <sup>[1330]</sup>
- [Region](#) <sup>[1328]</sup>
- [Final](#) <sup>[1305]</sup>
- [Flow Final](#) <sup>[1306]</sup>

#### OMG UML Specification:

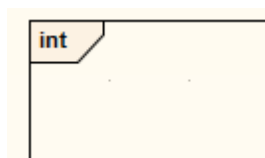
The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 537) states:

*An initial pseudostate represents a default vertex that is the source for a single transition to the default state of a composite state. There can be at most one initial vertex in a region. The outgoing transition from the initial vertex may have a behavior, but not a trigger or guard.*

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 378) also states:

*An initial node is a control node at which flow starts when the activity is invoked.*

### 6.4.1.22 Interaction

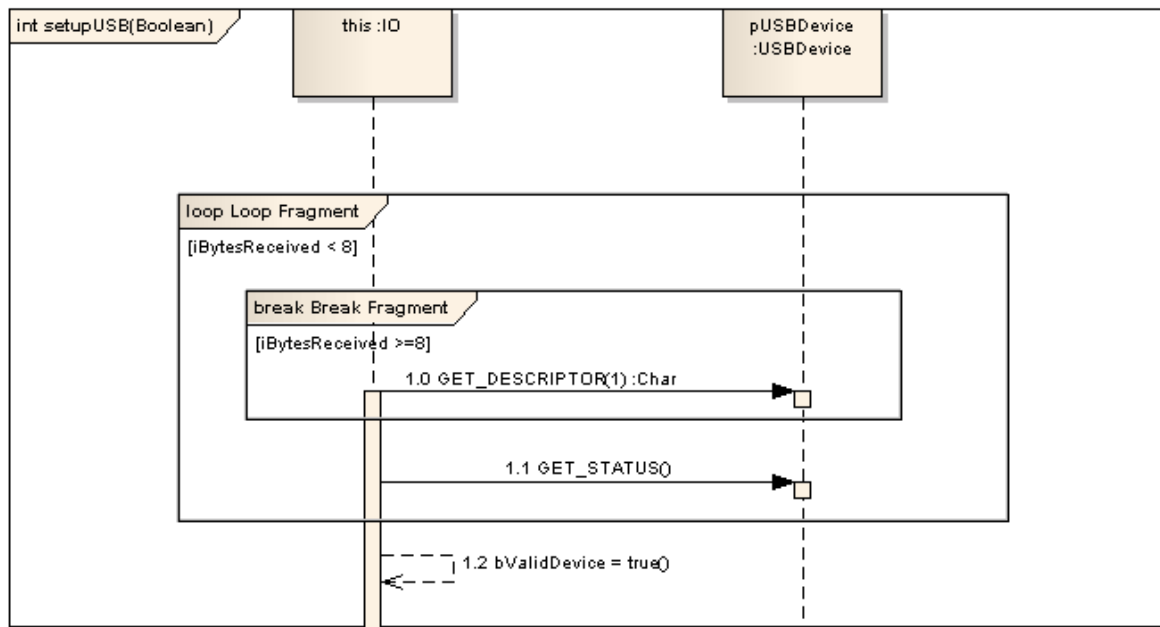


#### Description

You can use an **Interaction** element on a Sequence diagram to describe a system, representing its interactions at varying levels of detail, for review both by design professionals and by end users and stakeholders. An Interaction element can contain these types of diagram:

- Sequence

- Interaction Overview
- Communication
- Timing

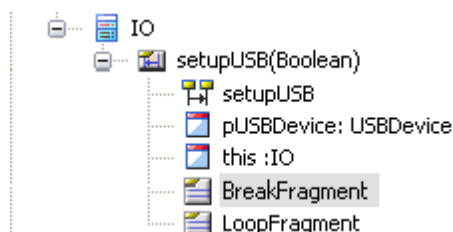


An Interaction element in Enterprise Architect is treated as a behavior of the classifier it is encapsulated within. It can have parameters and return types, which are modeled using the Behavior tab of the Interaction element's Properties dialog. The element is interpreted as a method of the containing Class in the generated code (see the *Generate Code From Behavioral Model* topic).

An Interaction element can also be set as the classifier for an **Interaction Occurrence** in a Sequence diagram, or for a **Call Behavior Action** in an Activity diagram. Establishing such an association (between a behavior and a behavior call) facilitates adding arguments that can be individually mapped to the associated behavior's parameters.

### Notes

- The behavioral code generation engine expects the Sequence diagram and all its associated Messages and Interaction Fragments to be encapsulated within an Interaction element (such as `setupUSB` in this example)



(The `IO` Class is available in the *EAExample* model, under **Systems Engineering Model | Implementation Model | Software**)

- The Interaction icon is available from the **Additional** page of the Interaction Toolbox

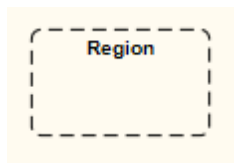
### Toolbox icon



### Learn more

- [Sequence Diagram](#) <sup>[1249]</sup>
- [Interaction Toolbox](#) <sup>[806]</sup>
- [Interaction Overview Diagram](#) <sup>[1262]</sup>
- [Communication Diagram](#) <sup>[1259]</sup>
- [Timing Diagram](#) <sup>[1225]</sup>
- [Generate Code From Behavioral Model](#) <sup>[2121]</sup>
- [Interaction Occurrence](#) <sup>[1317]</sup>
- [Call Behavior Action](#) <sup>[1268]</sup>
- [Associate with Different Behavior](#) <sup>[1027]</sup>
- [Arguments](#) <sup>[1027]</sup>

#### **6.4.1.23 Interruptible Activity Region**

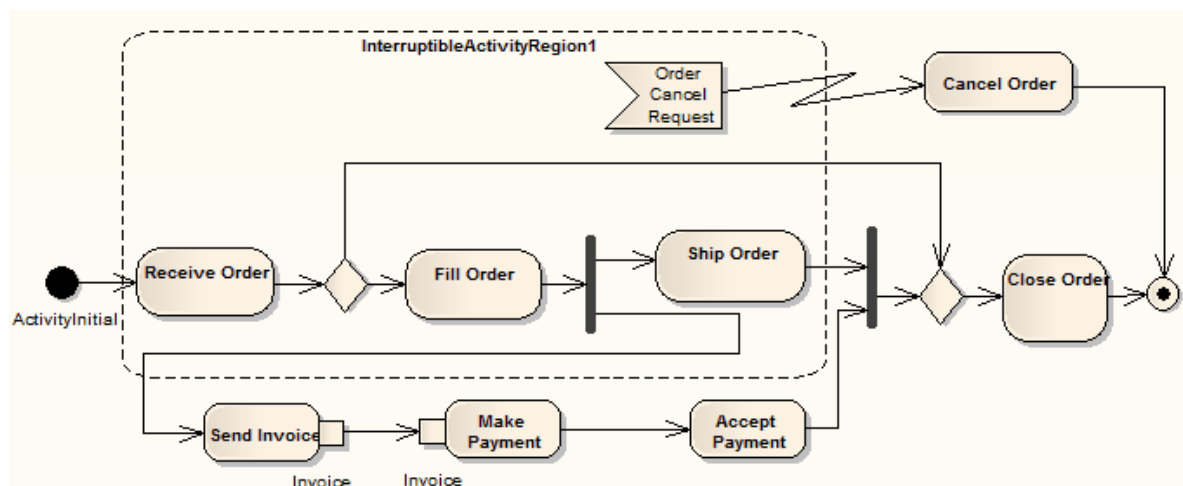


### Description

You create an Interruptible Activity Region as one variant of a Region (the other is an Expansion Region).

In an Activity diagram, an Interruptible Activity Region surrounds a group of Activity elements, all affected by certain interrupts in such a way that all tokens passing within the region are terminated should the interruption(s) be raised. Any processing occurring within the bounds of an Interruptible Activity Region is terminated when a flow is instigated across an interrupt flow to an external element.

The example below illustrates that an order cancellation kills any processing of the order at the receipt, filling or shipping stage.



See *UML Superstructure Specification, v2.1.1, figure 12.100, p. 381.*

#### Toolbox icon



#### Learn more

- [Activity diagram](#) <sup>[1199]</sup>
- [Activity](#) <sup>[1279]</sup>
- [Region](#) <sup>[1328]</sup>
- [Expansion Region](#) <sup>[1301]</sup>

#### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 380*) states:

*An interruptible region contains activity nodes. When a token leaves an interruptible region via edges designated by the region as interrupting edges, all tokens and behaviors in the region are terminated.*

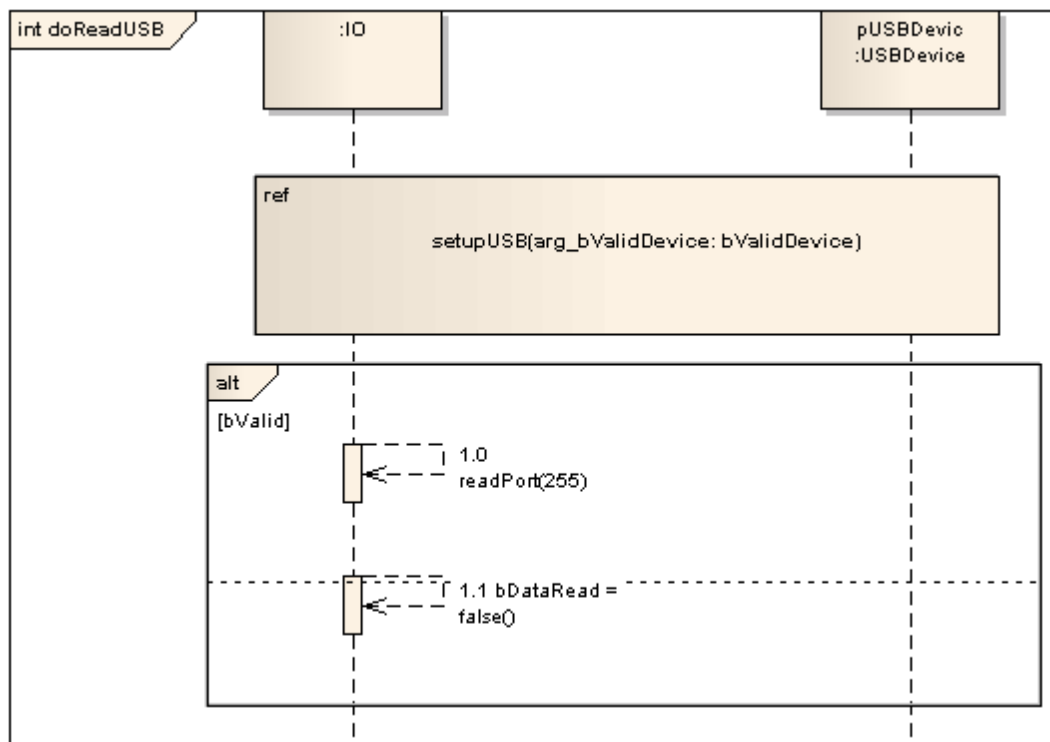
#### 6.4.1.23.1 Interaction Occurrence



An **Interaction Occurrence** (or **InteractionUse**) is a reference to an existing **Interaction** (Sequence) diagram. Interaction Occurrences are visually represented by a frame, with **ref** in the frame's title space. The diagram name is indicated in the frame contents. To create an Interaction Occurrence, simply open a Sequence diagram (preferably contained within an Interaction element) and drag another Sequence

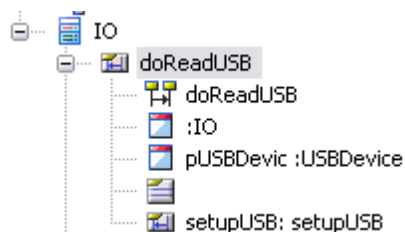
diagram (also preferably contained within an Interaction element) into its workspace. A dialog displays, providing configuration options. The resulting Interaction Occurrence acts as an invocation of the original Interaction. You use the Call tab of the element Properties dialog to set up the actual arguments of the Interaction and also to change to a different associated Interaction element.

The following figure illustrates the use of an Interaction Occurrence in another Interaction (Sequence) diagram. You can display the sequence represented by the Interaction Occurrence by double-clicking on the element.



### Notes

- The behavioral code generation engine expects the Sequence diagram and all its associated messages and interaction fragments to be encapsulated within an Interaction element (such as `doReadUSB` in the example below)



### Learn more

- [Sequence Diagram](#) <sup>[1249]</sup>
- [Interaction element](#) <sup>[1314]</sup>
- [Behavior Calls Invocations](#) <sup>[1026]</sup>
- [Arguments](#) <sup>[1027]</sup>

**OMG UML Specification:**

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 423) refers to an Interaction Occurrence as an **InteractionUse**, and states:

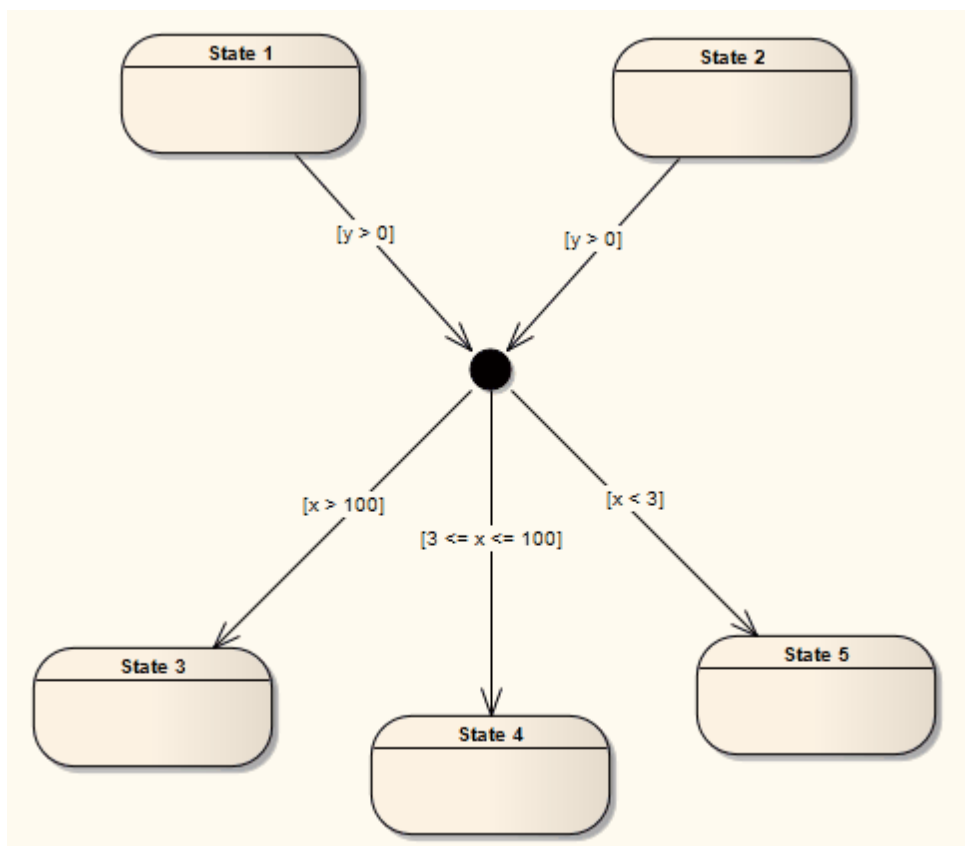
*An InteractionUse refers to an Interaction. The InteractionUse is a shorthand for copying the contents of the referred Interaction where the InteractionUse is. To be accurate the copying must take into account substituting parameters with arguments and connect the formal gates with the actual ones.*

*It is common to want to share portions of an interaction between several other interactions. An InteractionUse allows multiple interactions to reference an interaction that represents a common portion of their specification.*

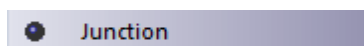
**6.4.1.24 Junction****Description**

Junction pseudo-states are used to design complex transitional paths in State Machine diagrams. A Junction can be used to combine or merge multiple paths into a shared transition path. Alternatively, a Junction can split an incoming path into multiple paths, similar to a Fork pseudostate. Unlike Forks or Joins, Junctions can apply guards to each incoming or outgoing transition, such that if the guard expression is false, the transition is disabled.

The following example illustrates how guards can be applied to transitions coming into or out of a Junction pseudo-state.



#### Toolbox icon



#### Learn more

- [State Machine Diagram](#) <sup>[1203]</sup>
- [Pseudo-states](#) <sup>[1208]</sup>
- [Fork](#) <sup>[1309]</sup>
- [Join](#) <sup>[1310]</sup>

#### OMG UML Specification:

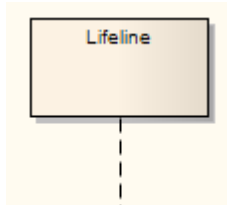
The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 538*) states:

*... junction vertices are semantic-free vertices that are used to chain together multiple transitions. They are used to construct compound transition paths between states. For example, a junction can be used to converge multiple incoming transitions into a single outgoing transition representing a shared transition path (this is known as a merge). Conversely, they can be used to split an incoming transition into multiple outgoing transition segments with different guard conditions. This realizes a static conditional branch. (In the latter case, outgoing transitions whose guard conditions evaluate to false are disabled. A predefined guard denoted "else" may be defined for at most one outgoing transition. This transition is enabled if all the guards*



labeling the other transitions are false.) Static conditional branches are distinct from dynamic conditional branches that are realized by choice vertices.

#### 6.4.1.25 Lifeline



##### Description

A Lifeline is an individual participant in an interaction (that is, Lifelines cannot have multiplicity). A Lifeline represents a distinct connectable element. To specify that representation within Enterprise Architect, right-click on the Lifeline and select the **Advanced | Instance Classifier** context menu option. The Select <Item> dialog displays which you use to locate the required project classifiers.

Lifelines are available in Sequence diagrams. There are different Lifeline elements for Timing diagrams (State Lifeline and Value Lifeline); however, although the representation differs between the two diagram types, the meaning of the Lifeline is the same.

##### Toolbox icon



##### Learn more

- [Instance Classifier](#) <sup>[994]</sup>
- [Sequence Diagram](#) <sup>[1249]</sup>
- [Timing diagram](#) <sup>[1225]</sup>
- [State Lifeline](#) <sup>[1335]</sup>
- [Value Lifeline](#) <sup>[1355]</sup>

##### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p.489) states:

*A lifeline represents an individual participant in the Interaction. While Parts and StructuralFeatures may have multiplicity greater than 1, Lifelines represent only one interacting entity.*

*Lifeline is a specialization of NamedElement.*

*If the referenced ConnectableElement is multivalued (i.e. has a multiplicity > 1), then the Lifeline may have an expression (the 'selector') that specifies which particular part is represented by this Lifeline. If the selector is omitted this means that an arbitrary representative of the multivalued ConnectableElement is chosen.*

### 6.4.1.26 Merge



#### Description

A Merge Node brings together a number of alternative flow paths in Activity, Analysis and Interaction Overview diagrams. For example, if a Decision is used after a Fork, the two flows coming out of the Decision must be merged into one before going to a Join; otherwise, the Join waits for both flows, only one of which arrives.

A Merge Node has multiple incoming edges and a single outgoing edge. The edges coming into and out of a Merge Node must be either all object flows or all control flows.

#### Toolbox icon



#### Learn more

- [Activity Diagram](#) <sup>[1199]</sup>
- [Analysis Diagram](#) <sup>[1801]</sup>
- [Interaction Overview Diagram](#) <sup>[1262]</sup>
- [Decision](#) <sup>[1294]</sup>
- [Fork](#) <sup>[1309]</sup>
- [Join](#) <sup>[1310]</sup>
- [Object flow](#) <sup>[1435]</sup>
- [Control flow](#) <sup>[1403]</sup>

#### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 387) states:

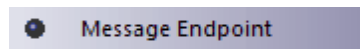
*A merge node is a control node that brings together multiple alternate flows. It is not used to synchronize concurrent flows but to accept one among several alternate flows.*

### 6.4.1.27 Message Endpoint



#### Description

A Message Endpoint element defines the termination of a State or Value Lifeline in a Timing diagram.

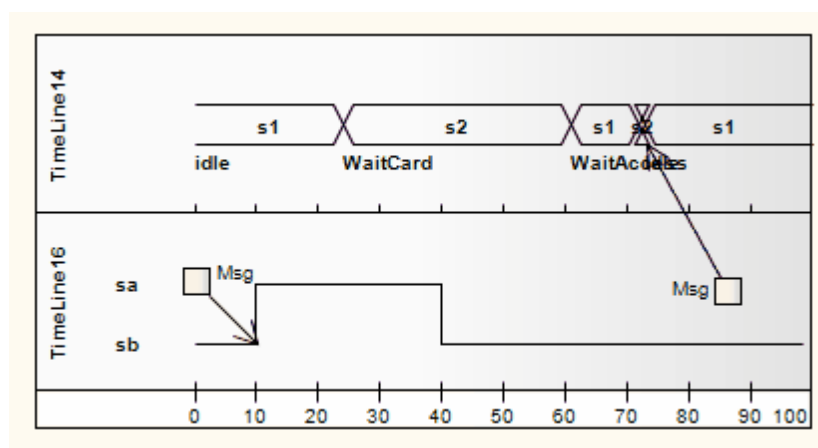
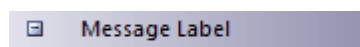
Toolbox iconLearn more

- [Timing Diagram](#)<sup>[1225]</sup>
- [State Lifeline](#)<sup>[1335]</sup>
- [Value Lifeline](#)<sup>[1355]</sup>

**6.4.1.28 Message Label**Description

A Message Label is an alternative way of denoting Messages between Lifelines, which is useful for 'uncluttering' Timing diagrams strewn with messages. To indicate a Message between Lifelines, draw a connector from the source Lifeline into a Message Label. Next, draw a connector from another Message Label to the target Lifeline. Note that the label names must match to reflect that the message occurs between the two Message Labels.

The following diagram illustrates how Message Labels are used to construct a message between Lifelines.

Toolbox iconLearn more

- [Timing diagram](#) <sup>[1225]</sup>
- [Messages](#) <sup>[1437]</sup>

#### **OMG UML Specification:**

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 518) states:

*Labels are only notational shorthands used to prevent cluttering of the diagrams with a number of messages crisscrossing the diagram between Lifelines that are far apart. The labels denote that a Message may be disrupted by introducing labels with the same name.*

### **6.4.1.29 Note**



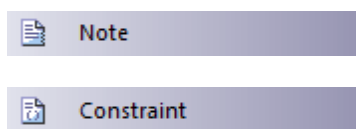
#### **Description**

A Note element is a textual annotation that can be attached to a set of elements of any other type. The attachment is created separately, using a Notelink connector. Both Note and Notelink are available in any Enterprise Architect diagram, through the Common pages of the Toolbox.

A Note is also called a Comment.

A Constraint is a form of Note, identifying a constraint on other elements. As for a Note, you can connect the Constraint element to other elements using a Notelink connector. This element is just a means of documenting the fact that there are constraints; it has no impact on the other elements. You define the types of constraint in the project reference data, apply them to the element in the element Properties dialog, and manage them through the Scenarios & Requirements window.

#### **Toolbox icon**



#### **Learn more**

- [Notelink Connector](#) <sup>[1435]</sup>
- [Constraint Types](#) <sup>[1160]</sup>
- [Scenarios & Requirements Window](#) <sup>[992]</sup>

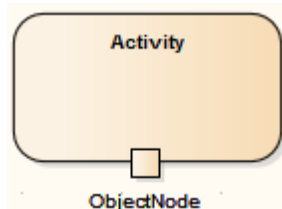
#### **OMG UML Specification:**

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 59) states:

*A comment gives the ability to attach various remarks to elements. A comment carries no semantic force, but may contain information that is useful to a modeler.*

*A comment can be owned by any element.*

#### 6.4.1.30 Object Node



##### Description

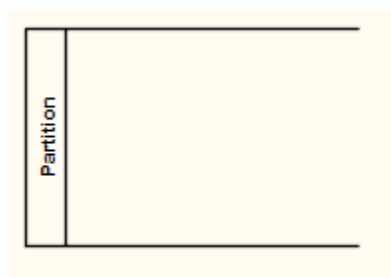
An **Object Node** holds data that is input to or output from an **Activity element**. To set the type of an Object Node, click on it and press **Ctrl+L** (to select an **Instance Classifier**). Object Nodes can be connected by **Object Flow connectors**; if the Object Nodes on each end of an Object Flow are typed, their types should be compatible.

An Action Pin is a specialized form of Object Node.

##### Learn more

- [Activity](#) <sup>[1279]</sup>
- [Object Flow](#) <sup>[1435]</sup>
- [Action Pin](#) <sup>[1277]</sup>
- [Select <item> Dialog](#) <sup>[994]</sup>

#### 6.4.1.31 Partition



##### Description

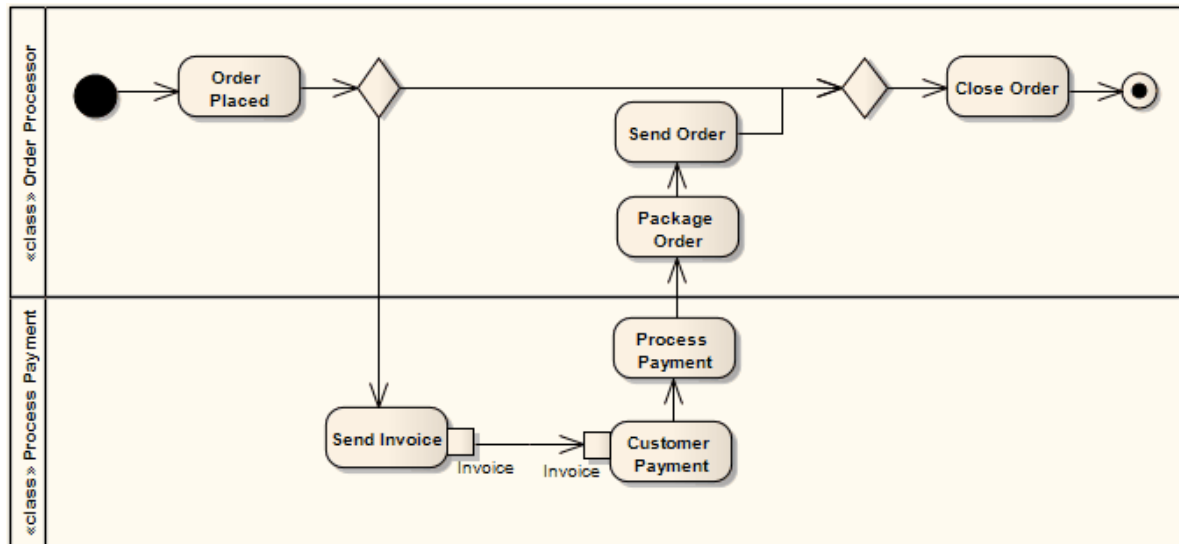
Enterprise Architect supports two types of Activity Partition:

- The Activity Partition feature, which is used to logically organize an Activity element
- The Activity Partition element, described in this topic, which is used to logically organize an Activity diagram

In effect, these are the same. They partition the Actions of the Activity without affecting the token flow, helping

to structure the view or parts of the Activity.

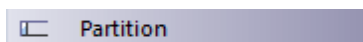
The following example depicts the partitioning between the Classes *Process Payment* and *Order Processor*.



The Partition orientation defaults to horizontal. To turn it into a vertical Partition, right-click on it and select the **Advanced | Vertical Partition** context menu option.

You can neatly align and join the Activity Partitions on a diagram using the element context menu **Dockable** option. For Partitions, the option defaults to selected.

#### Toolbox icon



#### Learn more

- [Activity Diagram](#)<sup>[1199]</sup>
- [Activity Element](#)<sup>[1279]</sup>
- [Activity Partition Feature](#)<sup>[1283]</sup>

#### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 341*) states:

*Partitions divide the nodes and edges to constrain and show a view of the contained nodes. Partitions can share contents. They often correspond to organizational units in a business model. They may be used to allocate characteristics or resources among the nodes of an activity.*

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 341*) also states:

An activity partition is a kind of activity group for identifying actions that have some characteristic in common.

#### 6.4.1.32 Receive

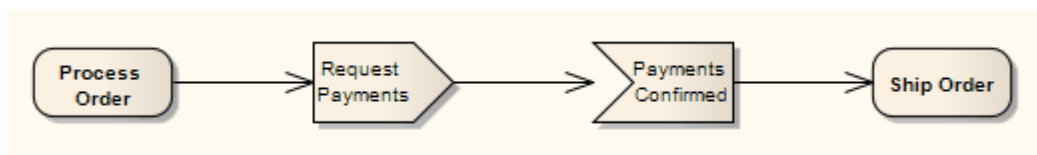


##### Description

A Receive element is used to define the acceptance or receipt of a request, in an Activity diagram. Movement from a Receive element occurs only once receipt is fulfilled according to its specification. The Receive element comes in two forms:

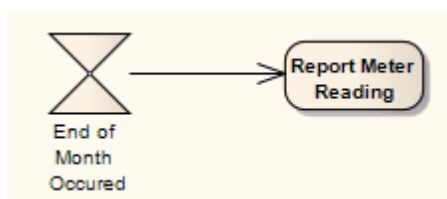
- *Accept Event Action* element (pennant shape)
- *Accept Time Event Action* element (hourglass shape)

The following example reflects a payment process on an order. Upon receiving the payment (from *Request Payments*, a Send element), the payment is confirmed and the flow continues to ship the order.



See *UML Superstructure Specification, v2.1.1, figure 12.26, p. 312.*

To depict an Accept Time Event, use the standard Receive element from the Toolbox. Right-click on this element, and select the **Advanced | Accept Time Event** context menu option. The following example shows the hourglass-shaped *Accept Time Event Action*:



See *UML Superstructure Specification, v2.1.1, figure 12.27, p. 312.*

##### Toolbox icon



##### Learn more

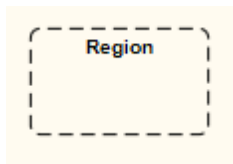
- [Activity Diagram](#) <sup>[1199]</sup>
- [Send Element](#) <sup>[1328]</sup>

### **OMG UML Specification**

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 239) states:

*AcceptEventAction* is an action that waits for the occurrence of an event meeting specified conditions.

#### **6.4.1.33 Region**



### **Description**

Enterprise Architect supports two types of *Region* element:

- Expansion Region
- Interruptible Activity Region

When you add a Region element to an Activity diagram, a prompt displays to enable you to specify the Region type.

### **Toolbox icon**



### **Learn more**

- [Activity Diagram](#) <sup>[1199]</sup>
- [Expansion Region](#) <sup>[1301]</sup>
- [Interruptible Activity Region](#) <sup>[1316]</sup>

#### **6.4.1.34 Send**



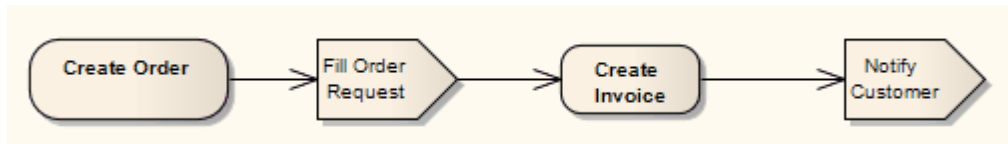
### **Description**

The Send element is used to depict the action of sending a signal, in an Activity diagram. It is the opposite of



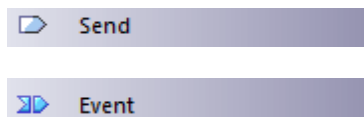
a Receive element. You can also create Send events using the Event icon on the State page of the Diagram Toolbox.

The following example shows an order being processed, where a signal is sent to fill the processed order and, upon creation of the resulting invoice, a notification is sent to the customer.



See *UML Superstructure Specification, v2.1.1, figure 12.132, p. 408*.

#### Toolbox icon



#### Learn more

- [Activity Diagram](#)<sup>[1199]</sup>
- [Receive](#)<sup>[1327]</sup>
- [State Toolbox](#)<sup>[807]</sup>

#### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 285*) states:

*SendObjectAction is an action that transmits an object to the target object, where it may invoke behavior such as the firing of state machine transitions or the execution of an activity. The value of the object is available to the execution of invoked behaviors. The requestor continues execution immediately. Any reply message is ignored and is not transmitted to the requestor.*

### 6.4.1.35 State



#### Description

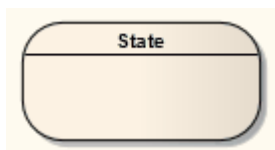
A *State* represents a situation where some invariant condition holds; this condition can be static (waiting for an event) or dynamic (performing a set of activities). State modeling is usually related to Classes, and describes the enableable states a Class or element can be in and the transitions that enable the element to move there. There are two types of State: *Simple States* and *Composite States*, both created from the State

element from the Toolbox.

Furthermore, there are pseudo-states, resembling some aspect of a State but with a pre-defined implication. Pseudo-states model complex transitional paths, and classify common State Machine behavior.

You can define entry, internal and exit actions for a State using operations.

If a State element has features such as attributes or operations, the depiction of the element in a diagram has a line under the element name. This line persists if the features are hidden. The line also displays if the **Show State Compartment** checkbox is selected on the Objects page of the Options dialog (**Tools | Options | Objects**).



#### Toolbox icon



#### Learn more

- [State Machine Diagram](#) <sup>[1203]</sup>
- [Class](#) <sup>[1363]</sup>
- [Pseudo-states](#) <sup>[1208]</sup>
- [General Properties of Operations](#) <sup>[1015]</sup>
- [Composite State](#) <sup>[1330]</sup>
- [Transition](#) <sup>[1446]</sup>

#### OMG UML Specification

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 546) states:

*A state models a situation during which some (usually implicit) invariant condition holds. The invariant may represent a static situation such as an object waiting for some external event to occur. However, it can also model dynamic conditions such as the process of performing some activity (i.e., the model element under consideration enters the state when the activity commences and leaves it as soon as the activity is completed).*

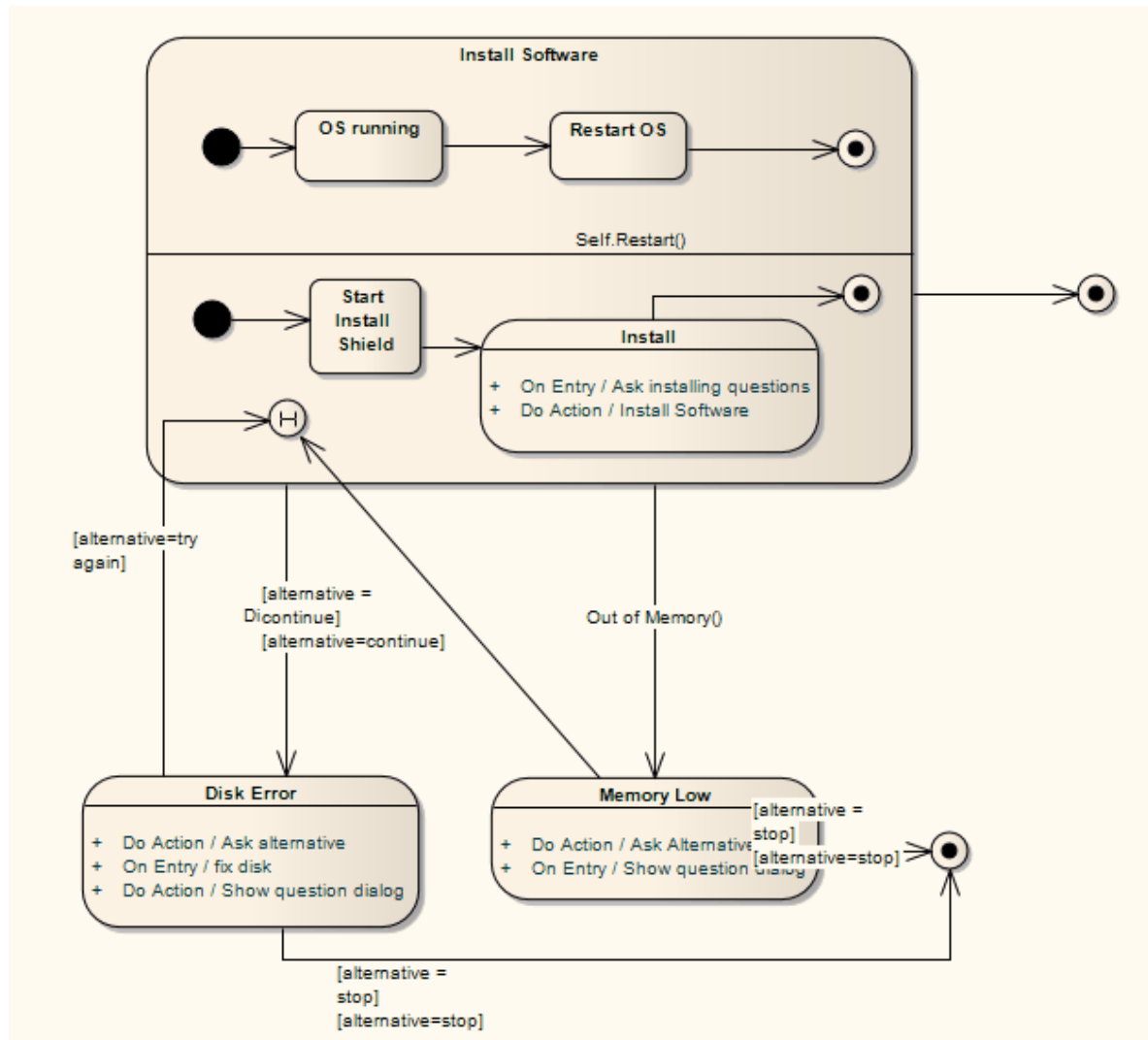
### **6.4.1.35.1 Composite State**

#### Description

Composite States are composed within the State Machine diagram by expanding a State element, adding Regions if applicable, and dragging further State elements, related elements and connectors within its boundaries. The internal State elements are then referred to as Sub-states.

(You can also define a State element, as with many other types of element, as a composite element; this then has a hyperlink to a child diagram that can be another State Machine diagram or other type of diagram elsewhere in the model.)

Composite States can be orthogonal, if Regions are created. If a Composite State is orthogonal, its entry denotes that a single Sub-state is concurrently active in each Region. The hierarchical nesting of Composite States, coupled with Region use, generates a situation of multiple States concurrently active; this situation is referred to as the active State configuration.



#### Learn more

- [State Machine Diagram](#)<sup>[1203]</sup>
- [State Element](#)<sup>[1329]</sup>
- [Regions](#)<sup>[1209]</sup>
- [Composite Element](#)<sup>[936]</sup>

#### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 478) states:

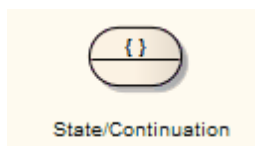
*A composite state either contains one region or is decomposed into two or more orthogonal regions. Each region has a set of mutually exclusive disjoint sub vertices and a set of transitions. A given state may only be decomposed in one of these two ways.*

*Any state enclosed within a region of a composite state is called a sub state of that composite state. It is called a direct sub state when it is not contained by any other state; otherwise it is referred to as an indirect sub state.*

*Each region of a composite state may have an initial pseudostate and a final state. A transition to the enclosing state represents a transition to the initial pseudostate in each region. A newly-created object takes its topmost default transitions, originating from the topmost initial pseudostates of each region.*

*A transition to a final state represents the completion of activity in the enclosing region. Completion of activity in all orthogonal regions represents completion of activity by the enclosing state and triggers a completion event on the enclosing state. Completion of the topmost regions of an object corresponds to its termination.*

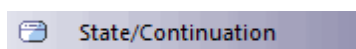
### 6.4.1.36 State/Continuation



#### Description

The State/Continuation element serves two different purposes for Interaction (Sequence) diagrams, as State Invariants and Continuations. The system prompts you to identify the purpose when you create the element.

#### Toolbox icon



#### Learn more

- [Sequence Diagram](#)<sup>[1249]</sup>
- [Continuation](#)<sup>[1332]</sup>
- [State Invariant](#)<sup>[1334]</sup>

#### 6.4.1.36.1 Continuation

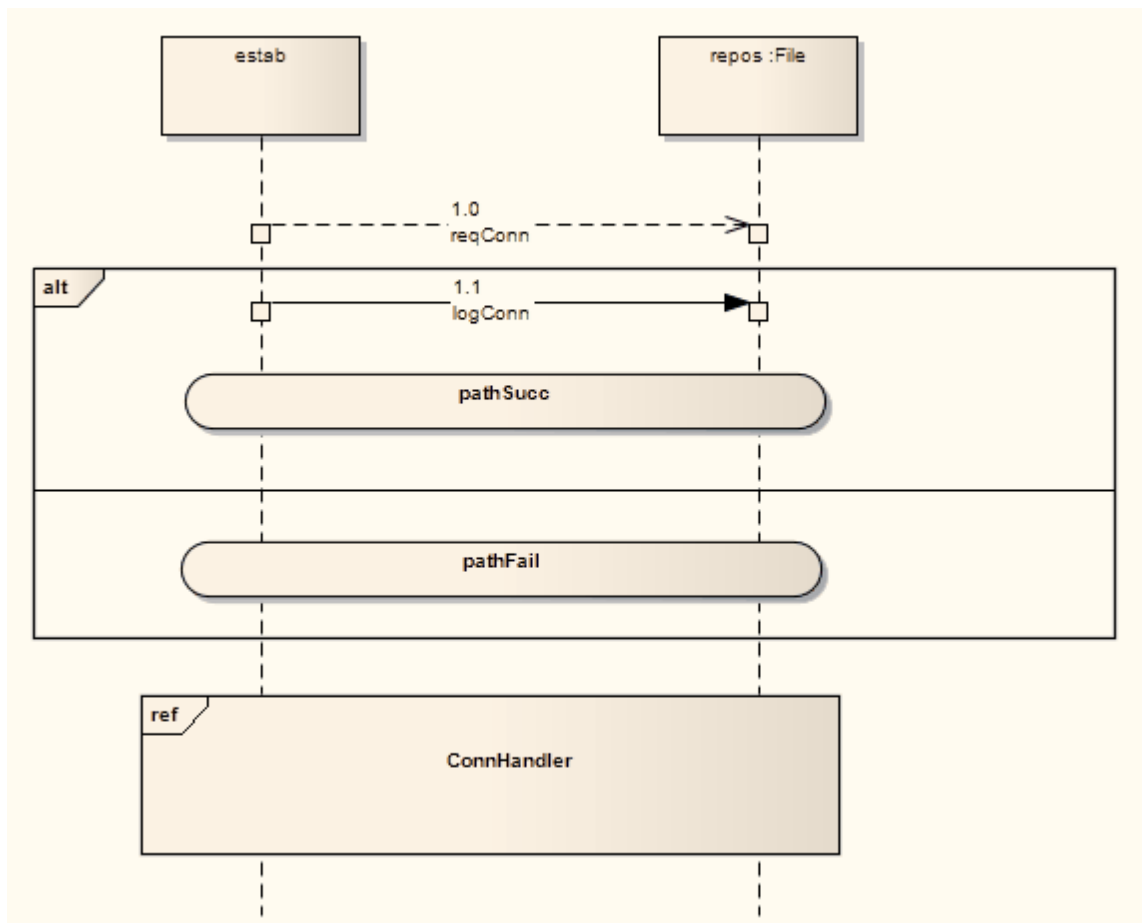
#### Description

A Continuation is used in *seq* and *alt* Combined Fragments, to indicate the branches of continuation an operand follows. To indicate a continuation, end an operand with a Continuation, and indicate the continuation branch with a matching Continuation (same name) preceding the Interaction Fragment.

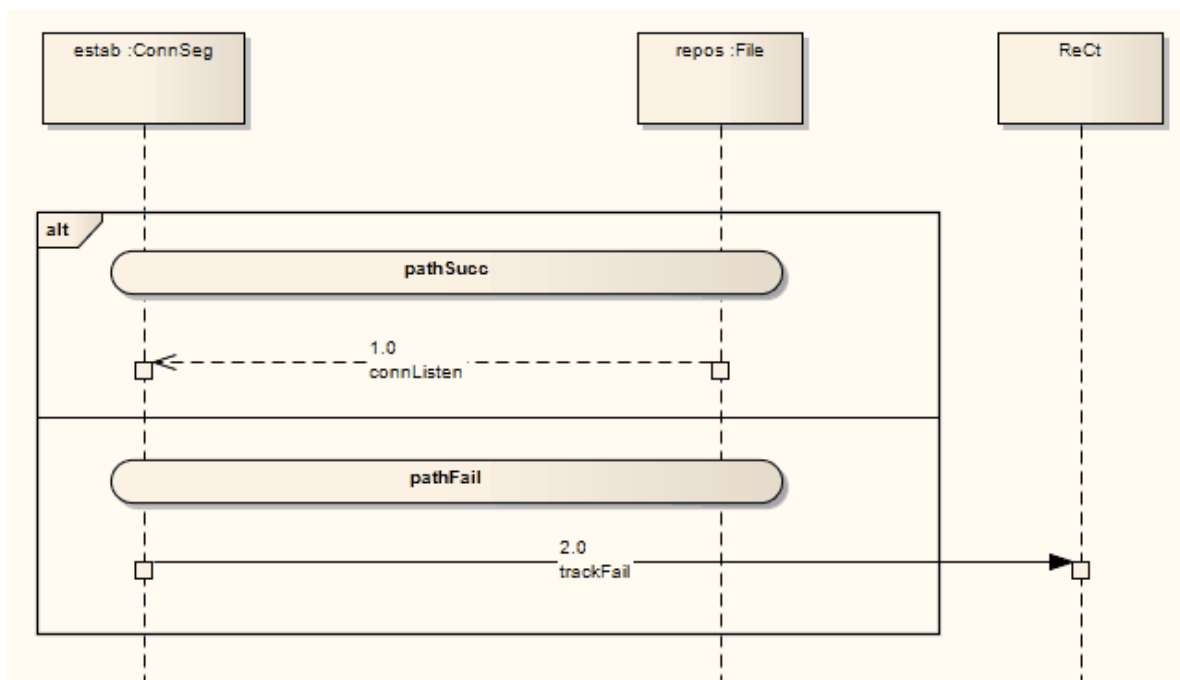
You create a Continuation by dragging the State/Continuation element onto the diagram from the Interaction Elements page of the Toolbox.

For the following continuation example, an *alt* Combined Fragment has Continuations *pathSucc* and *pathFail*. These Continuations are located within the Interaction Occurrence *ConnHandler*, which has

subsequent events based on the continuation.



The following diagram shows the interaction referenced by the *Interaction Occurrence*.



#### Learn more

- [Combined Fragments](#)<sup>[1287]</sup>
- [State/Continuation Element](#)<sup>[1332]</sup>
- [Interaction Occurrence](#)<sup>[1317]</sup>

#### OMG UML Specification

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 474) states:

*A Continuation is a syntactic way to define continuations of different branches of an Alternative CombinedFragment. Continuation is intuitively similar to labels representing intermediate points in a flow of control.*

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 474) also states:

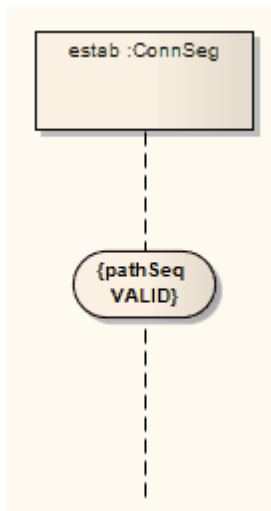
*Continuations have semantics only in connection with Alternative CombinedFragments and (weak) sequencing.*

*If an InteractionOperand of an Alternative CombinedFragment ends in a Continuation with name (say) X, only InteractionFragments starting with the Continuation X (or no continuation at all) can be appended.*

#### 6.4.1.36.2 State Invariant

A **State Invariant** is a condition applied to a Lifeline, which must be fulfilled for the Lifeline to exist. You create a State Invariant by dragging the State/Continuation element onto the diagram from the Interaction Elements page of the Toolbox.

The following diagram illustrates a State Invariant.



When a State Invariant is moved near to a Lifeline, it snaps to the center. If the sequence object is dragged left or right, the State Invariant moves with it.

#### Learn more

- [Lifeline](#) <sup>[1321]</sup>
- [State/Continuation](#) <sup>[1332]</sup>

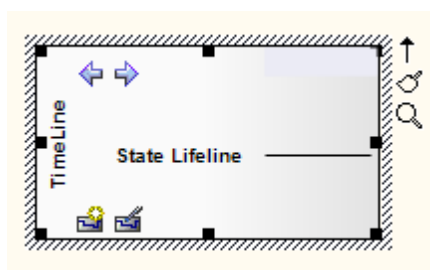
#### OMG UML Specification

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 502) states:

*A StateInvariant is a runtime constraint on the participants of the interaction. It may be used to specify a variety of different kinds of constraints, such as values of attributes or variables, internal or external states, and so on.*

*A StateInvariant is an InteractionFragment and it is placed on a Lifeline.*

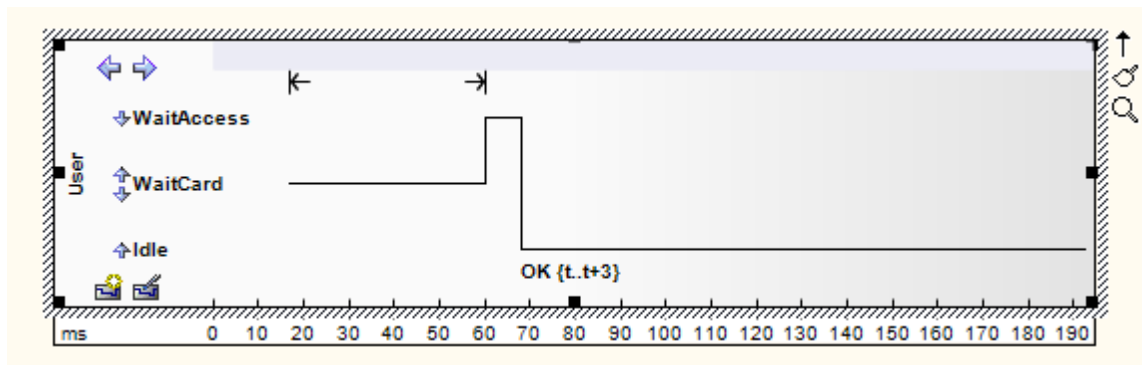
### 6.4.1.37 State Lifeline



A **Lifeline** is the path an object takes across a measure of time, as indicated by the x-axis. There are two sorts: **State Lifelines** (defined here) and **Value Lifelines**, both used in Timing diagrams.

A State Lifeline follows discrete transitions between states, which are defined along the y-axis of the timeline. Any transition has optional attributes of timing constraints, duration constraints and observations.

An example of a State Lifeline is shown below:



See *UML Superstructure Specification, v2.1.1, figure 14.29, p. 519*.

A State Lifeline consists of a set of transition points. Each transition point can be defined with the following properties:

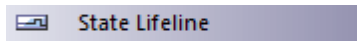
Property	Description
<b>At time</b>	Specifies the starting time for a change of state.
<b>Transition to</b>	Indicates the state to which the lifeline changes.
<b>Event</b>	Describes the occurring event.
<b>Timing constraints</b>	Refers to the time taken for a state to change within a lifeline, or the time taken to transmit a message (e.g. $t..t+3$ ).
<b>Timing observations</b>	Provides information on the time of a state change or sent message.
<b>Duration constraints</b>	Pertains to a lifeline's period at a particular state. The constraint could be instigated by a change of state within a lifeline, or that lifeline's receipt of a message.
<b>Duration observations</b>	Indicates the interval of a lifeline at a particular state, begun from a change in state or message receipt.

In the example diagram above, the **OK** transition point has these properties:



Property	Value
At Time	18 ms
Transition to	Idle
Event	OK
Timing constraints	t..t+3
Timing observations	—
Duration constraints	—
Duration observations	—

#### Toolbox icon



#### Learn more

- [Timing Diagram](#) 
- [Value Lifelines](#) 

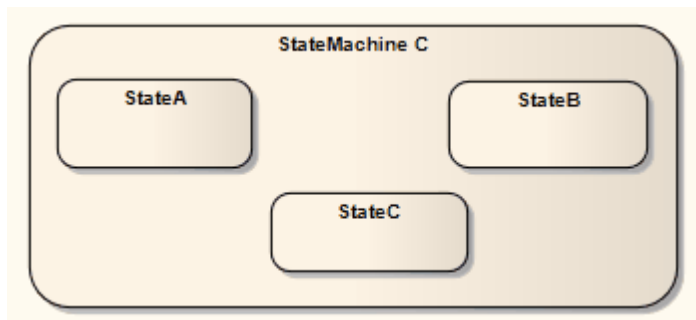
#### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 518) states:

*This is the state of the classifier or attribute, or some testable condition, such as an discrete enumerable value.*

*It is also permissible to let the state-dimension be continuous as well as discrete. This is illustrative for scenarios where certain entities undergo continuous state changes, such as temperature or density.*

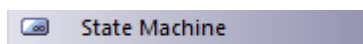
### 6.4.1.38 State Machine



#### Description

A State Machine element is a container for groups of related State elements. You can create sections of a State Machine diagram, showing the organization of the inter-related State elements, and enclose each section in a State Machine element. You can also create Regions on a State Machine element.

#### Toolbox icon



#### Learn more

- [State Machines](#)<sup>[1203]</sup> (diagrams)
- [State](#)<sup>[1329]</sup>
- [Regions](#)<sup>[1209]</sup>

### 6.4.1.39 Structured Activity

#### Description

*Structured Activity* elements are used in Activity diagrams. A Structured Activity is an activity node that can have subordinate nodes as an independent *Activity Group*. You can set an option to ensure that no other Activities or their side effects interfere with this Activity's processing (the **Must Isolate** checkbox in the Structured Activity element Properties dialog).

Enterprise Architect provides a number of forms of Structured Activity, both basic and specialized, for creating composite Activity elements quickly and simply.

**Access** [Diagram](#) | [Diagram Toolbox](#) | [More tools...](#) | [UML](#) | [Activity](#)

Topic	Description	See Also
<b>Create Structured Activities</b>	When you drag a Structured Activity icon from the Toolbox onto a diagram, a short menu displays from which you select one of the following options:	<a href="#">Loop Node</a> <sup>[1341]</sup> <a href="#">Conditional Node</a> <sup>[1345]</sup>

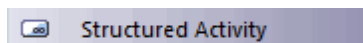
Topic	Description	See Also
	<ul style="list-style-type: none"> <li>• <b>Loop Node</b></li> <li>• <b>Conditional Node</b></li> <li>• <b>Other</b></li> </ul> <p>The first two options specifically create a Loop Node or Conditional Node.</p> <p>The <b>Other</b> option displays the New Structured Activity dialog, on which you can select to create one of the five following types of Structured Activity element.</p>	
<b>Simple Composite Activity</b>	Provides a Composite Activity element with a child Activity diagram.	<a href="#">Activity</a> <sup>[1275]</sup> <a href="#">Composite Elements</a> <sup>[936]</sup>
<b>Loop Node</b>	Represents a sequence of Actions and Activities that are to be repeated within the object.	<a href="#">Loop Node</a> <sup>[1341]</sup>
<b>Conditional Node</b>	Represents an arrangement of Actions and Activities where choice determines which Activities are performed.	<a href="#">Conditional Node</a> <sup>[1345]</sup>
<b>Structured Activity Node</b>	Represents an ordered arrangement of executable Activity nodes (Actions, Decisions, Merges and so on) that can include branched and nested nodes; this is the base element from which the other types of Structured Activity are derived.	<a href="#">Structured Node</a> <sup>[1340]</sup>
<b>Sequential Node</b>	Represents a sequential arrangement of executable Activity nodes.	<a href="#">Sequential Node</a> <sup>[1341]</sup>
<b>Nodes as Composite Elements</b>	<p>Loop Node and Conditional Node elements are composite elements, but the child element structures are created on the <i>same</i> diagram as the node element itself (in the same way as for a Composite State). Therefore:</p> <ul style="list-style-type: none"> <li>• The composite element symbol is not set</li> <li>• You cannot display a separate child diagram</li> <li>• The <b>Show Composite Diagram</b> facility is not available for these nodes</li> </ul> <p>You could set these nodes as 'normal' composite elements and create and open a child diagram, but you should not use Loop or Conditional Nodes in this way.</p> <p>The <b>Show Composite Diagram</b> facility is also not available on the Structured Activity Node, as this is the base element for the Loop and Conditional Nodes. You can, however:</p> <ul style="list-style-type: none"> <li>• Use the Sequential and Structured Activity nodes as</li> </ul>	<a href="#">Composite State</a> <sup>[1330]</sup>

Topic	Description	See Also
	composite elements, and <ul style="list-style-type: none"> <li>Display a child diagram structure on its parent Sequential node</li> </ul>	

### Notes

- To protect the processing of a Loop or Conditional Node Structured Activity from interference from other Activities or their side effects, open the element's Properties dialog and select the **Must Isolate** checkbox on the Loop or Condition page

### Toolbox icon



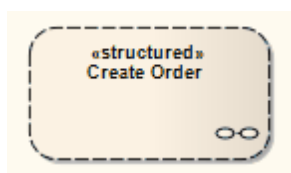
### Learn more

- [Activity Diagram](#) <sup>[1199]</sup>

#### 6.4.1.39.1 Structured Node

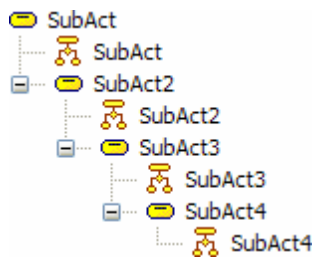
### Description

On a diagram, *Structured Activity Nodes* have broken borders and composite diagram icons, as shown below:



To display the Activity diagram represented by a Structured Activity Node element, double-click on the element.

Structured Activity Node elements can point to child diagrams that themselves contain or consist of Structured Activity elements; that is, the Structured Activity elements are nested, as shown in the section of Project Browser below.

**OMG UML Specification:**

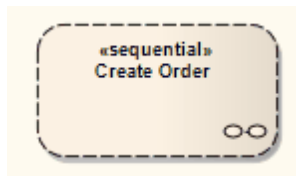
The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 409) states:

*A structured activity node is an executable activity node that may have an expansion into subordinate nodes as an ActivityGroup. The subordinate nodes must belong to only one structured activity node, although they may be nested.*

*A structured activity node represents a structured portion of the activity that is not shared with any other structured node, except for nesting.*

**6.4.1.39.2 Sequential Node****Description**

On a diagram, *Sequential Activity Nodes* have broken borders and composite diagram icons, as shown below:



To display the Activity diagram represented by a Sequential Activity Node element, double-click on the element.

**OMG UML Specification:**

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 408) states:

*A sequence node is a structured activity node that executes its actions in order.*

**6.4.1.39.3 Loop Node**

A *Loop Structured Activity Node* is used for defining a loop, and is commonly associated with 'While', 'Repeat' or 'For' loop statements.


Each Loop Node has three partitions:

- **Setup** commonly initiates variables to be used in the loop's exit-condition; it is executed once on entry to the loop
- **Test** defines the loop exit-condition
- **Body** can contain Actions to be executed repeatedly until the Test produces a false value

The results of the final execution of the Test or Body are available after execution of the Loop is complete.

### Create a Loop Node

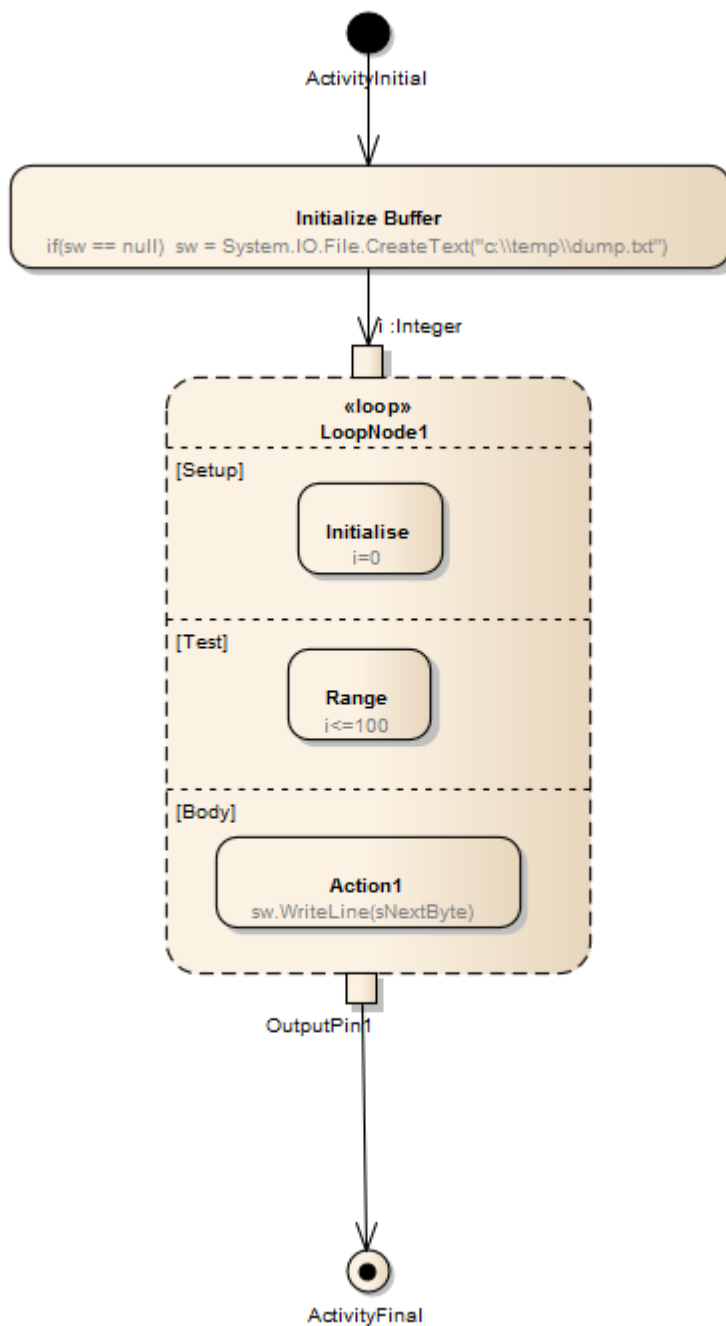
Step	Action	See Also
1	From the Activity page of the Diagram Toolbox, drag a Structured Activity icon onto the Activity diagram.  A short menu displays.	
2	Select the <b>Loop Node</b> option.  The Loop Node displays on the diagram, with the element Properties dialog (if the dialog does not display, double-click on the element).	
3	Complete as many of the common element Properties fields as required.	
4	Select the Loop page and set the following checkboxes as required: <ul style="list-style-type: none"> <li>• <b>Must Isolate</b> - defines concurrency: if selected, no object within the node can be used outside it; the objects are isolated from parallel use</li> <li>• <b>Tested First</b> - defines the loop type; select for a <i>For / While</i> loop, deselect for a <i>Repeat Until</i> loop</li> </ul>	
5	For each of the following fields, click on the ( ... ) or <b>Add</b> button as appropriate, to display the Select Pins dialog and select an Action Pin: <ul style="list-style-type: none"> <li>• <b>Decider</b> (an Output pin within the Test partition, the value of which is examined after execution of the Test to determine whether to execute the loop Body)</li> <li>• <b>Loop Variable Input</b></li> <li>• <b>Loop Variable</b></li> <li>• <b>Body Output</b> and</li> <li>• <b>Result</b></li> </ul> <p>The Select Pins dialog lists only Input pins for the <b>Loop Variable Input</b> field and only Output pins for the other fields.</p> <p>If the required Action Pin does not already exist, you can click on the <b>Add New</b> button on the dialog to automatically create the Input pin or an Output pin for the node.</p>	
6	In the Nodes panel, click on one of the <b>Setup</b> , <b>Test</b> or <b>Body</b> radio buttons to list the Actions and Activities contained in the corresponding partition of the Loop Node.  An element must be completely below the top edge of a partition to be listed for that partition - if it overlaps with the partition above in any way, it is treated as being part of that partition.	
7	Click on the <b>OK</b> button to save the properties of the Loop Node and close the Properties	

Step	Action	See Also
	dialog.	
8	<p>Right-click on the Node in the diagram and select the <b>Structural Elements</b> context menu option.</p> <p>The Structural Elements dialog displays.</p> <p>Select the checkbox against each embedded element and close the dialog.</p> <p>The Action pins should now be visible in the diagram, attached to the Node.</p>	<a href="#">Manage Structural Elements</a> 

A Loop Node is depicted on an Activity diagram as shown below:



You define the Loop nodes by dragging Action elements from the Toolbox page into the Setup, Test and Body partitions. The Body partition can contain several Actions, which can be linked and organized into the required structure. The elements are aligned on the top left of the partition, so that resizing the node maintains the organization of the structure within and between the partitions. If you try to shrink the node below the structure size, the node automatically defaults to the 'best fit' size.



### Notes

- You can check on the exact location of an existing Action Pin by right-clicking on the pin name in the Loop Node Properties dialog and selecting the **Find in Project Browser** context menu option; the location of the Action Pin in the Project Browser is expanded and highlighted

### OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, pp. 384-385*) states:

*A loop node is a structured activity node that represents a loop with setup, test, and body sections.*



*Each section is a well-nested subregion of the activity whose nodes follow any predecessors of the loop and precede any successors of the loop. The test section may precede or follow the body section. The setup section is executed once on entry to the loop, and the test and body sections are executed repeatedly until the test produces a false value. The results of the final execution of the test or body are available after completion of execution of the loop.*

#### 6.4.1.39.4 Conditional Node

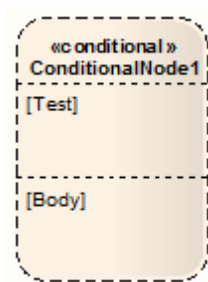
A **Conditional Structured Activity Node** is the modeling equivalent of an **If-Then-Else** programming construct. At its simplest, it consists of a **Clause** containing:

- A **Test** partition that evaluates a condition, and
- A **Body** partition that performs one or more actions if the Test condition is satisfied

You can have more than one Clause, so that if the Test condition is not satisfied its Body is ignored and processing moves to the next Clause and evaluates another Test condition.

Each Clause has a **Decider ActionPin** to hold the result of the Test, and a **Body Output ActionPin** to hold the result of the Body's actions (if executed). The Conditional Node itself has a **result ActionPin** that makes available the overall result of the Node (the output of the first Body to be executed).

A Conditional Node is depicted on an Activity diagram like this:



You define Conditional Nodes by dragging other **Activity diagram** elements from the Toolbox page into the appropriate partition of the element, and linking and organizing the structure as required. The elements are aligned on the top left of the partition, so that resizing the node maintains the organization of the structure within and between the partitions. If you try to shrink the node below the structure size, the node automatically defaults to the 'best fit' size.

When you create a Conditional Node, the element Properties dialog displays. Much of this you can complete as for any other element. However, for the Conditional Node the dialog shows an additional **Condition** page.

On this page, in the **Result** panel, add an Action Pin to hold the result for the node, clicking on the **Add** button to display the Select Pins dialog (a version of the Select <Item> dialog).

A Conditional Node **automatically** contains one **Clause** containing a Test partition and a Body partition, and a **Decider** Pin and **Body Output** Pin. You can add further Clauses as required. For each Clause you add an Action Pin for the **Decider** and for the **Body Output**. Click on the **Save** button to save the Clause definition.

The Select Pin dialog reveals only Output pins as appropriate to the context. If the required Action Pin does not already exist, you can click on the **Add New** button on the dialog to automatically create an Output pin under the appropriate parent node.

For the **Result** and **Body Output** entries, you can check on the exact location of each Action Pin by right-clicking on the entry and selecting the **Find in Project Browser** context menu option.

The Nodes panel, by default, lists the Actions and Activities contained in the Test partition. Click on the **Body**

radio button to list the elements contained in the Body partition. An element must be completely contained in the Body partition to be listed there - if it overlaps with the Test partition in any way, it is treated as being part of the Test partition.

### Add or Remove Clauses

To add another Clause, click on the **Add** button underneath the Clause(s) list. This inserts a new Clause in the list, and identifies which is the preceding (**Predecessor**) Clause and (if appropriate) which is the following (**Successor**) Clause. The remaining fields in the Clause(s) panel are cleared to enable you to add **Decider** and **Body Output** Action Pins. New Test and Body partitions are immediately added to the element on the diagram, and you can populate these partitions with Activity elements, which are then identified in the Nodes panel.

To remove a Clause, highlight it in the list and click on the **Delete** button. This immediately removes the Clause's corresponding partitions from the diagram, along with all their contained Activity elements. Removing a Clause from between two other Clauses adjusts the numerical order; for example, if Clause 2 is removed from between Clause 1 and Clause 3, Clause 3 is renamed as Clause 2, and any further Clauses are also moved up one place.

### OMG UML Specification

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p.355) states:

*A conditional node is a structured activity node that represents an exclusive choice among some number of alternatives.*

*A conditional node consists of one or more clauses. Each clause consists of a test section and a body section. When the conditional node begins execution, the test sections of the clauses are executed. If one or more test sections yield a true value, one of the corresponding body sections will be executed. If more than one test section yields a true value, only one body section will be executed. The choice is nondeterministic unless the test sequence of clauses is specified. If no test section yields a true value, then no body section is executed; this may be a semantic error if output values are expected from the conditional node.*

## 6.4.1.40 Synch



### Description

A Synch state is useful for indicating that concurrent paths of a State Machine are synchronized. After bringing the paths to a synch state, the emerging transition indicates unison.

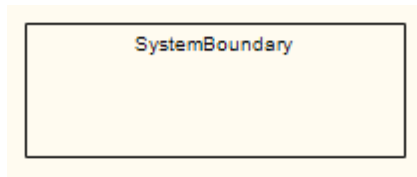
### Toolbox icon



### Learn more

- [State Machine Diagram](#) <sup>[1203]</sup>

#### 6.4.1.41 System Boundary



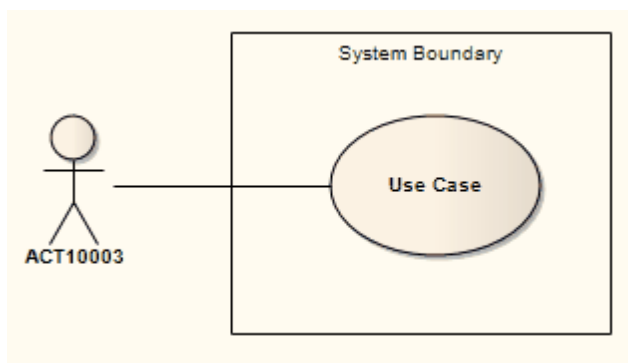
##### Description

A **System Boundary** element is a non-UML element used to define **conceptual** boundaries. You can use System Boundaries to help group logically related elements (from a visual perspective, not as part of the UML model).

In the *UML Superstructure Specification, v2.1.1*, System Boundaries are described in the sections on Use Cases, because the System Boundary is often used to indicate the **application** of a Use Case to another entity. In this context, the System Boundary:

- Encloses the Use Case, and
- Is associated with a classifier such as a Class, Component or Sub-system (Actor) through the Select <Item> dialog

By associating the System Boundary - and not the Use Case - with the classifier, the classifier is linked to the Use Case as a **user**, but not as an **owner**.



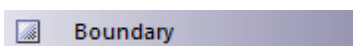
You can also define a Use Case as the classifier of a System Boundary element, to link the elements enclosed in the System Boundary (such as parts of an Activity diagram) to their representation in a logical Use Case.

The element properties for a System Boundary element comprise the name, the border style, and the number of horizontal or vertical swim lanes. You can also change the overall shape of the System Boundary.

A System Boundary element can be marked as **Selectable**, using the element's context menu. When the element is not selectable, you can click on the other elements within the System Boundary space without activating or selecting the System Boundary itself.

**Notes**

- A System Boundary is the basis for the Image element, which enables you to add icons or backgrounds to a diagram, automatically displaying the Image Manager dialog from which to select the appropriate image
- A **System Boundary** is not the same as the **Boundary** element used to capture user interactions in, for example, Analysis diagrams

**Toolbox icon****Learn more**

- [Use Cases](#) <sup>[1352]</sup>
- [Mapping Use Cases to Activity Diagrams](#) (Online resource)
- [Instance Classifier](#) <sup>[994]</sup>
- [System Boundary Properties](#) <sup>[1348]</sup>
- [Boundary](#) <sup>[1997]</sup>

**OMG UML Specification**

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 594*) states:

*If a subject (or system boundary) is displayed, the Use Case ellipse is visually located inside the system boundary rectangle. Note that this does not necessarily mean that the subject classifier owns the contained Use Cases, but merely that the Use Case applies to that classifier.*

**6.4.1.41.1 System Boundary Properties**

The System Boundary element has a small set of properties that are mainly concerned with the appearance of the element. You can also apply other element control options such as default appearance, locking the element and applying an image to the element.

**Access**    **Right-click on the System Boundary element and select:**  
                  **Properties**    or  
                  **Appearance | Shape**

**Set System Boundary Properties**

Field/Option/Button	Action	See Also
<b>Name</b>	(Optional) Type a name for the element.	
<b>Border Style</b>	Select the radio button for the style you prefer: <ul style="list-style-type: none"> <li>• <b>Solid</b> - a solid line border with the system default element fill</li> </ul>	

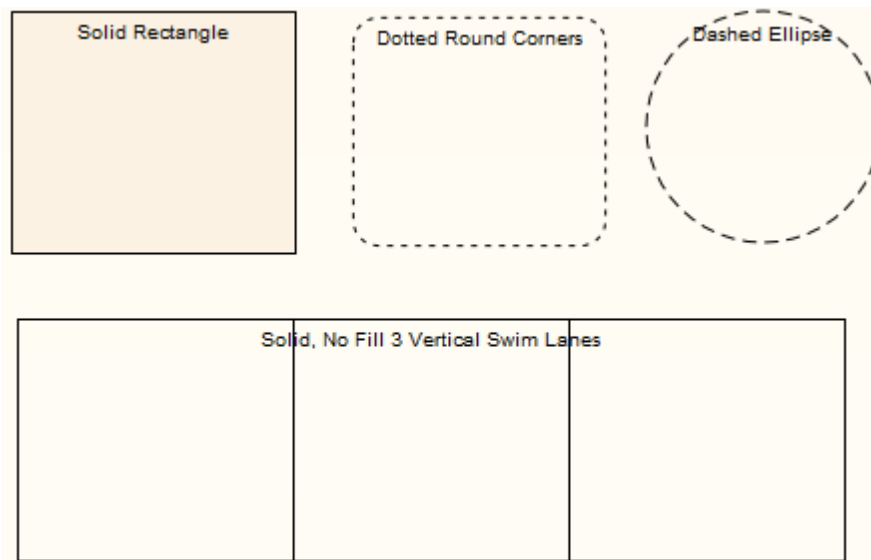
Field/Option/Button	Action	See Also
	<p>color</p> <ul style="list-style-type: none"> <li>• <b>Dotted</b> - a dotted line border with no element fill color</li> <li>• <b>Dashed</b> - a broken line border with no element fill color</li> <li>• <b>Solid-No Fill</b> - a solid line border with no element fill color</li> </ul>	
<b>Horizontal Swim Lanes</b>	<p>Type in the number of horizontal segments you want to divide the element into, to group the elements in the System Boundary in a horizontal context (for example, Client, Application and Database tiers could be represented in swim lanes).</p> <p>The field defaults to <b>1</b>.</p> <p>The swim lanes are equal divisions of the System Boundary - you cannot change their relative heights.</p>	
<b>Vertical Swim Lanes</b>	<p>Type in the number of vertical segments you want to divide the element into, to group the elements in the System Boundary in a vertical context.</p> <p>The field defaults to <b>1</b>.</p> <p>The swim lanes are equal divisions of the System Boundary - you cannot change their relative widths.</p>	

### Set System Boundary Shape

The **Shape** menu has three options. Select:

- **Rectangle** - to return the shape to the default rectangular border with sharp corners
- **Rounded Rectangle** - to set the shape to a rectangle with rounded corners
- **Ellipse** to set the shape to a circle or oval to accommodate the enclosed elements

### Example Shapes



#### Learn more

- [Operations on Elements](#)<sup>[939]</sup>

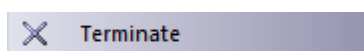
### 6.4.1.42 *Terminate*



#### Description

The Terminate pseudo-state indicates that upon entry of its pseudo-state, the State Machine's execution ends.

#### Toolbox icon



#### Learn more

- [State Machine Diagram](#)<sup>[1203]</sup>
- [Pseudo-state](#)<sup>[1208]</sup>

### 6.4.1.43 *Trigger*



### Description

A Trigger indicates an event that initiates an action (and might arise from completion of a previous action). You initially define a Trigger in one of four ways:

- As a property of a Transition relationship
- As a property of an Accept Event Action (on the Triggers tab of the element Properties dialog)
- As an event in a State Machine Table
- Directly, as a Trigger element, through the New Element dialog or Diagram Toolbox (State Additional page)

When you save the Trigger, it is added to the list of elements for the parent package in the Project Browser. You can then right-click on it and select the **Properties** context menu option to view and, if required, edit its properties as an element rather than as a property itself. Triggers created as events remain as Event elements, whilst Triggers created in other ways are Trigger elements, with a Trigger tab in the Properties dialog.

Field	Action	See also
<b>Type</b>	<p>If necessary, edit the type of trigger:</p> <ul style="list-style-type: none"> <li>• <b>Call</b> - specifies that the event is a CallEvent, which sends a message to the associated object by invoking an operation</li> <li>• <b>Change</b> - specifies that the event is a ChangeEvent, which indicates that the transition is the result of a change in value of an attribute</li> <li>• <b>Signal</b> - specifies that the event is a SignalEvent, which corresponds to the receipt of an asynchronous signal instance</li> <li>• <b>Time</b> - corresponds to a TimeEvent; which specifies a moment in time</li> </ul>	
<b>Specification</b>	<p>Either type in the event instigating the Trigger, or click on the ( ... ) button and select the event (depending on the <b>Type</b> value).</p>	
<b>Ports</b>	<p>Click on the <b>Add</b> button and select the appropriate Port from the Select Port dialog.</p> <ul style="list-style-type: none"> <li>• To create new Ports using the Select Port dialog, the Trigger should be created as a child of a Class or Component element</li> <li>• To add several Ports at once, press ( <b>Ctrl</b> ) as you select each Port</li> <li>• To check the exact location of a Port, right-click on the Port name and select the <b>Find in Project Browser</b> context menu option</li> </ul>	<a href="#">Select Port Dialog</a> <sup>[994]</sup>

### Notes

- You can also drag an existing Trigger element onto another diagram, although there are limited uses

for the element in that context

- This element is not the same as a Trigger Operation, which is an operation automatically executed as a result of the modification of data in a database

#### Toolbox icon



#### Learn more

- [Action](#)<sup>[1266]</sup>
- [State Toolbox](#)<sup>[807]</sup>
- [Transition Relationship](#)<sup>[1446]</sup>
- [Insert Trigger](#)<sup>[1218]</sup>
- [Triggers](#)<sup>[2368]</sup> (as an Operation)

#### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 456) states:

*Events may cause execution of behavior (e.g., the execution of the effect activity of a transition in a state machine). A trigger specifies the event that may trigger a behavior execution as well as any constraints on the event to filter out events not of interest.*

### 6.4.1.44 Use Case



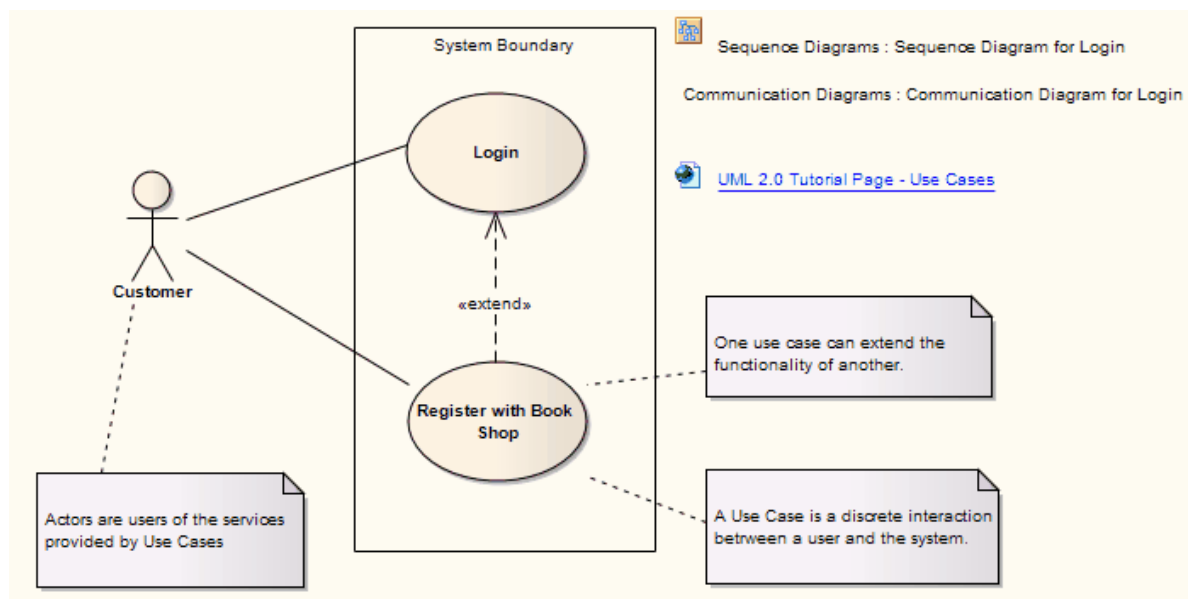
#### Description

A **Use Case** is a UML modeling element that describes how a user of the proposed system interacts with the system to perform a discrete unit of work. It describes and signifies a single interaction over time that has meaning for the end user (person, machine or other system), and is required to leave the system in a complete state: the interaction either completed or rolled back to the initial state. A Use Case:

- Typically has requirements and constraints that describe the essential features and rules under which it operates
- Can have an associated Sequence diagram illustrating behavior over time; who does what to whom, and when
- Typically has scenarios associated with it that describe the work flow over time that produces the end result; alternative work flows (for example, to capture exceptions) are also enabled

#### Example Use Case diagram

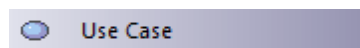




If extending a Use Case, you can specify the points of extension with Use Case Extension Points. To display the attributes, operations or constraints of a Use Case on a diagram, use Rectangle Notation.

Enterprise Architect also provides two stereotyped Use Cases: the Test Case and the Business Use Case.

#### Toolbox icon



#### Learn more

- [Sequence Diagram](#) <sup>[1249]</sup>
- [Use Case Extension Points](#) <sup>[1354]</sup>
- [Rectangle Notation](#) <sup>[1355]</sup>
- [Test Case](#) <sup>[2010]</sup>
- [Business Use Case](#) <sup>[1805]</sup>

#### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 592) states:

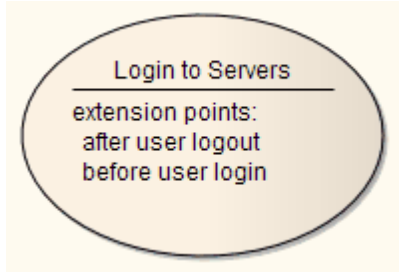
*A UseCase is a kind of behaviored classifier that represents a declaration of an offered behavior. Each Use Case specifies some behavior, possibly including variants, that the subject can perform in collaboration with one or more actors.*

#### 6.4.1.44.1 Use Case Extension Points

The behavior defined for a Use Case can **add to** the behavior of another Use Case; that is, the first Use case **extends** the second one. This is represented on the model by an **Extend** connector from the first Use Case to the second. If the extended behavior takes effect at a specific point, you can define that point as an **extension point** on the extended Use Case. The name (description) text of the extension point can be as informal or precise as is appropriate to define the point in behavior at which the extension applies. A Use Case can have more than one extension point, to allow for different source Use Cases to extend this target Use Case, or for changes in where the extending behavior applies depending on the constraints defined for the Extend connector. The connector also identifies which extension point is in effect.

**Access** [Right-click on extended Use Case element | Advanced | Edit Extension Points](#)

##### Add extension points to a Use Case

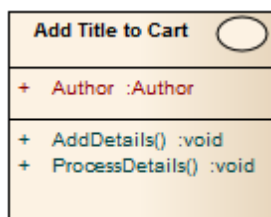
Field/Button	Action
<b>Defined Extension Points</b>	Lists the extension points currently defined for the selected Use Case.
<b>Add</b>	Click on this button to display a prompt for the name of a new extension point. Type the name and click on the <b>OK</b> button. The name is added to the Defined Extension Points list.
<b>Edit</b>	Click on an existing extension point and click on this button to display a prompt for changes to the name of the selected extension point. Overtyping the name and clicking on the <b>OK</b> button. The name is updated in the Defined Extension Points list.
<b>Remove</b>	Click on an existing extension point and click on this button to immediately remove the name from the Defined Extension Points list.
<b>OK</b>	Click on this button to save all changes to the extension points, and to close the dialog. The extension points you have defined are represented on the Use Case element in the diagram as shown. 

Learn more

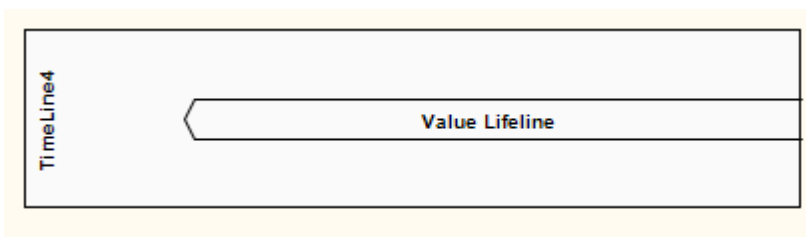
- [Use Case](#)<sup>[1352]</sup>
- [Extend](#)<sup>[1406]</sup>

**6.4.1.44.2 Rectangle Notation**

You can display various shaped elements, such as an Interface, Use Case or Actor, using *rectangle notation*. This displays the element as a rectangle, with an icon of the 'normal' shape in the top right-hand corner. Any attributes, operations or constraints belonging to the element are shown, in the same style as a Class.



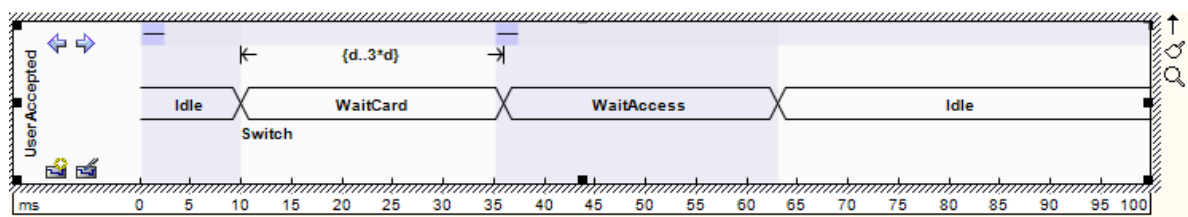
To show an element using rectangle notation, right-click on the element on the diagram and select the **Advanced | Use Rectangle Notation** context menu option. This setting only applies to the selected element, and can be toggled on and off either by deselecting the context menu option or by selecting the reciprocal option such as **Use Circle Notation** or **Use Actor Notation**.

**6.4.1.45 Value Lifeline**Description

A Lifeline is the path an object takes across a measure of time, indicated by the x-axis. There are two sorts: **Value Lifelines** (defined here) and State Lifelines, both used in Timing diagrams.

A Value Lifeline shows the Lifeline's state across the diagram, with parallel lines indicating a steady state. A cross between the lines indicates a transition or change in state.

An example of a Value Lifeline is shown below:



See *UML Superstructure Specification, v2.1.1, Figure 14.30, p. 520*.

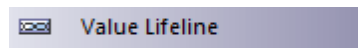
A Value Lifeline consists of a set of transition points. Each transition point can be defined with the following properties:

Property	Description
<b>At time</b>	Specifies the starting time for a change of state.
<b>Transition to</b>	Indicates the state to which the Lifeline is to change.
<b>Event</b>	Describes the occurring event.
<b>Timing constraints</b>	Refers to the time taken for a state to change within a Lifeline, or the time taken to transmit a message.
<b>Timing observations</b>	Provides information on the time of a state change or sent message.
<b>Duration constraints</b>	Pertains to a Lifeline's period at a particular state. The constraint could be instigated by a change of state within a Lifeline, or that Lifeline's receipt of a message.
<b>Duration observations</b>	Indicates the interval of a Lifeline at a particular state, begun from a change in state or message receipt.

In the example diagram above, the **10ms** transition point has these properties:

Property	Text
<b>At Time</b>	10ms
<b>Transition to</b>	Waitcard
<b>Event</b>	Switch
<b>Timing constraints</b>	–
<b>Timing observations</b>	–
<b>Duration constraints</b>	d..3*d

Property	Text
<b>Duration observations</b>	–

**Toolbox icon****Learn more**

- [Timing Diagrams](#)<sup>[1225]</sup>
- [State Lifelines](#)<sup>[1335]</sup>

**OMG UML Specification:**

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 518*) states:

*Shows the value of the connectable element as a function of time. Value is explicitly denoted as text. Crossing reflects the event where the value changed.*

## 6.4.2 Structural Diagram Elements

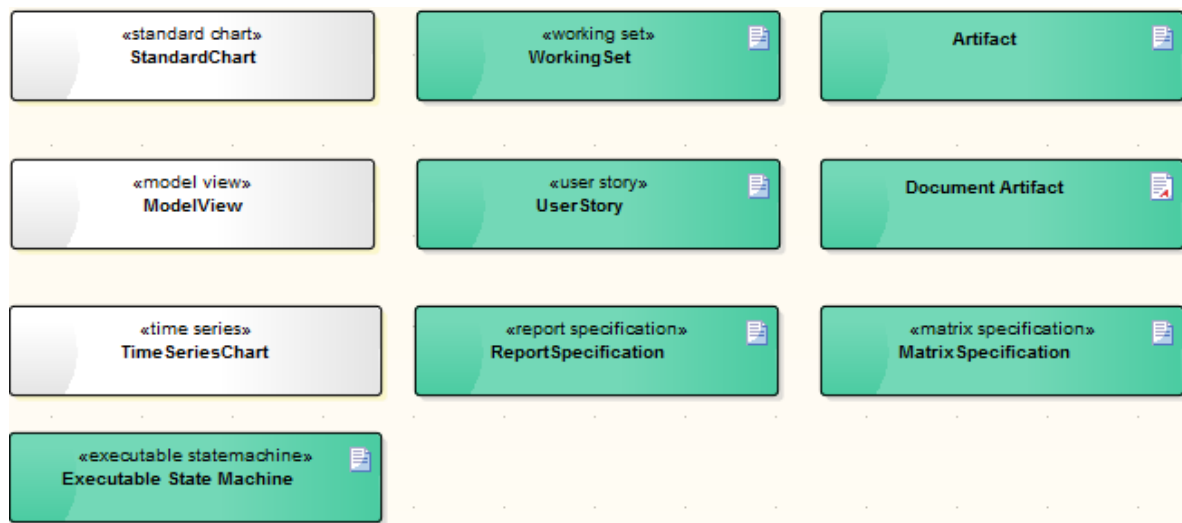
The following elements are used in UML Structural Diagrams. For more information on using each element, click on the element name in this list:

- [Actor](#)<sup>[1284]</sup>, [Artifact](#)<sup>[1358]</sup>
- [Class](#)<sup>[1363]</sup>, [Collaboration](#)<sup>[1367]</sup>, [Collaboration Use](#)<sup>[1368]</sup>, [Component](#)<sup>[1370]</sup>
- [Data Type](#)<sup>[1370]</sup>, [Deployment Specification](#)<sup>[1371]</sup>, [Document Artifact](#)<sup>[1373]</sup>
- [Enumeration](#)<sup>[1374]</sup>, [Execution Environment](#)<sup>[1374]</sup>, [Expose Interface](#)<sup>[1375]</sup>
- [Information Item](#)<sup>[1376]</sup>, [Interface](#)<sup>[1377]</sup>
- [Node](#)<sup>[1378]</sup>, [Note](#)<sup>[1324]</sup>
- [Object](#)<sup>[1379]</sup>
- [Package](#)<sup>[1382]</sup>, [Part](#)<sup>[1383]</sup>, [Port](#)<sup>[1384]</sup>, [Primitive](#)<sup>[1385]</sup>
- [Qualifiers](#)<sup>[1395]</sup>
- [Signal](#)<sup>[1387]</sup>

**Learn more**

- [Structural Diagrams](#)<sup>[1182]</sup>

### 6.4.2.1 Artifact





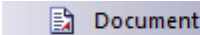
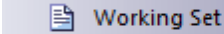
#### Description




An **Artifact** is any physical piece of information used or produced by a system. In Enterprise Architect these are represented by the **Artifact element**, which can have one of a number of stereotypes to tailor it to a specific purpose, including internal operations and structures within the model, as indicated in the examples. Artifacts can have associated properties or operations, and can be instantiated or associated with other Artifacts according to the object they represent.

You can create an Artifact element by dragging one of the Artifact icons from the **Artifacts** page of the Diagram Toolbox, or from one of several other Toolbox pages according to type. The Common page of the Toolbox also has a generic Artifact icon that - when you drag it onto a diagram - offers a choice of types of Artifact to create.




#### Types of Artifact

Type	Description	See also
<b>Base Artifact</b>	<p>A <b>Base Artifact</b> defines the external artifacts used in a process and the internal artifacts generated in the process, such as model files, source files, database tables, development deliverables or support documents. The files represented by the Artifact are listed on the Files tab of the element Properties dialog.</p> <p>To open the files represented by the Artifact, click on the element on the diagram and press <b>Ctrl+E</b>. Each file is opened either on a separate tab in the Diagram View workspace (if the file can be opened within Enterprise Architect) or in the default Windows viewer/editor for the file type (if the file cannot be opened within Enterprise Architect).</p> <p>Files can also be launched individually from the Files tab (opening in the Windows default editor), as for elements of any other type that have associated files.</p> <p><b>Toolbox Icon</b></p>	<p><a href="#">Create File Artifacts</a> <sup>[1362]</sup></p> <p><a href="#">Associated Files</a> <sup>[988]</sup></p> <p><a href="#">Generate, Save and Load (Profiler) Reports</a> <sup>[2562]</sup></p>

Type	Description	See also
	 <b>Artifact</b>	
<b>Document Artifact</b>	<p>A <b>Document Artifact</b> is an Artifact having a stereotype of «<i>document</i>». You create the Document Artifact using the Artifacts, Component, Documentation or Deployment pages of the Diagram Toolbox, and associate it with an RTF document or CSV file.</p> <p>Double-click on the element to display the Linked Document Editor. When you have created the linked document, the Document Artifact element on the diagram shows an <b>A</b> symbol in the bottom right corner.</p>  <p><b>Toolbox icon</b></p> 	<p><a href="#">Linked Documents</a> <sup>[1044]</sup></p> <p><a href="#">Add Quick Linker Definition To Profile</a> <sup>[1523]</sup></p>
<b>Working Set Artifact</b>	<p>A <b>Working Set Artifact</b> defines a Working Set that opens various windows, diagrams and views, recreating a work environment that you frequently use.</p> <ul style="list-style-type: none"> <li>To create or modify the Working Set, right-click on the element and select the <b>Edit Working Set</b> option</li> <li>To execute the Working Set to open the defined windows and views and execute any commands, double-click on the element</li> </ul> <p><b>Toolbox icon</b></p> 	<p><a href="#">Working Sets</a> <sup>[561]</sup></p>
<b>Executable State Machine Artifact</b>	<p>An <b>Executable State Machine Artifact</b> is the vehicle through which you can generate, build (compile) and execute - via simulation - code for a State Machine or complex of State Machines.</p> <p>Each State Machine is the child of a Class element; when you drag the Class from the Project Browser onto the Artifact element, it is pasted inside the Artifact as a <b>Part</b>. You can paste several Classes - and, therefore, Parts - into a single Artifact.</p> <p>Having set up the Executable State Machine Artifact, you use simple context menu options on the Artifact to perform the code generation, build and execution operations on all State Machines bound within the Artifact.</p> <p><b>Toolbox icon</b></p>	<p><a href="#">Code Generation - State Machines</a> <sup>[2122]</sup></p>

Type	Description	See also
	 Executable StateMachine	
<b>Matrix Specification</b>	<p>A <b>Matrix Specification Artifact</b> encapsulates a <b>Relationship Matrix Profile definition</b>. When you have created the element on the diagram, you double-click on it to display the Matrix Specification dialog, in which you create the Profile definition. The Profile takes the name of the element.</p> <p>After you create the Profile definition, each time you double-click on the Artifact element the Relationship Matrix displays with the Profile applied.</p> <p>To edit the Profile you can either open it through the Relationship Matrix, or right-click on the Artifact and select the <b>Documentation   Edit Matrix Profile</b> option.</p> <p><u>Toolbox icon</u></p>  Matrix Specification	<a href="#">Create Matrix Profile</a> <sup>[1744]</sup>
<b>Report Specification</b>	<p>A <b>Report Specification Artifact</b> encapsulates a <b>report definition</b>. When you have created the element on the diagram, you double-click on it to display the Generate Documentation dialog, on which you enter the report parameters and, if you wish, generate the report.</p> <p>After you create the Report Specification, each time you double-click on the Artifact element the Generate Documentation dialog again displays with the same report parameters. You can continue to generate the same report, or alter the parameters if necessary. If you change the parameters, they are re-presented until such time as you change them again,</p> <p><u>Toolbox icon</u></p>  Report Specification	<a href="#">Generate Documentation</a> <sup>[2644]</sup>
<b>User Story</b>	<p>A <b>User Story Artifact</b> provides a means of documenting a business <b>Use Case</b> in the context of Agile methodologies such as <b>Extreme Programming (XP)</b>. In the Linked Document, you define the functions a business system must provide; it captures the 'who', 'what' and 'why' of a requirement in a simple, concise format. The User Story Artifact behaves as a Document Artifact (above), prompting you to select a Linked Document template to base the document on.</p> <p><u>Toolbox icon</u></p>  User Story	



Type	Description	See also
<b>Standard Chart</b>	<p>A <b>Standard Chart Artifact</b> provides the facilities for generating a <b>Pie Chart</b> or <b>Bar Chart</b> on an aspect of the data in your model. It adds three Chart Details pages to the standard pages of the element Properties dialog.</p> <p>After you have added the element to your diagram, double-click on it. The element Properties dialog automatically opens at the Chart Details - Source page. Define the chart type and data source, then go on to define any filters you want to apply, and how the chart should display.</p> <p>Once you have defined the chart, it automatically displays with the latest information whenever you open the parent diagram.</p> <p><b>Toolbox icon</b></p>  Standard Chart	<p><a href="#">Standard Chart Data</a> <sup>[2770]</sup></p> <p><a href="#">Chart Appearance</a> <sup>[2774]</sup></p>
<b>Model View</b>	<p>A <b>Model View Artifact</b> provides the facilities for generating a tabular <b>Model View Chart</b> on a segment of the data in your mode, extracted using a custom SQL search.</p> <p>After you have added the element to your diagram, double-click on it. The element Properties dialog automatically opens at the Chart Details - Source page. Define the SQL Search to extract and tabulate the information.</p> <p>Once you have defined the chart, it automatically displays with the latest information whenever you open the parent diagram.</p> <p><b>Toolbox icon</b></p>  Model View	<p><a href="#">Define a Model View Chart</a> <sup>[2765]</sup></p>
<b>Time Series Chart</b>	<p>A <b>Time Series Chart Artifact</b> provides the facilities for generating a linear graph of a model property over time.</p> <p>After you have added the element to your diagram, double-click on it. The element Properties dialog automatically opens at the Chart Details - Source page. Define the Package from which the data is to be extracted, and the time interval over which the data is to be sampled. Then go on to define the appearance of the chart.</p> <p>Once you have defined the chart, it automatically displays with the latest information whenever you open the parent diagram.</p> <p><b>Toolbox icon</b></p>  Time Series Chart	<p><a href="#">Define a Time Series Chart</a> <sup>[2767]</sup></p>

Type	Description	See also

### Learn more

- [Deployment Diagram](#) 

### OMG UML Specification

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 201) states:

*An Artifact defined by the user represents a concrete element in the physical world. A particular instance (or 'copy') of an artifact is deployed to a node instance. Artifacts may have composition associations to other artifacts that are nested within it. For instance, a deployment descriptor artifact for a component may be contained within the artifact that implements that component. In that way, the component and its descriptor are deployed to a node instance as one artifact instance.*

#### 6.4.2.1.1 Create File Artifacts

A File Artifact is an Artifact element that represents and is linked to a file. You can create the Artifact element on a diagram, from the file itself.

#### Create the Artifact

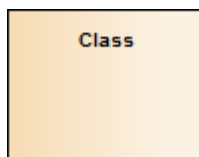
Step	Action	See also
1	<p>Locate the file in a file list (such as Windows Explorer) or on your Desktop. Drag the file onto your diagram.</p> <p>A context menu displays.</p>	
2	<p>Click on the menu option you need:</p> <ul style="list-style-type: none"> <li>• <b>Hyperlink</b> - to create a Hyperlink element on the diagram; you can select a sub-option to define whether users, when they double-click on the Hyperlink, will either just display the file content or open it within the appropriate file editor</li> <li>• <b>Artifact External</b> - to create an Artifact element on the diagram; the element Properties dialog displays, in which you enter any element properties you need</li> </ul> <p>Save the data you have entered and close the Properties dialog (note that the file name becomes the element name)</p> <p>If you double-click on the Artifact the <b>Properties dialog redisplay</b>s; click on the Files tab to see the file pathname listed in the Files panel, from which you can launch it in its registered application</p> <ul style="list-style-type: none"> <li>• <b>Artifact Internal</b> - to immediately create an Artifact element on the diagram</li> </ul>	

Step	Action	See also
	<p>with the file name as the element name</p> <p>The file is stored in your model, but is managed by the registered external application for the file type; if you double-click the Artifact, the <b>file is opened</b> within its external application</p> <p>If the file is changed, you are prompted to update the element within the model - click on the <b>Save</b> button to update the element, or the <b>Discard</b> button to ignore the changes</p> <ul style="list-style-type: none"> <li>• <b>Insert</b> - (graphics files) to insert the file into the diagram as a filled Boundary element; double-click on the image to display the Boundary Properties dialog</li> </ul>	

#### Learn more

- [Artifact](#) <sup>[1358]</sup>
- [Hyperlinks To Files](#) <sup>[2004]</sup>

### 6.4.2.2 Class

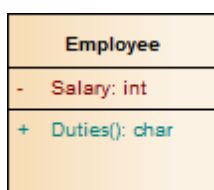


#### Description

A Class is a representation of a type of object that reflects the structure and behavior of such objects within the system. It is a template from which actual running instances are created, although a Class can be defined either to control its own execution or as a template or parameterized Class that specifies parameters that must be defined by any binding Class.

A Class can have *attributes* (data) and *methods* (*operations* or behavior). Classes can inherit characteristics from parent Classes and delegate behavior to other Classes. Class models usually describe the logical structure of the system and are the building blocks from which components are built.

The top section of a Class shows the attributes (or data elements) associated with the Class. These hold the 'state' of an object at run-time. If the information is saved to a data store and can be reloaded, it is termed 'persistent'. The lower section contains the Class operations (or methods at run-time). Operations describe the behavior a Class offers to other Classes, and the internal behavior it has (private methods).



Class elements are generally used in Class diagrams and Composite Structure diagrams.

Enterprise Architect also supports a number of stereotyped Class elements to represent various entities in web-page modeling. A Class can also be integrated with an Associate connector to form an Association Class, to allow the Associate connector to have operations and attributes that define certain types of UML relationship.

#### Toolbox icon



#### Learn more

- [Active Classes](#) <sup>[1365]</sup>
- [Parameterized Classes \(Templates\)](#) <sup>[1365]</sup>
- [Class Diagrams](#) <sup>[1184]</sup>
- [Composite Structure Diagram](#) <sup>[1188]</sup>
- [Association Class](#) <sup>[1398]</sup>
- [Association](#) <sup>[1393]</sup>
- [Attributes](#) <sup>[999]</sup>
- [Operations](#) <sup>[1014]</sup>
- [Web-page Modeling](#) <sup>[1995]</sup>

#### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, pp. 52-53*) states:

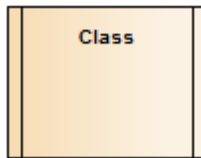
*The purpose of a class is to specify a classification of objects and to specify the features that characterize the structure and behavior of those objects.*

*Objects of a class must contain values for each attribute that is a member of that class, in accordance with the characteristics of the attribute, for example its type and multiplicity.*

*When an object is instantiated in a class, for every attribute of the class that has a specified default, if an initial value of the attribute is not specified explicitly for the instantiation, then the default value specification is evaluated to set the initial value of the attribute for the object.*

*Operations of a class can be invoked on an object, given a particular set of substitutions for the parameters of the operation. An operation invocation may cause changes to the values of the attributes of that object. It may also return a value as a result, where a result type for the operation has been defined. Operation invocations may also cause changes in value to the attributes of other objects that can be navigated to, directly or indirectly, from the object on which the operation is invoked, to its output parameters, to objects navigable from its parameters, or to other objects in the scope of the operation's execution. Operation invocations may also cause the creation and deletion of objects.*

### 6.4.2.2.1 Active Classes



#### Description

An *Active Class* indicates that, when instantiated, the Class controls its own execution. Rather than being invoked or activated by other objects, it can operate standalone and define its own thread of behavior.

#### Define an Active Class in Enterprise Architect

Step	Action
1	Highlight a Class, and display its Properties dialog.
2	Select the Details page.
3	Select the <b>Is Active</b> checkbox.
4	Click on the <b>OK</b> button to save the changes.

#### OMG UML Specification

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 438) states:

*An active object is an object that, as a direct consequence of its creation, commences to execute its classifier behavior, and does not cease until either the complete behavior is executed or the object is terminated by some external object. (This is sometimes referred to as "the object having its own thread of control.") The points at which an active object responds to communications from other objects is determined solely by the behavior of the active object and not by the invoking object. If the classifier behavior of an active object completes, the object is terminated.*

### 6.4.2.2.2 Parameterized Classes (Templates)

#### Description

Enterprise Architect supports **template** or **parameterized Classes**, which specify parameters that must be defined by any binding Class.

Parameterized Classes are commonly implemented in C++; Enterprise Architect imports and generates templated Classes for C++.

A template Class enables its functionality to be reused by any bound Class. If a default value is specified for

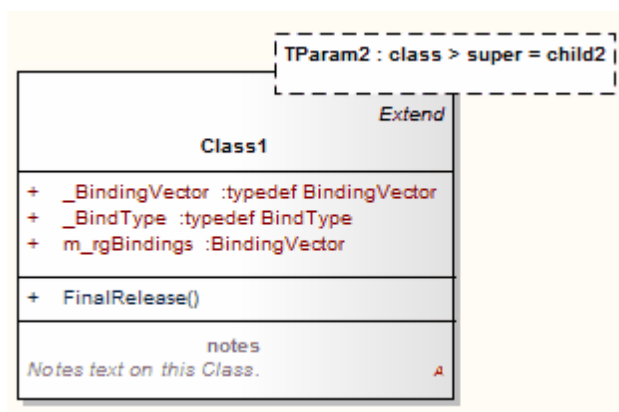
a parameter, and a binding Class doesn't provide a value for that parameter, the default is used.

### Create a parameterized Class

Step	Action
1	Display the Properties dialog for the required Class.
2	Select the Templates page.
3	In the Template Parameter(s) panel, click on the <b>Add</b> button. The Template Parameter dialog displays.
4	Type in the name and type of the parameter and, if required, click on the ( ... ) buttons after the <b>Constraints</b> and <b>Default</b> fields to select the required constraining and default Classes from the Select <Item> dialog.  The default Class can be either the constraining classifier or any Class that derives from the constraining classifier.

### Notation Example

On a diagram, template Classes are shown with the parameters in a dashed outline box in the upper right corner of the Class.



### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 622*) states:

*A template is a parameterized element that can be used to generate other model elements using TemplateBinding relationships. The template parameters for the template signature specify the formal parameters that will be substituted by actual parameters (or the default) in a binding.*

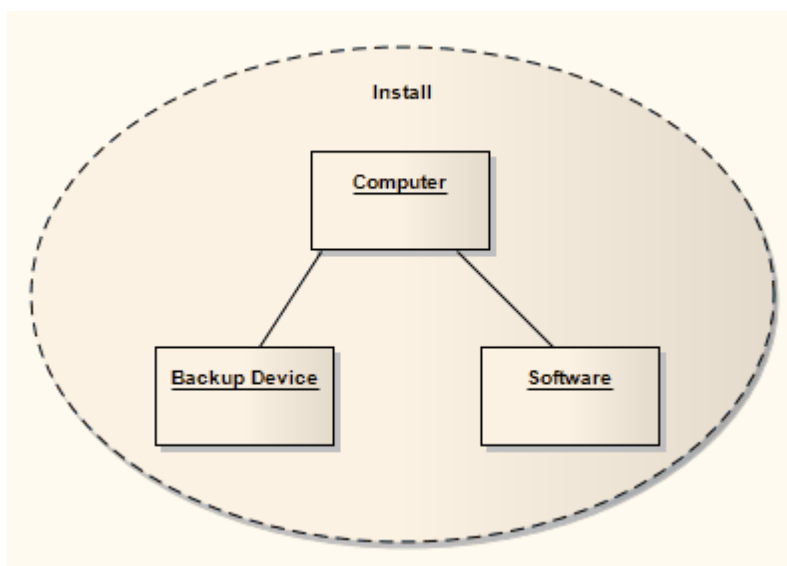
### 6.4.2.3 Collaboration



#### Description

A *Collaboration* defines a set of cooperating roles and their connectors. These are used to collectively illustrate a specific functionality, in a Composite Structure diagram. A Collaboration should specify only the roles and attributes required to accomplish a specific task or function. Although in practice a behavior and its roles could involve many tangential attributes and properties, isolating the primary roles and their requisites simplifies and clarifies the behavior, as well as providing for reuse. A Collaboration often implements a pattern to apply to various situations.

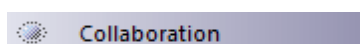
The following example illustrates an *Install* Collaboration, with three roles (Objects) connected as shown. The process for this Collaboration can be demonstrated by attaching an Interaction diagram (Sequence, Timing, Communication or Interaction Overview).



To understand referencing a Collaboration in a specific situation, see the *Collaboration Use* topic.

Enterprise Architect supports a stereotyped Collaboration to represent a Business Use Case Realization in business modeling.

#### Toolbox icon



**Learn more**

- [Composite Structure Diagram](#) <sup>[1188]</sup>
- [Sequence Diagram](#) <sup>[1249]</sup>
- [Timing Diagram](#) <sup>[1225]</sup>
- [Communication Diagram](#) <sup>[1259]</sup>
- [Interaction Overview Diagram](#) <sup>[1262]</sup>
- [Collaboration Use](#) <sup>[1368]</sup>
- [Object Element](#) <sup>[1379]</sup>
- [Business Use Case Realization](#) <sup>[1805]</sup>

**OMG UML Specification:**

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 171*) states:

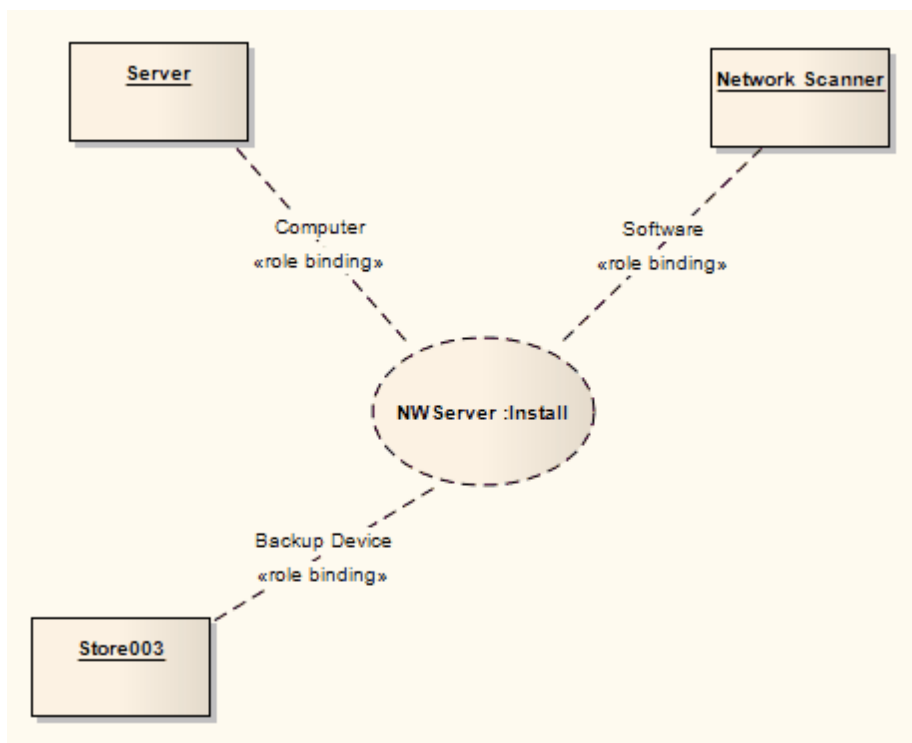
*A collaboration describes a structure of collaborating elements (roles), each performing a specialized function, which collectively accomplish some desired functionality. Its primary purpose is to explain how a system works and, therefore, it typically only incorporates those aspects of reality that are deemed relevant to the explanation.*

**6.4.2.4 Collaboration Use****Description**

Use a *Collaboration Use* to apply a pattern defined by a Collaboration to a specific situation, in a Composite Structure diagram.

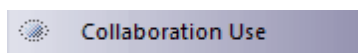
The following example shows a Use, *NWServer*, of the Collaboration *Install*, to define the installation process of a network scanner. This process can be defined by an interaction attached to the Collaboration. (See the *Collaboration* topic for a representation of the *Install* Collaboration.)





To create a Collaboration Use, drag the icon from the Toolbox onto the diagram.

#### Toolbox icon



#### Learn more

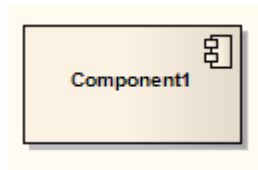
- [Composite Structure Diagram](#) <sup>[1188]</sup>
- [Collaboration Element](#) <sup>[1367]</sup>

#### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 173*) states:

*A collaboration use represents one particular use of a collaboration to explain the relationships between the properties of a classifier. A collaboration use shows how the pattern described by a collaboration is applied in a given context, by binding specific entities from that context to the roles of the collaboration. Depending on the context, these entities could be structural features of a classifier, instance specifications, or even roles in some containing collaboration. There may be multiple occurrences of a given collaboration within a classifier, each involving a different set of roles and connectors. A given role or connector may be involved in multiple occurrences of the same or different collaborations.*

### 6.4.2.5 Component

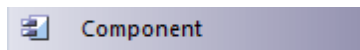


#### Description

A Component is a modular part of a system, whose behavior is defined by its provided and required interfaces; the internal workings of the Component should be invisible and its usage environment-independent. Source code files, DLLs, Java beans and other artifacts defining the system can be manifested in Components.

A Component can be composed of multiple Classes, or Components pieced together. As smaller Components come together to create bigger Components, the eventual system can be modeled, building-block style, in Component diagrams. By building the system in discrete Components, localization of data and behavior enables decreased dependency between Classes and Objects, providing a more robust and maintainable design.

#### Toolbox icon



#### Learn more

- [Component Diagram](#) <sup>1194</sup>
- [Class Element](#) <sup>1363</sup>
- [Object Element](#) <sup>1379</sup>

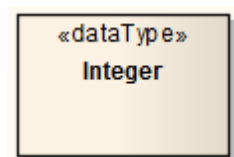
#### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 148*) states:

*A component represents a modular part of a system that encapsulates its contents and whose manifestation is replaceable within its environment.*

*A component defines its behavior in terms of provided and required interfaces. As such, a component serves as a type whose conformance is defined by these provided and required interfaces (encompassing both their static as well as dynamic semantics).*

### 6.4.2.6 Data Type



### Description

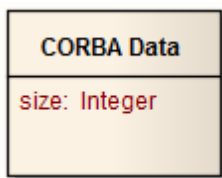
A Data Type is a specific kind of classifier, similar to a Class except that a Data Type cannot own sub Data Types, and instances of a Data Type are identified only by their value. For example, an instance of a Person Class is a Helen object, but an instance of an Integer Data Type is 12.

All copies of an instance of a Data Type, and any instances of that Data Type with the same value, are considered to be the same instance. That is, instances of Helen are not necessarily the same Helen, but all 12s are the same 12. For example, the 12 on a watch face is exactly the same integer as the number of months in a year.

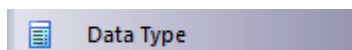
Instances of a Data Type that have attributes (that is, are instances of a structured Data Type) are considered to be the same if the structure is the same and the values of the corresponding attributes are the same. If a Data Type has attributes, instances of that Data Type contain attribute values matching the attributes.

A typical use of Data Types would be to represent programming language primitive types or CORBA basic types. For example, integer and string types are often treated as Data Types.

A Data Type is denoted by a rectangle with the keyword «dataType», as above or, when it is referenced by (for example) an attribute, by a string containing the name of the Data Type, as below:



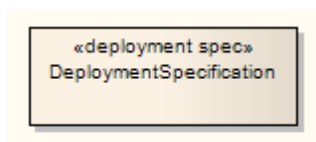
### Toolbox icon



### Learn more

- [Class Element](#)<sup>[1363]</sup>

## 6.4.2.7 Deployment Spec

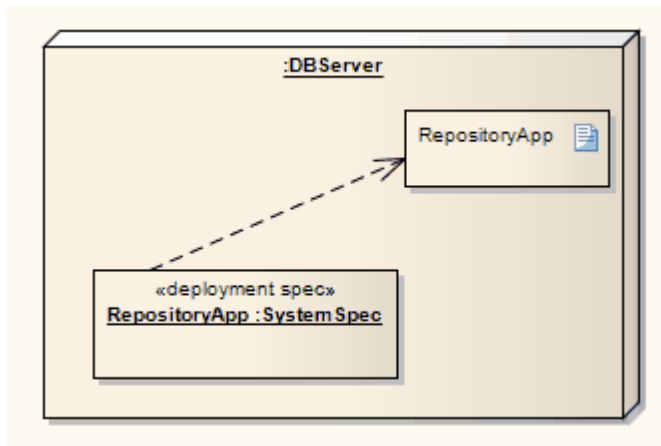


### Description

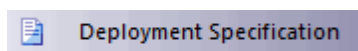
A Deployment Specification (spec) specifies parameters guiding deployment of an artifact, as is necessary with most hardware and software technologies. A specification lists those properties that must be defined for deployment to occur, as represented in a Deployment diagram. An instance of this specification specifies the values for the parameters; a single specification can be instantiated for multiple artifacts.

These specifications can be extended by certain component profiles. Examples of standard Tagged Values that a profile might add to a Deployment Specification are «*concurrencyMode*» with Tagged Values {*thread*, *process*, *none*} or «*transactionMode*» with Tagged Values {*transaction*, *nestedTransaction*, *none*}.

The following example depicts the artifact *RepositoryApp* deployed on the server node, as per the specifications of *RepositoryApp*, instantiated from the *Deployment Specification SystemSpec*.



#### Toolbox icon



#### Learn more

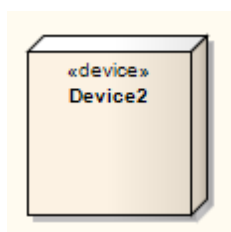
- [Deployment Diagram](#) 1197

#### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 206) states:

*A deployment specification specifies a set of properties that determine execution parameters of a component artifact that is deployed on a node. A deployment specification can be aimed at a specific type of container. An artifact that reifies or implements deployment specification properties is a deployment descriptor.*

### 6.4.2.8 Device



#### Description

A Device is a physical electronic resource with processing capability upon which Artifacts can be deployed for execution, as represented in a Deployment diagram. Complex Devices can consist of other devices; that is, a Device can be a nested element, where a physical machine is decomposed into its elements either through namespace ownership or through attributes that are typed by Devices.

#### Toolbox icon



#### Learn more

- [Deployment Diagram](#) <sup>[1191]</sup>
- [Artifact Element](#) <sup>[1358]</sup>

#### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, 10.3.7, v2.1.1, p. 207*) states:

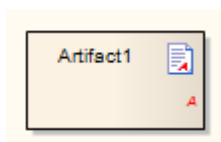
*In the metamodel, a Device is a subclass of Node.*

### 6.4.2.9 Document Artifact

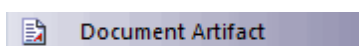


#### Description

A Document Artifact is an artifact having a stereotype of «*document*». You create the Document Artifact using the Artifact, Common, Component, Documentation or Deployment pages of the Diagram Toolbox, and associate it with a document or CSV file. Double-click on the element to display the Linked Document Editor. When you have created the linked document, the Document Artifact element on the diagram shows an **A** symbol in the bottom right corner.



#### Toolbox icon

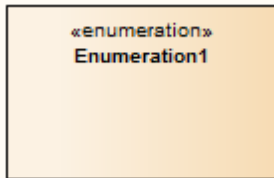


#### Learn more

- [Linked Documents](#) <sup>[1044]</sup>

- [Artifact Element](#) <sup>[1358]</sup>
- [Add Quick Linker Definition To Profile](#) <sup>[1523]</sup>

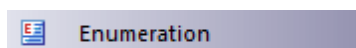
#### 6.4.2.10 Enumeration



##### Description

An *Enumeration* is a data type, whose instances can be any of a number of user-defined enumeration literals. It is possible to extend the set of applicable enumeration literals in other packages or profiles. You create Enumerations in Class or Package diagrams, and in diagrams developed using the Metamodel and Profile pages of the Diagram Toolbox.

##### Toolbox icon



##### Learn more

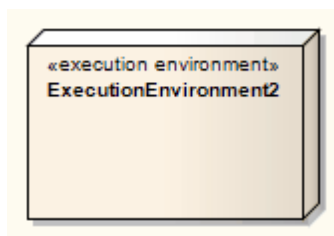
- [Class Diagram](#) <sup>[1184]</sup>
- [Package Diagram](#) <sup>[1182]</sup>

##### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 69) states:

*An enumeration is a data type whose values are enumerated in the model as enumeration literals.*

#### 6.4.2.11 Execution Environment

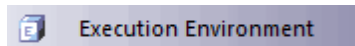


##### Description

An Execution Environment is a node that offers an execution environment for specific types of components that are deployed on it in the form of executable artifacts. This is depicted in a Deployment diagram.

Execution Environments can be nested; for example, a database Execution Environment can be nested in an operating system Execution Environment. Components of the appropriate type are then deployed to specific Execution Environment nodes.

### Toolbox icon



### Learn more

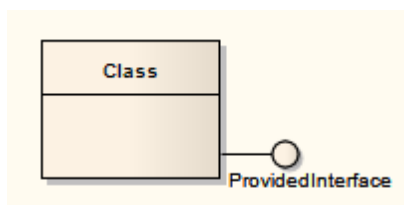
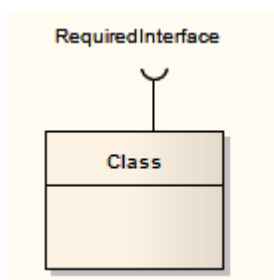
- [Deployment Diagram](#) <sup>[1191]</sup>
- [Node Element](#) <sup>[1378]</sup>
- [Component Element](#) <sup>[1370]</sup>
- [Artifact Element](#) <sup>[1358]</sup>

### OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 210*) states:

*... an ExecutionEnvironment is ... usually part of a general Node, representing the physical hardware environment on which the ExecutionEnvironment resides. In that environment, the ExecutionEnvironment implements a standard set of services that Components require at execution time (at the modeling level these services are usually implicit). For each component Deployment, aspects of these services may be determined by properties in a DeploymentSpecification for a particular kind of ExecutionEnvironment.*

## 6.4.2.12 Expose Interface



### Description

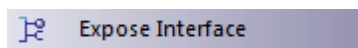
The Expose Interface element is a graphical method of depicting the required or supplied interfaces of a Component, Class or Part, in a Component or Composite Structure diagram. It just identifies the fact that the element provides or requires an interface; to depict the fact that the provided interface is used, or the

required interface provided, by another element using the Assembly connector.

The Expose Interface element must be attached to the Class or Component element, and it becomes a child element of that Class or Component; it cannot exist independently. You can attach more than one Expose Element to another element.

When you create the Expose Interface element, a dialog displays in which you enter a name for the element and specify whether it represents a required interface or a provided interface.

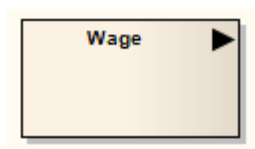
#### Toolbox icon



#### Learn more

- [Component Diagram](#) <sup>[1194]</sup>
- [Composite Structure Diagram](#) <sup>[1188]</sup>
- [Interface](#) <sup>[1377]</sup>
- [Component](#) <sup>[1370]</sup>
- [Class](#) <sup>[1363]</sup>
- [Part](#) <sup>[1383]</sup>
- [Assembly](#) <sup>[1393]</sup>

### 6.4.2.13 Information Item

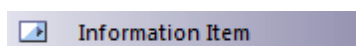


#### Description

An Information Item *element* represents an abstraction of data, which data can be conveyed between two objects. The *term* 'Information Item' is also more loosely applied to any classifier that represents a more specific identification of the type of data that can be conveyed between two objects

The conveyance and realization of Information Items (of either kind) between the two objects is represented by an Information Flow connector.

#### Toolbox icon



#### Learn more

- [Information Flow](#) <sup>[1410]</sup>

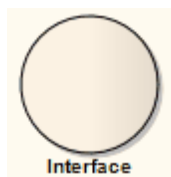


**OMG UML Specification:**

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 608) states:

*An information item is an abstraction of all kinds of information that can be exchanged between objects. It is a kind of classifier intended for representing information at a very abstract way, one which cannot be instantiated.*

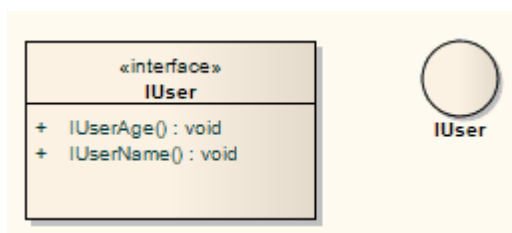
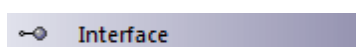
*One purpose of information items is to be able to define preliminary models, before having made detailed modeling decisions on types or structures. One other purpose of information items and information flows is to abstract complex models by a less precise but more general representation of the information exchanged between entities of a system.*

**6.4.2.14 Interface****Description**

An *Interface* is a specification of behavior (or contract) that implementers agree to meet. By implementing an Interface, *Classes* are guaranteed to support a required behavior, which enables the system to treat non-related elements in the same way; that is, through the common interface. You also use Interfaces in a Composite Structure diagram.

Interfaces are drawn in a similar way to a Class, with operations specified, as shown below. They can also be drawn as a circle with no explicit operations detailed - right-click on the element and select the **Use Circle Notation** context menu option to switch between styles. Realize connectors to an Interface drawn as a circle are drawn as a solid line without target arrows.

An Interface cannot be instantiated (that is, you cannot create an object from an Interface). You must create a Class that 'implements' the Interface specification, and in the Class body place operations for each of the Interface operations. You can then instantiate the Class.

**Toolbox icon****Learn more**

- [Class Diagram](#) <sup>1184</sup>

- [Composite Structure Diagram](#) <sup>[1188]</sup>
- [Realization](#) <sup>[1440]</sup>

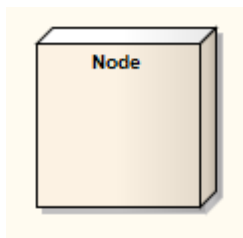
**OMG UML Specification:**

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 88) states:

*An interface is a kind of classifier that represents a declaration of a set of coherent public features and obligations. An interface specifies a contract; any instance of a classifier that realizes the interface must fulfill that contract. The obligations that may be associated with an interface are in the form of various kinds of constraints (such as pre- and post-conditions) or protocol specifications, which may impose ordering restrictions on interactions through the interface.*

*Since interfaces are declarations, they are not instantiable. Instead, an interface specification is implemented by an instance of an instantiable classifier, which means that the instantiable classifier presents a public facade that conforms to the interface specification. Note that a given classifier may implement more than one interface and that an interface may be implemented by a number of different classifiers.*

#### 6.4.2.15 Node

**Description**

A Node is a physical piece of equipment on which the system is deployed, such as a workgroup server or workstation. A Node usually hosts components and other executable pieces of code, which again can be connected to particular processes or execution spaces. Typical Nodes are client workstations, application servers, mainframes, routers and terminal servers.

Nodes are used in Deployment diagrams to model the deployment of a system, and to illustrate the physical allocation of implemented artifacts. They are also used in web modeling, from dedicated web modeling pages in the Toolbox.

**Toolbox icon****Learn more**

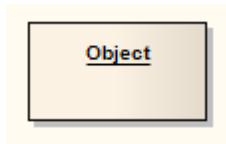
- [Deployment Diagram](#) <sup>[1191]</sup>

**OMG UML Specification:**

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 213) states:

*In the metamodel, a Node is a subclass of Class. It is associated with a Deployment of an Artifact. It is also associated with a set of Elements that are deployed on it. This is a derived association in that these PackageableElements are involved in a Manifestation of an Artifact that is deployed on the Node. Nodes may have an internal structure defined in terms of parts and connectors associated with them for advanced modeling applications.*

#### 6.4.2.16 Object

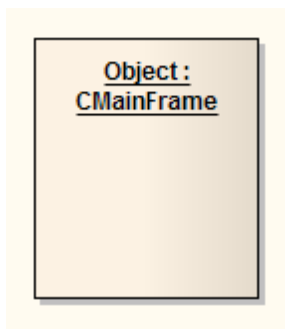


##### Description

An Object is a particular instance of a Class at run time. For example a car with the license plate AAA-001 is an instance of the general class of cars with a license plate number attribute. Objects are often used in analysis to represent the numerous artifacts and items that exist in any business, such as pieces of paper, faxes and information. To model the varying behavior of Objects at run-time, use run-time states.

Early in analysis, Objects can be used to quickly capture all the things that are of relevance within the system domain, in an Object, Composite Structure or Communication diagram. As the model progresses these analysis Objects are refined into generic Classes from which instances can be derived to represent common business items. Once Classes are defined, Objects can be typed; that is they can have a classifier set that indicates their base type - see the *Classifiers and Instances* topic.

Enterprise Architect also supports a number of stereotyped Object elements to represent various entities in business modeling.



##### Toolbox icon



##### Learn more

- [Object Diagram](#) <sup>[1186]</sup>
- [Composite Structure Diagram](#) <sup>[1188]</sup>
- [Communication Diagram](#) <sup>[1259]</sup>

- [Class](#) <sup>[1363]</sup>
- [Classifiers and Instances](#) <sup>[1009]</sup>
- [Business Interaction Objects](#) <sup>[1805]</sup>
- [Run-time State](#) <sup>[1380]</sup>
- [Define a Run-time Variable](#) <sup>[1380]</sup>
- [Remove a Defined Variable](#) <sup>[1381]</sup>
- [Object State](#) <sup>[1381]</sup>

#### 6.4.2.16.1 Run-time State

At run-time, an Object instance can have specific values for its attributes, or exist in a particular state. To model the varying behavior of Objects at run-time, use instance values selected from the Select <Item> dialog and run-time states or run-states.

Typically there is interest in the run-time behavior of Objects that already have a classifier set. You can select from the classifier's attribute list and apply specific values for your Object instance. If the classifier has a child State Machine, its States propagate to a list where the run-time state for the Object can be defined.

##### Example

The following example defines run-time values for the listed variables, which are attributes of the instances' classifier *AccountItem*.



##### Learn more

- [Object](#) <sup>[1379]</sup>
- [Object State](#) <sup>[1381]</sup>
- [Define a Run-Time Variable](#) <sup>[1380]</sup>
- [Remove a Defined Variable](#) <sup>[1381]</sup>

#### 6.4.2.16.1.1 Define a Run-time Variable

##### How to

To add run-time state instance variables to an Object

Step	Action
1	<p>Right-click on the Object and, if Instance Variables are supported, select the <b>Advanced   Set Run State</b> menu option (or press ( <b>Ctrl+Shift+R</b> ) ).</p> <p>The Set Run State dialog displays.</p>

Step	Action
2	In the <b>Variable</b> field, click on the drop-down arrow and select the variable, or type in the new variable name.
3	Set the <b>Operator</b> , the <b>Value</b> and optionally type in a <b>Note</b> .
4	Click on the <b>OK</b> button to save the variable.

#### 6.4.2.16.1.2 Remove a Defined Variable

##### How to

To delete a run-time state variable for an Object

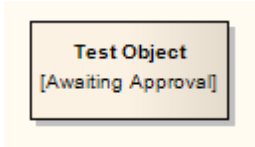
Step	Action
1	Right-click on the required Object and select the <b>Set Run State</b> context menu option. The Run State dialog displays.
2	In the <b>Variable</b> field, click on the drop-down arrow and select the variable to delete.
3	Clear the <b>Value</b> field.
4	Click on the <b>OK</b> button.

#### 6.4.2.16.2 Object State

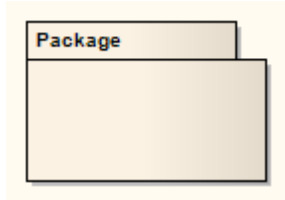
##### How to

To set the Object state for a Class instance

Step	Action
1	Right-click on the required Object and select the <b>Advanced   Set Object State</b> context menu option. The Set Instance State dialog displays.
2	In the <b>State</b> field, either type the required State (such as <b>Awaiting Approval</b> ) or select a State from the drop-down list.

Step	Action
	<p>The drop-down list for the State field is populated with:</p> <ul style="list-style-type: none"> <li>Any States owned by the object's classifier</li> <li>Any States owned by any superclasses of the object's classifier</li> <li>Any States owned by State Machines owned by the object's classifier</li> <li>Any States owned by State Machines owned by any superclasses of the object's classifier</li> </ul>
3	<p>Click on the <b>OK</b> button to apply the State.</p> <p>The object now shows the run-time state in square brackets below the object name.</p> 

### 6.4.2.17 Package



#### Description

A *Package* is a namespace as well as an element that can be contained in other Package's namespaces. A Package can own or merge with other Packages, and its elements can be imported into a Package's namespace. In addition to using Packages in the Project Browser to organize your project contents, you can drag the Packages onto a diagram workspace (most diagram types, both standard and extended) for structural or relational depictions, including Package imports or merges.

#### Toolbox icon



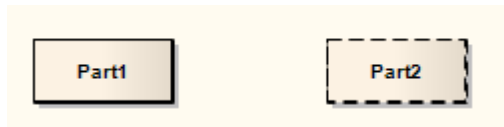
#### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 109) states:

*A package is a namespace for its members, and may contain other packages. Only packageable elements can be owned members of a package. By virtue of being a namespace, a package can import either*

individual members of other packages, or all the members of other packages. In addition a package can be merged with other packages.

### 6.4.2.18 Part



#### Description

*Parts* are run-time instances of Classes or Interfaces. Multiplicity can be specified for a Part, using the notation:

( x{ . . . y } )

where x specifies the initial or set number of instances when the composite structure is created, and y indicates the maximum number of instances at any time.

Parts are used to express composite structures, or modeling patterns that can be invoked by various objects to accomplish a specific purpose. When illustrating the composition of structures, Parts can be embedded as properties of other Parts. When embedded as properties, Parts can be bordered by a solid outline, indicating the surrounding Part owns the Part by composition. Alternatively, a dashed outline indicates that the property is referenced and used by the surrounding Part, but is not composed within it.

#### Toolbox icon



#### Learn more

- [Composite Structure Diagram](#) <sup>[1188]</sup>
- [Class](#) <sup>[1363]</sup>
- [Interface](#) <sup>[1377]</sup>
- [Properties](#) <sup>[1189]</sup>
- [Add Property Value](#) <sup>[1383]</sup>

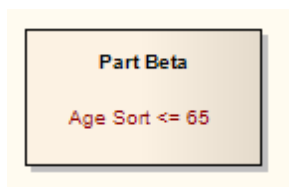
#### 6.4.2.18.1 Add Property Value

##### Add property value variables to a Part

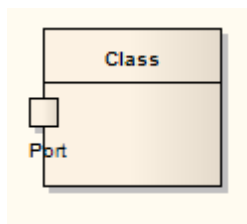
Step	Action
1	Right-click on the Part and select the <b>Advanced   Set Property Values</b> context menu option (or press <b>( Ctrl+Shift+R )</b> ). The Set Property Values dialog displays.
2	In the <b>Variable</b> field, click on the drop-down arrow and select the variable, or type in the new variable

Step	Action
	name.
3	Set the <b>Operator</b> , the <b>Value</b> and optionally type in a <b>Note</b> .
4	Click on the <b>OK</b> button to save the variable.

A Part with a property value resembles the following figure:



#### 6.4.2.19 Port



##### Description

*Ports* define the interaction between a classifier and its environment. *Interfaces* controlling this interaction can be depicted using the Interface element. Any connector to a Port must provide the required interface, if defined. Ports can appear on a contained Part, a Class, or the boundary of a Composite element.

A Port is a *typed* structural feature or property of its containing classifier. Ports are typically created in Class diagrams, Object diagrams and Composite Structure diagrams.

##### Toolbox icon



##### Learn more

- [Add a Port to an Element](#) <sup>1385</sup>
- [Inherited and Redefined Ports](#) <sup>1385</sup>
- [The Property Tab](#) <sup>1386</sup>



- [Composite Structure Diagram](#) <sup>[1188]</sup>

#### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 182*) states:

*A port is a property of a classifier that specifies a distinct interaction point between that classifier and its environment or between the (behavior of the) classifier and its internal parts. Ports are connected to properties of the classifier by connectors through which requests can be made to invoke the behavioral features of a classifier. A Port may specify the services a classifier provides (offers) to its environment as well as the services that a classifier expects (requires) of its environment.*

#### 6.4.2.19.1 Add a Port to an Element

##### Add a new Port to an element

Use *one* of the following steps:

Step	Action
1	Click on the Port symbol in the Composite Elements page of the Toolbox, and drag the symbol to (or click on) the target host element.  This creates an untyped, simple Port on the boundary, near the cursor position.
2	On the context menu of a suitable Class, Part or Composite element in the <b>Project Browser</b> , select the <b>Add   Port</b> option.
3	Drag a suitable classifier from the <b>Project Browser</b> onto a Class or Part on a <b>diagram</b> .  A prompt displays to add a typed Port or Part at the cursor position.  The new Port is typed by the original dragged classifier.
4	Use the Structural Elements dialog (on a <b>diagram</b> , <b>right-click on element   Structural Elements</b> ) to add a new Port to the currently selected element.

#### Learn more

- [Composite Elements](#) <sup>[936]</sup>
- [Inherited and Redefined Ports](#) <sup>[1385]</sup>
- [Manage Structural Elements](#) <sup>[935]</sup>

#### 6.4.2.19.2 Inherited and Redefined Ports

A Port is a *redefinable* and *re-useable* property of a composite classifier such as a Component. A Component can inherit Ports from its parent; if a Component's parent owns Ports, when you open the Structural Elements dialog for the Component and select the **Show Owned/Inherited** checkbox the inherited Ports and their named owners are listed.

If you want to show an inherited Port on a Component, the Structural Elements dialog provides two options:

- Expose an inherited Port - tick the checkbox next to the Port's name, to create a read-only copy of the Port; this is convenient for modeling Port interactions in child elements where the Ports are defined in the parent elements
- Redefine an inherited Port - select the row for the Port and click on the **Redefine** button, to create an editable copy of the Port; this is useful where a child element places additional restrictions or behavior on the Port

The above is also true for Components that inherit Ports from realized Interfaces, and for Component instances that inherit Ports from their classifying Component.

#### Learn more

- [Manage Structural Elements](#) 

### 6.4.2.19.3 The Property Page

The element Properties dialog for Ports and Parts has a Property page in place of the Class element Details page.

This page defines the type, initial value, Qualifiers, multiplicity, and redefined and subsetted properties of the Port or Part.

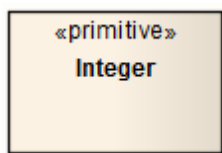
You set the Qualifiers by clicking on the **Qualifiers** button, to display the Qualifiers dialog.

You add **Redefined** and **Subsetted Properties** by clicking on the appropriate **Add** button, to display the Select Property dialog.

#### Learn more

- [Qualifiers Dialog](#) 
- [Select Property Dialog](#) 
- [Inherited and Redefined Ports](#) 

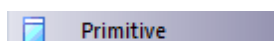
### 6.4.2.20 Primitive



#### Description

A Primitive element identifies a predefined data type, without any relevant substructure (that is, it has no parts in the context of UML). It could be regarded as a conceptual Data Type.

#### Toolbox icon



### Learn more

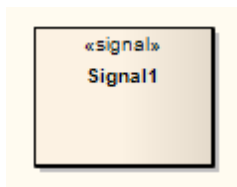
- [Data Type Element](#)<sup>[1370]</sup>

### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 124) states:

*A primitive data type may have an algebra and operations defined outside of UML, for example, mathematically ... The run-time instances of a primitive type are data values. The values are in many-to-one correspondence to mathematical elements defined outside of UML (for example, the various integers). Instances of primitive types do not have identity. If two instances have the same representation, then they are indistinguishable.*

## 6.4.2.21 Signal



### Description

A Signal is a specification of Send request instances communicated between objects, typically in a Class or Package diagram. The receiving object handles the Received request instances as specified by its receptions. The data carried by a Send request is represented as attributes of the Signal. A Signal is defined independently of the classifiers handling the signal occurrence.

To define a reception, create an operation in the receiving object and assign the stereotype <<signal>> to it. The reception has the same name as the signal that the object can receive.

### Toolbox icon



### Learn more

- [Class Diagram](#)<sup>[1184]</sup>
- [Package Diagram](#)<sup>[1182]</sup>
- [State Toolbox](#)<sup>[807]</sup>
- [Send Element](#)<sup>[1328]</sup>
- [Receive Element](#)<sup>[1327]</sup>

### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 450) states:

*A signal triggers a reaction in the receiver in an asynchronous way and without a reply. The sender of a signal will not block waiting for a reply but continue execution immediately. By declaring a reception associated to a given signal, a classifier specifies that its instances will be able to receive that signal, or a subtype thereof, and will respond to it with the designated behavior.*

*And (UML Superstructure Specification, v2.1.1, p. 447 - 448):*
















































*A reception is a declaration stating that a classifier is prepared to react to the receipt of a signal. A reception designates a signal and specifies the expected behavioral response. The details of handling a signal are specified by the behavior associated with the reception or the classifier itself. ...Receptions are shown using the same notation as for operations with the keyword <signal>*

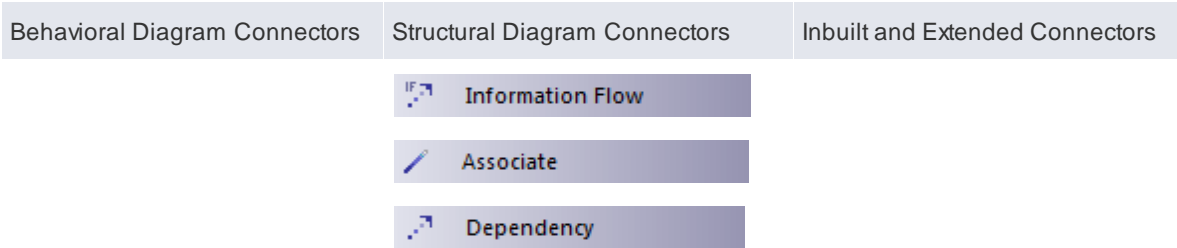
## 6.5 UML Connectors

A **connector** is a logical or functional relationship between model elements. There are several different connector types, each having a particular purpose and syntax. Enterprise Architect supports all of the UML connectors as well as various custom connectors. Together with the UML Elements, these form the basis of UML models.

For more information on using these connectors, display the appropriate topic by clicking on the appropriate connector icon in the table below.

Behavioral Diagram Connectors	Structural Diagram Connectors	Inbuilt and Extended Connectors
<b>Activity Diagrams</b>	<b>Composite Structure Diagrams</b>	<b>Analysis Diagrams</b>
 Control Flow	 Connector	 Information Flow
 Object Flow	 Assembly	 Object Flow
 Interrupt Flow	 Delegate	 Associate
	 Role Binding	 Realize
<b>Use Case Diagrams</b>	 Represents	 Representation
 Use	 Occurrence	
 Associate		<b>Common Connectors</b>
 Generalize	<b>Package and Class Diagrams</b>	 Dependency
 Include	 Associate	 Realize
 Extend	 Generalize	 Trace
 Realize	 Compose	 Information Flow
 Invokes	 Aggregate	 Note Link
 Precedes	 Association Class	
	 Assembly	<b>Profile Diagrams</b>
<b>State Diagrams</b>	 Realize	 Extension
 Transition	 Nesting	 Generalize
 Object Flow	 Package Merge	 Tagged Value
	 Package Import	
<b>Maintenance Diagrams</b>	 Abstraction	<b>Data Modeling Diagrams</b>

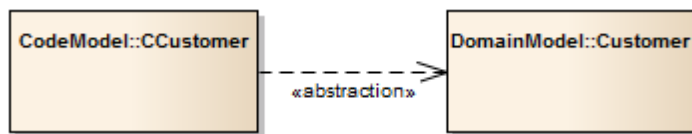
Behavioral Diagram Connectors	Structural Diagram Connectors	Inbuilt and Extended Connectors
 Aggregate	 Substitution	 Associate
	 Usage	
<b>Timing Diagrams</b>	<b>Component Diagrams</b>	<b>Metamodel Diagrams</b>
 Message	 Assembly	 Generalize
	 Delegate	 Associate
<b>Sequence Diagrams</b>	 Associate	 Compose
 Message	 Realize	 Aggregate
 Self-Message	 Generalize	
 Recursion		<b>Custom Diagrams</b>
 Call	<b>Deployment Diagrams</b>	 Associate
	 Associate	 Aggregate
<b>Communication Diagrams</b>	 Communication Path	 Generalize
 Associate	 Association Class	 Realize
 Realize	 Generalize	 Nesting
 Nesting	 Realize	
	 Deployment	<b>Requirements Diagrams</b>
<b>Interaction Overview Diagrams</b>	 Manifest	 Aggregate
 Control Flow	 Nesting	 Inheritance
 Object Flow		 Associate
 Interrupt Flow		 Implements
	<b>User Interface Diagrams</b>	
<b>XML Schema Diagrams</b>	 Associate	<b>Documentation Diagrams</b>
 Generalize	 Aggregate	No special connectors
 Associate	 Generalize	
	 Realize	<b>WSDL Diagrams</b>
		No special connectors
	<b>Object Diagrams</b>	



### Notes

- **Invokes** and **Precedes** relationships are defined by the Open Modeling Language (OML); they are stereotyped Dependency relationships - **Invokes** indicates that Use Case A, at some point, causes Use Case B to happen, whilst **Precedes** indicates that Use Case C must complete before Use Case D can begin
- An **Extension** relationship shows that a Stereotype extends one or more Metaclasses; all Stereotypes must extend either one or more Metaclasses, or another Stereotype that extends a Metaclass (or another Stereotype; in theory you can have a chain of Stereotypes ultimately extending a Metaclass, but in practice you are unlikely to have more than two Stereotypes in the chain)
- A **Tagged Value** relationship defines a reference-type (that is, RefGUID) Tagged Value owned by the source stereotype; the Tagged Value is named for the target role of this association, and is limited to referencing elements with the stereotype by the association target element

## 6.5.1 Abstraction

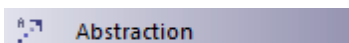


### Description

An Abstraction is a relationship between two elements that represent the same concept, either at different levels of abstraction or from different viewpoints. This diagram above shows that two different customer Classes from different models (the Domain model and the Code model) represent the same concept.

The Abstraction relationship is a subtype of a Dependency relationship.

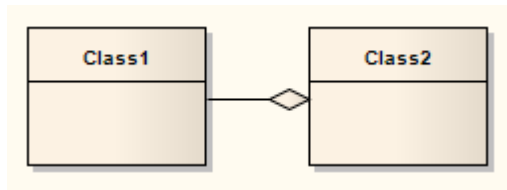
### Toolbox icon



### Learn more

- [Class Diagram](#) <sup>1184</sup>
- [Dependency](#) <sup>1404</sup>

## 6.5.2 Aggregation

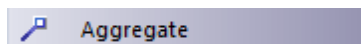


### Description

An Aggregation connector is a type of association that shows that an element contains or is composed of other elements. It is used in Class models, Package models and Object models to show how more complex elements (aggregates) are built from a collection of simpler elements (component parts; for example, a car from wheels, tires, motor and so on).

A stronger form of aggregation, known as Composite Aggregation, is used to indicate ownership of the whole over its parts. The part can belong to only one Composite Aggregation at a time. If the composite is deleted, all of its parts are deleted with it.

### Toolbox icon



### Learn more

- [Class Diagram](#) <sup>[1184]</sup>
- [Package Diagram](#) <sup>[1182]</sup>
- [Object Diagram](#) <sup>[1186]</sup>
- [Change Aggregation Connector Form](#) <sup>[1392]</sup>

### 6.5.2.1 Change Aggregation Connector Form

In your modeling, when you create an **Aggregation** relationship it defaults to the **weak** (shared) form of the relationship, represented by a hollow diamond head. You can change this to the strong form (Composition), represented by a solid black diamond head.

#### Change the form of an Aggregation connector from weak to strong

Step	Action
1	Right-click on an Aggregation connector to display the context menu.
2	Select <b>Set Aggregation to Composite</b> . The diamond is shown as filled.



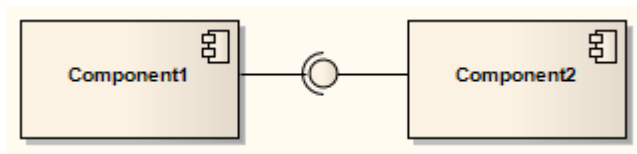
### Notes

- If the connector is already a Strong (Composition) connector, the context menu option changes to **Set Aggregation to Shared**

### Learn more

- [Aggregation](#) <sup>[1392]</sup>

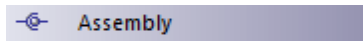
## 6.5.3 Assembly



### Description

An Assembly connector bridges a component's required interface (*Component1*) with the provided interface of another component (*Component2*), typically in a Component diagram.

### Toolbox icon



### Learn more

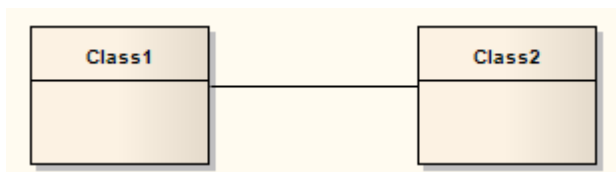
- [Component Diagram](#) <sup>[1194]</sup>

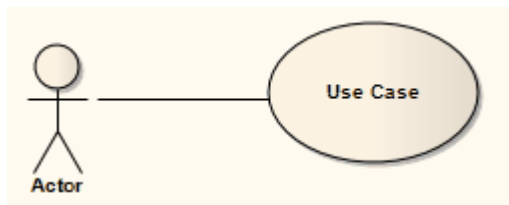
### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 156*) states:

*An assembly connector is a connector between two components that defines that one component provides the services that another component requires. An assembly connector is a connector that is defined from a required interface or port to a provided interface or port.*

## 6.5.4 Association





### Description

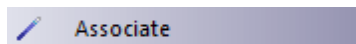
An **Association** implies that two model elements have a relationship, usually implemented as an instance variable in one or both Classes. The connector can include named roles at each end, multiplicity, direction and constraints. You can also indicate the reading direction by adding a Name Direction Indicator arrow to a **label** on the connector, and define **template binding parameters** for an Association connector between a binding Class and a parameterized Class.

Associations act as the keys in providing possible **classifiers** for a structure of instance elements, and for automatically generating **Property** (Part) elements on the source **SysML Block** element in the Association.

When code is generated for Class diagrams, Associations become member variables in the target Class. The relationship is also used in Package, Object, Communication, Data Modeling and Deployment diagrams.

'Association' is the general relationship type between two elements; to connect more than two elements in an Association, you can use the **N-Ary Association** element. An Association connector can also be integrated with a Class element to form an **Association Class**, to allow the connector to have operations and attributes that define certain types of UML relationship.

### Toolbox icon



### Learn more

- [N-Ary Association](#) <sup>[2007]</sup>
- [Association Class](#) <sup>[1398]</sup>
- [Class Diagrams](#) <sup>[1184]</sup>
- [Template Binding](#) <sup>[1444]</sup>
- [Qualifiers](#) <sup>[1395]</sup>
- [Manage Object Labels](#) <sup>[866]</sup>
- [Classify Object Via Classifier Associations](#) <sup>[1012]</sup>
- [Generate Properties From Block Associations](#) <sup>[2304]</sup>

### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 41) states:

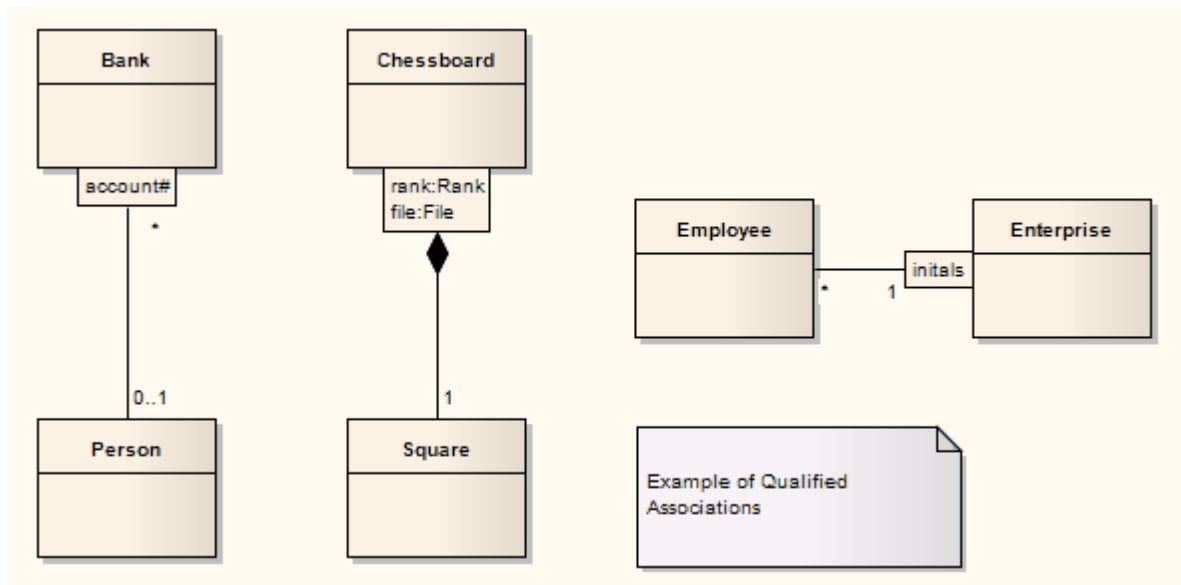
*An association specifies a semantic relationship that can occur between typed instances. It has at least two ends represented by properties, each of which is connected to the type of the end. More than one end of the association may have the same type.*

An end property of an association that is owned by an end class or that is a navigable owned end of the association indicates that the association is navigable from the opposite ends; otherwise, the association is not navigable from the opposite ends.

### 6.5.4.1 Qualifiers

Qualifiers are ordered sets of properties of an Association end point, a Part, a Port, or an Attribute, that limit the nature of the relationship between two classifiers or objects. You define a qualifier on the Qualifiers dialog, which you display by clicking on the ( ... ) button at the end of the **Qualifiers** field on the Association, Part, Port or Attribute Properties dialog.

#### Examples



#### Notes

- When typing multiple Qualifiers into the **Qualifier(s)** field on a Properties dialog, separate them with a semi-colon; each Qualifier then displays on a separate line - for example, in the diagram the Qualifier 'rank:Rank;file:File' has been rendered in two lines, with a line break at the ; character
- You can enable or disable Qualifier rectangles in the Diagram page of the Options dialog (select the **Tools | Options | Diagram** menu option) - if disabled, the old style text Qualifiers are used; it is not recommended that you disable Qualifiers as they are an integral part of the UML
- You can enable or disable a mild shading on the Qualifier rectangles in the Links page of the Options dialog

#### Learn more

- [Qualifiers Dialog](#) <sup>[1396]</sup>
- [Source Role](#) <sup>[1130]</sup>
- [Port](#) <sup>[1384]</sup>
- [Part](#) <sup>[1383]</sup>
- [Attribute](#) <sup>[999]</sup>

**OMG UML Specification:**

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 129) states:

*A qualifier declares a partition of the set of associated instances with respect to an instance at the qualified end (the qualified instance is at the end to which the qualifier is attached). A qualifier instance comprises one value for each qualifier attribute. Given a qualified object and a qualifier instance, the number of objects at the other end of the association is constrained by the declared multiplicity. In the common case in which the multiplicity is 0..1, the qualifier value is unique with respect to the qualified object, and designates at most one associated object. In the general case of multiplicity 0..\*, the set of associated instances is partitioned into subsets, each selected by a given qualifier instance. In the case of multiplicity 1 or 0..1, the qualifier has both semantic and implementation consequences. In the case of multiplicity 0..\*, it has no real semantic consequences but suggests an implementation that facilitates easy access of sets of associated instances linked by a given qualifier value.*


**6.5.4.1.1 Qualifiers Dialog**

The Qualifiers dialog is used to define the Qualifiers of an Association connector end, Port, Part or Attribute.

**General Tab**

Review, edit or complete the fields as indicated in the following table:

Field	Action	See also
<b>Name</b>	Display the name of the Qualifier. For a new Qualifier, type the name (with no spaces).	
<b>Alias</b>	Display an optional alias for the Qualifier. If necessary, type in a new alias.	
<b>Type</b>	Display the Qualifier type.  The type can be defined by the code language (data type) or by a classifier element. When you click on the drop-down arrow, the set of values in the list provides the appropriate data types.  To select or define possible classifiers, either click on the <b>Select Type</b> option in the list, or click on the ( ... ) (Select) button to display the Select <Item> dialog.  To add new code language data types that can be displayed in this list, see the <i>Data Types</i> topic.	<a href="#">Select &lt;Item&gt; Dialog</a> <a href="#">Data Types</a>
<b>Scope</b>	Define the Qualifier as <b>Public</b> , <b>Protected</b> , <b>Private</b> or <b>Package</b> . If necessary, click on the drop-down arrow and select a different scope.	
<b>Stereotype</b>	Define the optional stereotype of the Qualifier. If necessary, either type a different stereotype name or click on	

Field	Action	See also
	the drop-down arrow and select a stereotype.	
<b>Derived</b>	Indicate that the Qualifier is a calculated value. If you select this checkbox, the Qualifier name on the element has the derived symbol (/) as a prefix.	
<b>Static</b>	Indicate that the Qualifier is a static member.	
<b>Const</b>	Indicate that the Qualifier is a constant.	
<b>Initial</b>	Display an optional initial value. If necessary, type in a new initial value.	
<b>Notes</b>	Enter any free text notes associated with the Qualifier. You can format the notes text using the Notes toolbar at the top of the field.	<a href="#">Notes</a>  toolbar

To change the position of a Qualifier in the list in the Qualifiers panel, click on the **Scroll Up** or **Scroll Down** (hand) buttons.

### Detail Tab

Use the Detail tab to model additional properties of a selected Qualifier, such as its multiplicity, redefined properties and subsetting properties.

Select a Qualifier on the General tab, then review, edit or complete the Detail tab fields as indicated in the following table.

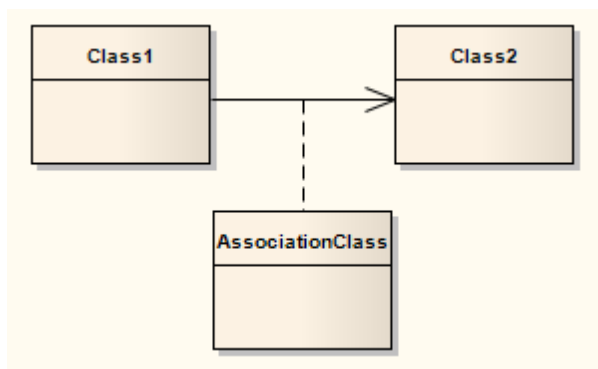
Field	Action	See also
<b>Multiplicity</b>		
<b>Lower bound</b>	Define a lower limit to the number of elements allowed in the collection.	
<b>Upper bound</b>	Define an upper limit to the number of elements allowed in the collection.	
<b>Allow Duplicates</b>	Indicate that duplicates are allowed.	

Field	Action	See also
	Maps to the UML property <i>isUnique</i> , value <i>FALSE</i> .	
<b>Multiplicity is Ordered</b>	Indicate that the collection is ordered.	
<b>Redefined Property</b>	Review the redefined properties for the Qualifier. Add redefined properties by clicking on the <b>Add</b> button to display the Select Property dialog.	<a href="#">Select Property Dialog</a> [996]
<b>Subsetted Property</b>	Review the subsetted properties for the qualifier. Add subsetted properties by clicking on the <b>Add</b> button to display the Select Property dialog.	<a href="#">Select Property Dialog</a> [996]

#### Learn more

- [Qualifiers](#) [1395]
- [Source Role](#) [1130]
- [Port](#) [1384]
- [Part](#) [1383]
- [Attribute](#) [999]

### 6.5.5 Association Class



#### Description

An *Association Class* is a UML construct that enables an Association to have *attributes* and *operations* (*features*). This results in a hybrid relation with the characteristics of an Association and a Class.

When you add an Association Class connection, Enterprise Architect also creates a Class that is automatically connected to the Association. When you hide or delete the Association, the Class is also hidden or deleted.

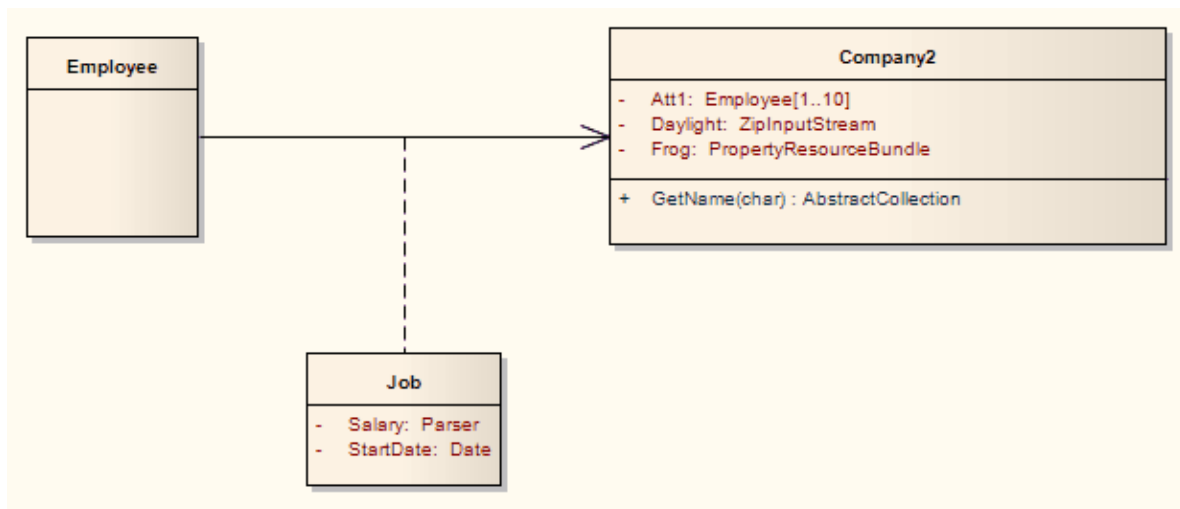
To add an Association Class to a Class or Deployment diagram, click on the *Association Class* icon in the Toolbox. Click and hold on the source object in the diagram while you drag the line to the target element,

then release the mouse button. Enterprise Architect draws the connector and adds the Class, then prompts you to add the Class name. Note that the names of the Class and the connector are the same. You can also connect a new Class to an existing Association.

You can highlight the Class part of an Association Class in the Project Browser, by selecting the **Find Association Class** context menu option on the Association connector.

### Example

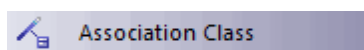
The following diagram illustrates an Association Class between model elements. Note the dotted line from the Class to the Association. You cannot move or delete this line.



### Notes

- If you are applying a stereotype with a Shape Script to an Association Class, be aware that the Shape Script is applied to both the Class part and the Association part; therefore, you might have to include logic in the *shape main* that tests the type of the element so that you can give separate drawing instructions for Class and for Association
- Such logic is not necessary in the:
  - Shape source or shape target, which are ignored by Classes, or the
  - Decoration shapes, which are ignored by Association connectors
- If you dissociate the Class from the Association connector, both parts keep their Shape Scripts until the stereotypes are removed.

### Toolbox icon



### Learn more

- [Class Diagram](#) <sup>[1184]</sup>
- [Connect New Class to Association](#) <sup>[1400]</sup>

**OMG UML Specification:**

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 49) states:

*A model element that has both association and class properties. An AssociationClass can be seen as an association that also has class properties, or as a class that also has association properties. It not only connects a set of classifiers but also defines a set of features that belong to the relationship itself and not to any of the classifiers.*

### 6.5.5.1 Connect New Class to Association

**How to**

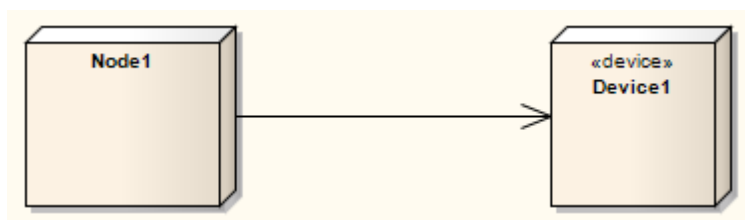
To connect a new Class to an existing Association

Step	Action
1	Create a Class in the diagram containing the Association to connect.
2	Right-click on the new Class and select the <b>Advanced   Association Class</b> menu option. The Create Association Class dialog displays.
3	Select the connector to connect to.
4	Click on the <b>OK</b> button.

**Learn more**

- [Association Class](#) [1398]

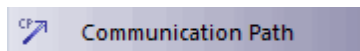
### 6.5.6 Communication Path

**Description**

A *Communication Path* defines the path through which two *DeploymentTargets* are able to exchange signals and messages. Communication Path is a specialization of *Association*. A *DeploymentTarget* is the target for a deployed *Artifact* and can be a *Node*, *Property* or *InstanceSpecification* in a Deployment diagram.



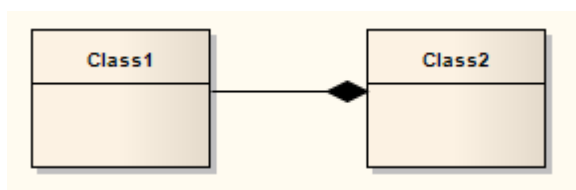
### Toolbox icon



### Learn more

- [Deployment Diagram](#) <sup>[1197]</sup>
- [Association](#) <sup>[1393]</sup>

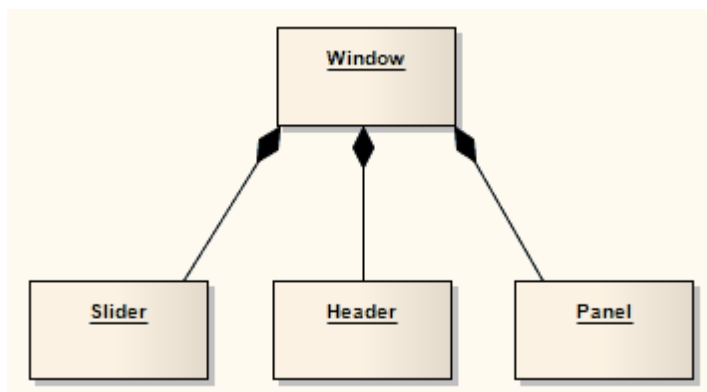
## 6.5.7 Composition



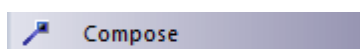
### Direction:

A Composition is used to depict an element that is made up of smaller components, typically in a Class or Package diagram. A component - or part instance - can be included in a maximum of one composition at a time. If a composition is deleted, usually all of its parts are deleted with it; however, a part can be individually removed from a composition without having to delete the entire composition. Compositions are transitive, asymmetric relationships and can be recursive.

### Example



### Toolbox icon



### Learn more

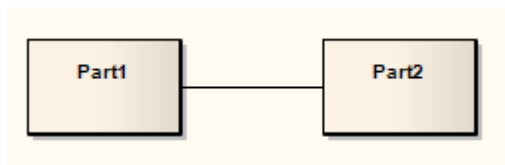
- [Aggregation](#) <sup>[1392]</sup>
- [Class Diagram](#) <sup>[1184]</sup>

**OMG UML Specification:**

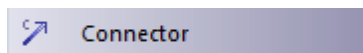
The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 43) states:

*Composite aggregation is a strong form of aggregation that requires a part instance be included in at most one composite at a time. If a composite is deleted, all of its parts are normally deleted with it.*

### 6.5.8 Connector

**Description**

Connectors illustrate communication links between Parts to fulfill the structure's purpose, typically in a Composite Structure diagram. Each Connector end is distinct, controlling the communication pertaining to its connecting element. These elements can define constraints specifying this behavior. Connectors can have multiplicity.

**Toolbox icon****Learn more**

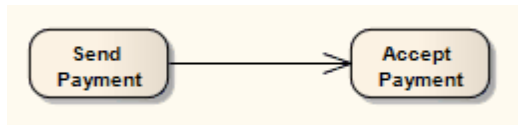
- [Composite Structure Diagram](#) <sup>[1188]</sup>

**OMG UML Specification:**

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 177) states:

*Specifies a link that enables communication between two or more instances. This link may be an instance of an association, or it may represent the possibility of the instances being able to communicate because their identities are known by virtue of being passed in as parameters, held in variables or slots, or because the communicating instances are the same instance. The link may be realized by something as simple as a pointer or by something as complex as a network connection. In contrast to associations, which specify links between any instance of the associated classifiers, connectors specify links between instances playing the connected parts only.*

### 6.5.9 Control Flow



#### Description

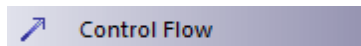
The *Control Flow* is a connector connecting two nodes in an Activity diagram. Control Flow connectors bridge the flow between Activity nodes, by directing the flow to the target node once the source node's activity is completed.

Control Flows and Object Flows can define a *Guard* and a *Weight* condition.

A Guard defines a condition that must be true before control passes along that activity edge. A practical example of this is where two or more activity edges (Control Flows) exit from a Decision element. Each flow should have a Guard condition that is exclusive of the other and defines which edge is taken under what conditions. The Control Flow Properties dialog enables you to set up Guard conditions on Control Flows and on Object Flows.

A Weight defines the number of tokens that can flow along a Control or Object Flow connection when that edge is traversed. Weight can also be defined on the Control Flow and Object Flow Properties dialogs.

#### Toolbox icon



#### Learn more

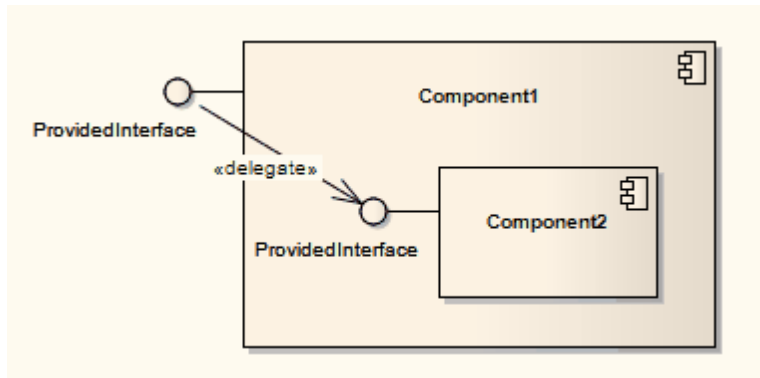
- [Activity diagram](#) <sup>[1198]</sup>
- [Object Flow](#) <sup>[1435]</sup>
- [Decision Node](#) <sup>[1294]</sup>

#### OMG UML specification:

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 356) states:

*A control flow is an edge that starts an activity node after the previous one is finished.*

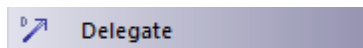
### 6.5.10 Delegate



#### Description

A Delegate connector defines the internal assembly of a component's external Ports and Interfaces, on a Component diagram. Using a Delegate connector wires the internal workings of the system to the outside world, by a delegation of the external interfaces' connections.

#### Toolbox icon



#### Learn more

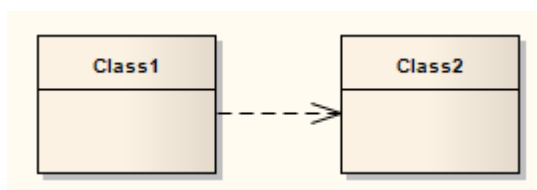
- [Component Diagram](#) 1194

#### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 156) states:

*A delegation connector is a connector that links the external contract of a component (as specified by its ports) to the internal realization of that behavior by the component's parts. It represents the forwarding of signals (operation requests and events): a signal that arrives at a port that has a delegation connector to a part or to another port will be passed on to that target for handling.*

### 6.5.11 Dependency

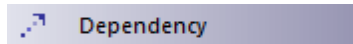


#### Description

Dependency relationships are used to model a wide range of dependent relationships between model

elements in Use Case, Activity and Structural diagrams, and even between models themselves. You can create the Dependency from the Common page of the Toolbox. The Dependencies package as defined in UML 2.4.1 has many derivatives, such as Realize, Deployment and Use. Once you create a Dependency you can further refine its meaning by applying a specialized stereotype.

### Toolbox icon



### Learn more

- [Realization](#) <sup>[1440]</sup>
- [Deployment](#) <sup>[1408]</sup>
- [Use](#) <sup>[1450]</sup>
- [Apply a Stereotype](#) <sup>[1405]</sup>

### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 64) states:

*A dependency is a relationship that signifies that a single or a set of model elements requires other model elements for their specification or implementation. This means that the complete semantics of the depending elements is either semantically or structurally dependent on the definition of the supplier element(s).*

## 6.5.11.1 Apply a Stereotype

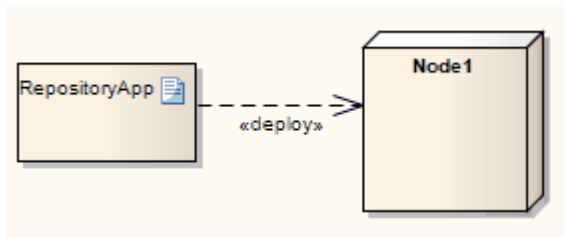
### How to

To apply a stereotype to a Dependency relationship

Step	Action
1	Select the Dependency relationship to change.
2	Right-click on the connector and, from the context menu, select the <b>Dependency Properties</b> option. The Dependency Properties dialog displays.
3	In the <b>Stereotype</b> field, either type in the required stereotype name or click on the drop-down arrow and select the stereotype from the list.
4	Click on the <b>OK</b> button.

Alternatively, you can right-click on the Dependency relationship and select the **Advanced | Dependency Stereotypes** context menu option, then select from a shorter list of standard stereotypes.

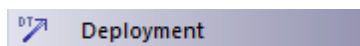
### 6.5.12 Deployment



#### Description

A Deployment is a type of Dependency relationship that indicates the deployment of an artifact onto a node or executable target, typically in a Deployment diagram. A Deployment can be made at type and instance levels. At the type level, a Deployment would be made for every instance of the node. Deployment can also be specified for an instance of a node, so that a node's instances can have varied deployed artifacts. With composite structures modeled with nodes defined as Parts, Parts can also serve as targets of a Deployment relationship.

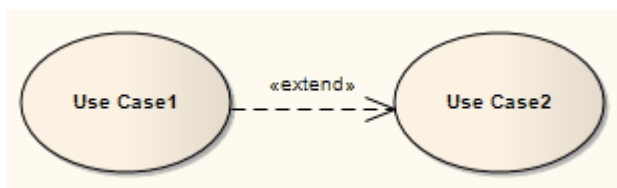
#### Toolbox icon



#### Learn more

- [Deployment Diagram](#) 1197

### 6.5.13 Extend

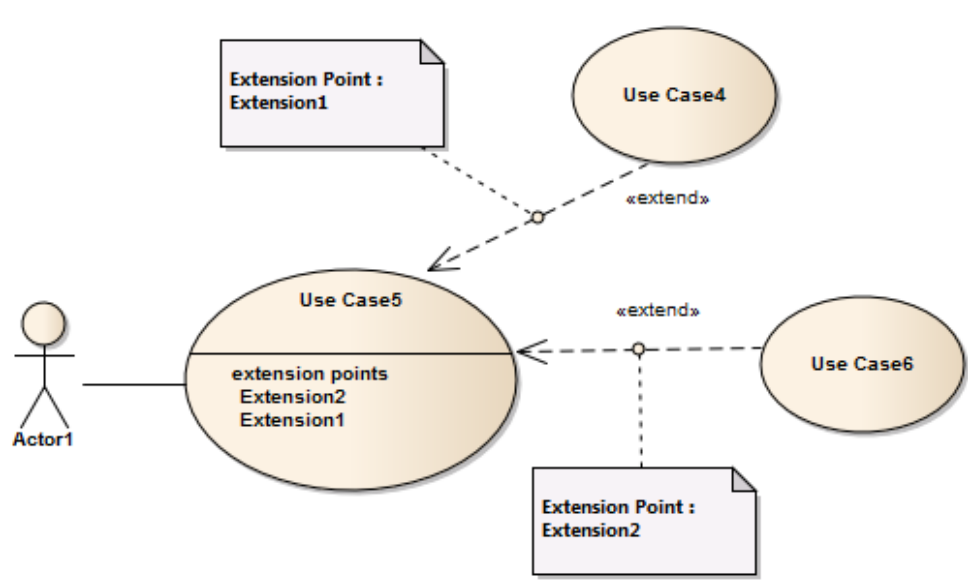


#### Description

An **Extend** connector is used to indicate that an element extends the behavior of another, mainly in **Use Case** models where one Use Case (optionally) extends the behavior of another Use Case. An extending Use Case often expresses alternative flows that are integrated with the behavior of the extended Use Case, at a specific point in the behavior flow identified within the element by an **extension point**. The extension point is represented by a text string such as 'on startup' or 'before connection is established'.

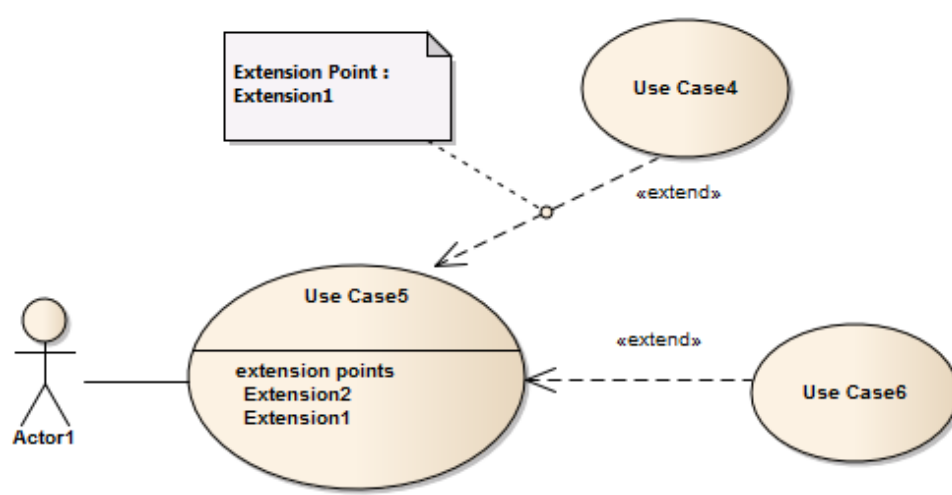
A Use Case can have more than one extension point, and can extend or be extended by more than one other Use Case. The precise relationship between the extending Use case, extended Use Case and the point at which the extension applies can be identified on the Extend relationship, as defined below.

#### Identify Extension Point

Step	Action	See also
1	<p>Right-click on the Extend connector and select the <b>Advanced   Extension Point   Set Extension Point</b> context menu option.</p> <p>The Element Usage dialog displays, listing the Extension Points currently defined in the target Use Case element.</p>	
2	<p>Click on the Extension Point on which the source Use Case acts, and click on the <b>Open</b> button.</p> <p>The dialog closes and the Extend connector shows a small circle at the mid-point, with a Notelink to a Note element that identifies the selected Extension Point.</p>  <p>(The Note might not initially display close to the Extend connector - check the upper left corner of the diagram and drag the Note to the position you want it to occupy.)</p> <p>Use these same steps to <b>change</b> the extension point identified in the Note.</p>	

#### Show/Hide Extension Point Note

Step	Action	See also
1	<p>Right-click on the Extend connector and select the <b>Advanced   Extension Point   Show Extension Point</b> context menu option.</p> <p>If there are any Extension Points identified on the selected Extend connector, they are displayed as shown above.</p>	

Step	Action	See also
2	<p>Right-click on the Extend connector and <b>deselect</b> the <b>Advanced   Extension Point   Show Extension Point</b> context menu option.</p> <p>Any Extension Points identified on the selected Extend connector are hidden, as shown:</p> 	

#### Toolbox icon



#### Notes

- The **Extend** connector is not the same as the **Extension** connector used in **Profile** diagrams to indicate that a Stereotype element extends a Metaclass or another Stereotype element; the two types of connector have different appearances

#### Learn more

- [Use Case Diagram](#) <sup>[1201]</sup>
- [Use Case](#) <sup>[1352]</sup>
- [Use Case Extension Points](#) <sup>[1354]</sup>
- [Profile Toolbox](#) <sup>[810]</sup> (Extension connector)

#### OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 587*) states:

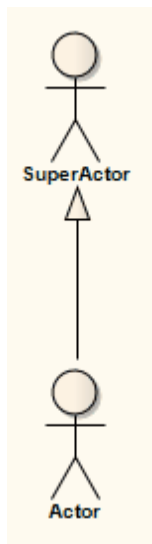
*This relationship specifies that the behavior of a Use Case may be extended by the behavior of another (usually supplementary) Use Case. The extension takes place at one or more specific extension points defined in the extended Use Case. Note, however, that the extended Use Case is defined independently of the extending Use Case and is meaningful independently of the extending Use Case. On the other hand, the*



*extending Use Case typically defines behavior that may not necessarily be meaningful by itself. Instead, the extending Use Case defines a set of modular behavior increments that augment an execution of the extended Use Case under specific conditions.*

*Note that the same extending Use Case can extend more than one Use Case. Furthermore, an extending Use Case may itself be extended.*

### 6.5.14 Generalization

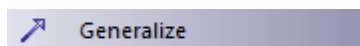


#### Description

A Generalization is used to indicate inheritance. Drawn from the specific classifier to a general classifier, the generalization's implication is that the source inherits the target's characteristics. It is used typically in Class, Component, Object, Package, Use Case and Requirements diagrams.

You can also define template binding parameters for a Generalize connector between a binding Class and a parameterized Class.

#### Toolbox icon

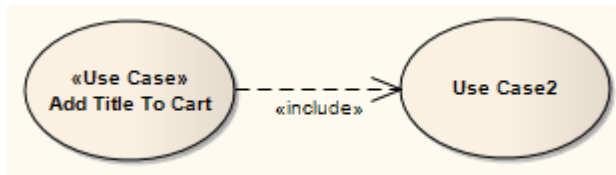


#### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 73) states:

*A generalization is a taxonomic relationship between a more general classifier and a more specific classifier. Each instance of the specific classifier is also an indirect instance of the general classifier. Thus, the specific classifier inherits the features of the more general classifier.*

### 6.5.15 Include



#### Description

An Include connection indicates that the source element includes the functionality of the target element. Include connections are used in Use Case models to reflect that one Use Case includes the behavior of another. Use an Include relationship to avoid having the same subset of behavior in many Use Cases; this is similar to delegation used in Class models.

#### Toolbox icon



#### Learn more

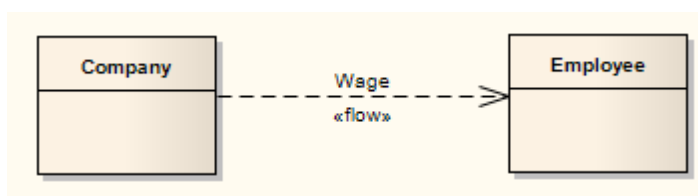
- [Use Case Diagram](#) f1201

#### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 591) states:

*Include is a DirectedRelationship between two Use Cases, implying that the behavior of the included Use Case is inserted into the behavior of the including Use Case. It is also a kind of NamedElement so that it can have a name in the context of its owning Use Case. The including Use Case may only depend on the result (value) of the included Use Case. This value is obtained as a result of the execution of the included Use Case.*

### 6.5.16 Information Flow



#### Description

An Information Flow represents the flow of Information Items (either Information Item elements or classifiers) between two elements in any diagram. The connector is available from:

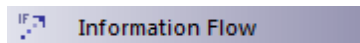
- The Common page of the Toolbox
- Every Quick Link menu, and

- Automatically whilst directly defining Information Item realization

When you create the Information Flow connector, Enterprise Architect automatically prompts you to identify which information items are conveyed.

You can have more than one Information Flow connector between the same two elements, identifying which items flow between the elements under differing conditions. The connectors can flow in the same direction, or opposite directions. You can locate the items conveyed in any Information Flow, by right-clicking on the connector and selecting the **Find Items Conveyed** context menu option.

#### Toolbox icon



#### Learn more

- [Information Item](#) <sup>[1376]</sup>
- [Using Information Flows](#) <sup>[1411]</sup>
- [Convey Information on a Flow](#) <sup>[1413]</sup>
- [Realize an Information Flow](#) <sup>[1414]</sup>

#### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 606) states:

*An InformationFlow specifies that one or more information items circulates from its sources to its targets.*

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 607) also states:

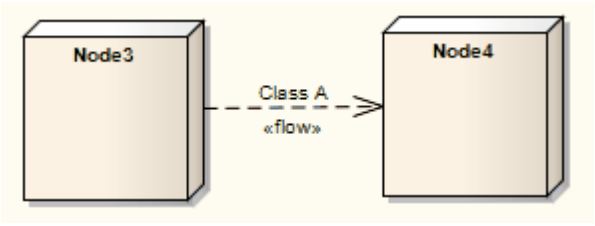
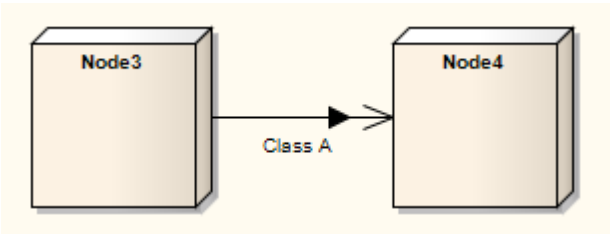
*An information flow is an abstraction of the communication of an information item from its sources to its targets. It is used to abstract the communication of information between entities of a system. Sources or targets of an information flow designate sets of objects that can send or receive the conveyed information item.*

### **6.5.16.1 Using Information Flows**

When you drag an Information Flow connector between two elements on a diagram, Enterprise Architect automatically prompts you to identify the Information items conveyed. You can also immediately go on to realize the Information Flows. However, you can also defer the identification of Information Items at either stage.

You can also create an Information Flow automatically whilst directly defining Information Flow realization, as you might do on a Message on a Sequence diagram.

#### Create and realize Information Flows

Step	Action	See also
1	Open a diagram and add two elements (for example, Nodes on a Deployment diagram).	
2	Click on the <i>Information Flow</i> connector in the Common page of the Toolbox and drag the cursor between the two elements.  The Information Items Conveyed dialog displays.  (If you prefer, click on the <b>Cancel</b> button to create the connector without identifying any information items, to return to the process at a later time.)	<a href="#">Convey Information on a Flow</a> [1413]
3	Add the classifier or Information Item element(s) to the Information Flow.  The diagram now resembles the following:    (If you prefer, click on the <b>OK</b> button to leave the information flow without realizing the flow, to return to the process at a later time.)	<a href="#">Convey Information on a Flow</a> [1413]
4	Add another connector between the same two elements (for example, a <i>Communication Path</i> connector).	
5	Right-click the connector and select the <b>Advanced   Information Flows Realized</b> context menu option.  The Information Flows Realized dialog displays.	
6	Tick the checkbox against each required information item in the realized flow and click on the <b>OK</b> button.  The combined connector now resembles the following:  	<a href="#">Realize an Information Flow</a> [1414]

**Notes**

- Once the Information Flow is realized, you cannot access the Information Items Conveyed dialog directly; you add or delete information items on the connector using the Information Items Realized dialog
- If you have more than one Information Flow connector between the elements, they form part of the same combined connector; you can again work on them separately through the Information Items Realized dialog
- If you have information flows in a diagram that you use as the source for a Pattern, the Information Items Conveyed and Information Flows Realized data is **not** copied into the Pattern
- You can locate, in the Project Browser, the classifier or information item element(s) conveyed on the Information Flow connector, using the **Find Items Conveyed** context menu option on the connector

**Learn more**

- [Message \(Sequence Diagram\)](#)<sup>[1418]</sup>


**6.5.16.2 Convey Information on a Flow**

When you create an Information Flow connector between two elements, Enterprise Architect automatically prompts you to specify which Information Items or classifiers are conveyed on this flow. You can ignore the prompt and add the information items later, or - if you have not realized the information flow with its *existing* information items - you can change and/or add to the information items conveyed.




**Access**    **Information Flow context menu | Advanced | Information Items Conveyed**

**Specify the Information Items conveyed on an Information Flow**

Step	Action	See also
1	On the Information Items Conveyed dialog, click on the <b>Add</b> button. The Select Classifier dialog displays.	
2	Browse or search for the required Information Item or classifier element or elements and select them as required.  If you do not want to retain a selected item on the Select Classifier dialog, press <b>(Ctrl)</b> and click on the item.	<a href="#">Select &lt;Item&gt; Dialog</a> <sup>[994]</sup>
3	Click on the <b>OK</b> button to return to the Information Items Conveyed dialog. Each information item you have selected is listed on a separate line in the dialog.	
4	If you do not want to retain a selected item on this dialog, click on it and click on the <b>Remove</b> button.  Click on the <b>OK</b> button to close the dialog and to show the selected information item element names on the Information Flow connector label.  If appropriate, you can now realize the information items conveyed in this Information	<a href="#">Realize an Information</a>

Step	Action	See also
	Flow.	<a href="#">Flow</a> 

#### Learn more

- [Information Flow](#) 
- [Information Item](#) 
- [Using Information Flows](#) 

### 6.5.16.3 Realize an Information Flow

After you create an information Flow connector you might want to:

- Realize one or more existing flows on the Information Flow connector
- Edit an existing flow on the Information Flow connector

You might also want to create and realize information flows on a non Information Flow connector, such as a Message on a Sequence diagram. You can perform these actions using the Information Flows Realized dialog, which displays all existing flows that can be realized on the selected connector.

**Access** [Connector context menu](#) | [Advanced](#) | [Information Flows Realized](#)

#### Review item flows on an Information Flow Connector

Operation	Action
<b>Realize Information Flows on the selected connector</b>	Select the checkbox for each required flow and click on the <b>OK</b> button.
<b>Cancel realization of a flow</b>	Deselect the checkbox against the appropriate flow, and click on the <b>OK</b> button.
<b>Change the classifier or Information Item elements conveyed on an Information Flow</b>	<p>Click on the flow text, then click on the [ ... ] button.</p> <p>The Select Classifier dialog displays:</p> <ul style="list-style-type: none"> <li>• Click on a single item to select it</li> <li>• <b>[Ctrl]+click</b> on each of several items to select them all, or</li> <li>• <b>[Ctrl]+click</b> on a selected item to deselect it</li> </ul> <p>Click on the <b>OK</b> button to return to the Information Flows Realized dialog and, if required, realize the changed flow as above.</p>
<b>Create a realized</b>	1. Right-click on the connector and select the <b>Information Flows Realized</b> context

Operation	Action
<b>information flow connector on another connector</b>	<p>menu option.</p> <ol style="list-style-type: none"> <li>Click on the <i>Click to create new information flow...</i> text. The Select Classifier dialog displays.</li> <li>Select the required classifier or Information Item elements, and click on the <b>OK</b> button to return to the Information Flows Realized dialog; the selected elements are listed first on the dialog, with the activation checkbox ticked.</li> <li>Click on the <b>OK</b> button to return to the diagram; the connector now displays as a realized Information Flow, with the selected classifier or Information Item elements named in the connector label.</li> </ol>

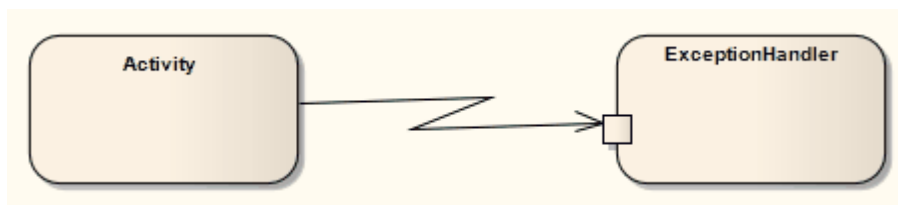
### Notes

- If there are several Information flows and you do not realize all of them, those that are not realized are represented by a separate Information Items Conveyed iteration of the Information Flow connector; you can only realize those flows on the *original* connector, at which point the flow is represented on that original connector
- If you realize all of the flows, they are combined on the one connector line
- If you realize an information flow on a connector, you can use the **Find Items Conveyed** context menu option to locate the corresponding Information Flow item in the Project Browser

### Learn more

- [Information Flow](#) <sup>[1410]</sup>
- [Message \(Sequence Diagram\)](#) <sup>[1418]</sup>

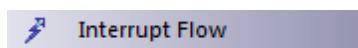
## 6.5.17 Interrupt Flow



### Description

The *Interrupt Flow* is a connection used to define the two UML concepts of connectors for Exception Handler and Interruptible Activity Region. An Interrupt Flow is a type of activity edge. It is typically used in an Activity diagram.

### Toolbox icon



### Learn more

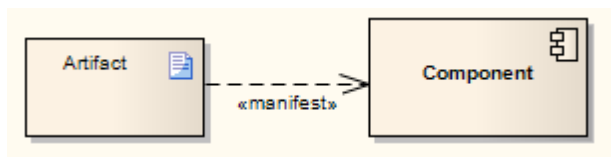
- [Exception Handler](#) <sup>[1300]</sup>
- [Interruptible Activity Region](#) <sup>[1318]</sup>
- [Activity diagram](#) <sup>[1199]</sup>

#### **OMG UML Specification:**

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 327) states:

*An activity edge is an abstract class for directed connections between two activity nodes.*

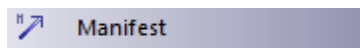
### **6.5.18 Manifest**



#### **Description**

A Manifest relationship indicates that the Artifact source embodies the target model element, typically in Component and Deployment diagrams. Stereotypes can be added to Enterprise Architect to classify the type of manifestation of the model element.

#### **Toolbox icon**



#### **Learn more**

- [Artifact Element](#) <sup>[1358]</sup>
- [Component Diagram](#) <sup>[1194]</sup>
- [Deployment Diagram](#) <sup>[1197]</sup>

#### **OMG UML Specification:**

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 212) states:

*An artifact embodies or manifests a number of model elements. The artifact owns the manifestations, each representing the utilization of a packageable element.*

*Specific profiles are expected to stereotype the manifestation relationship to indicate particular forms of manifestation, e.g. «tool generated» and «custom code» might be two manifestations for different classes embodied in an artifact.*

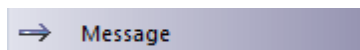


### 6.5.19 Message

Messages indicate a flow of information or transition of control between elements. Messages can be used in Timing Diagrams, Sequence Diagrams and Communication Diagrams (but not Interaction Overview diagrams) to reflect system behavior. If between Classes or classifier instances, the associated list of operations is available to specify the event.

Moving a Message can disrupt the organization of other features on the diagram. To avoid this, and move *only* the Message, press **Alt** while you move the Message.

#### Toolbox icon



#### Learn more

- [Message \(Sequence Diagram\)](#) <sup>[1418]</sup>
- [Message \(Communication Diagram\)](#) <sup>[1428]</sup>
- [Message \(Timing Diagram\)](#) <sup>[1431]</sup>
- [Timing Diagram](#) <sup>[1431]</sup>
- [Sequence Diagram](#) <sup>[1418]</sup>
- [Communication Diagram](#) <sup>[1428]</sup>

#### OMG UML Specification:

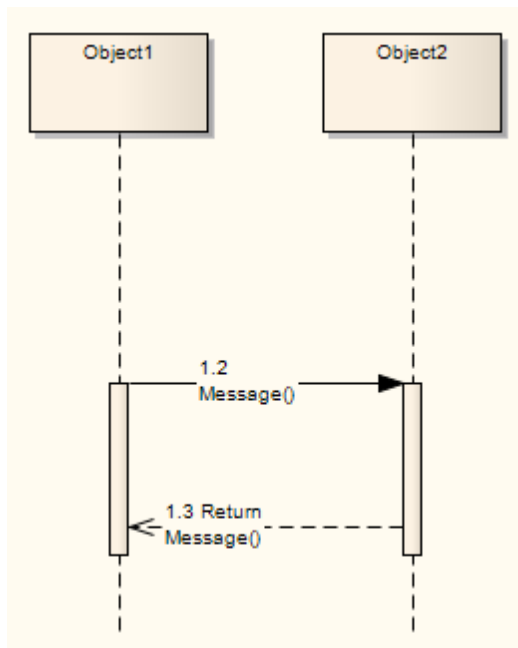
The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 491) states:

*A Message defines a particular communication between Lifelines of an Interaction.*

*A Message is a NamedElement that defines one specific kind of communication in an Interaction. A communication can be, for example, raising a signal, invoking an Operation, creating or destroying an Instance. The Message specifies not only the kind of communication given by the dispatching ExecutionSpecification, but also the sender and the receiver.*

*A Message associates normally two OccurrenceSpecifications - one sending OccurrenceSpecification and one receiving OccurrenceSpecification.*

### 6.5.19.1 Message (Sequence Diagram)



**Sequence** diagrams depict work flow or activity over time using **Messages** passed from element to element. These Messages correspond in the software model to Class operations and behavior. When you display a Sequence diagram, the Diagram Toolbox automatically switches to the **Interaction** Toolbox pages, containing the Message icon.

**Access** Click on the **Message** icon, click on the source object and drag the cursor to the target object (If the Message Properties dialog does not display, right-click on the Message and on the Message Properties menu option)

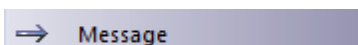
#### Create a Message on a Sequence diagram

Field/Option/ Button	Action	See also
<b>Message</b>	<p>Type the Message name.</p> <p>If the Message flow is <b>towards</b> a Class element (dropped in from a Class diagram) or a Lifeline element having a classifier, and the <b>destination</b> Class has defined <b>operations</b>, you can click on the drop-down arrow and select an appropriate operation name; the Message then reflects the destination Class operations.</p> <p>You can also include operations that the element's classifier has inherited, in the list. To do this, select the <b>Show Inherited Methods</b> checkbox (below).</p>	<a href="#">Class Element</a> <small>[1363]</small> <a href="#">Lifeline</a> <small>[1321]</small>
<b>Operations</b>	<p>If the available operations on the destination Class are not appropriate, click on this button and define a new operation in the destination element, using the Operations dialog.</p>	<a href="#">General Properties of Operations</a> <small>[1015]</small>

Field/Option/ Button	Action	See also
	If you create a Message without making reference to the target Class operations, no new operation is added to the target Class.	
<b>Parameters</b>	Type any parameters that the Message has, as a comma-separated list.	
<b>Argument(s)</b>	(Optional) Type the actual value that corresponds to each parameter, as a comma-separated list.	
<b>Return Value</b>	If the Message has a return value or type, specify it in this field.	
<b>Show Inherited Methods</b>	Select this checkbox to include operations that the destination element's classifier has inherited, in the drop-down list of operations available in the <b>Message</b> field.  Clear the checkbox to show only operations from the classifier itself.	
<b>Assign to</b>	If the Message flow is <b>from</b> a Class element or Lifeline element with classifier that has defined attributes, click on the drop-down arrow and select an appropriate attribute name.  The Message reflects the attributes from the source Class; you cannot add further attributes to the source Class here - if no appropriate attribute is listed, open the Class element Properties dialog and add the required attribute.  Otherwise, optionally type the name of the object to assign the message flow to.	
<b>Stereotype</b>	(Optional) Type or select a stereotype for the connector (this is displayed on the diagram, if entered).	
<b>Alias</b>	(Optional) Type an alias for the name of the Message.  On the diagram, the alias displays if the <b>Use Alias if Available</b> checkbox is selected on the Diagram tab of the Diagram Properties dialog.  The Alias displays instead of or as well as the Message name, depending on the setting selected in the Alias Usage panel of the Diagram Behavior page of the Options dialog.	<a href="#">Diagram Behavior Options</a> [625]
<b>Condition</b>	Type any conditions that must be <b>true</b> in order for the Message to be sent.	
<b>Constraint</b>	Type any constraints that might exist on when the Message is sent.	

Field/Option/ Button	Action	See also
<b>Is Iteration</b>	<p>Select the checkbox to indicate that the Message will iterate until the specified condition takes the value <b>false</b>. The condition statement on the diagram is prefixed by an asterisk (*).</p> <p>Clear the checkbox to indicate that the Message will only be sent once within the process cycle, if the specified condition is <b>true</b>.</p>	
<b>Synch</b>	<p>Click on the drop-down arrow and select <b>Synchronous</b> or <b>Asynchronous</b> as appropriate.</p> <p>The value <b>Synchronous</b> disables the <b>Kind</b> field (below), synchronous Messages are always Calls.</p>	<a href="#">Call</a> <sup>[1422]</sup>
<b>Kind</b>	<p>This field is enabled when the <b>Synch</b> field is set to <b>Asynchronous</b>.</p> <p>Click on the drop-down arrow and select either <b>Call</b> or <b>Signal</b>, as appropriate.</p>	<a href="#">Asynchronous Signal Message</a> <sup>[1427]</sup>
<b>Lifecycle</b>	<p>Select <b>New</b> to create a new element at the end of the Message, or <b>Delete</b> to terminate the message flow at the end of the Message.</p> <p>If neither case applies, set the field to <b>&lt;none&gt;</b>.</p>	
<b>Is Return</b>	If the Message you have created is a return message, select this checkbox.	
<b>Notes</b>	(Optional) Type any explanatory notes, formatted if you prefer.	
<b>OK</b>	<p>Click on this button to save the Message definition.</p> <ul style="list-style-type: none"> <li>You can change the timing details of a message on the Timing Details dialog, and emphasize the sequence of closely-ordered messages using <i>General Ordering</i></li> <li>To toggle the numbering of messages on a Sequence diagram, select or deselect the <b>Show Sequence Numbering</b> checkbox on the Options dialog</li> </ul>	<a href="#">Change the Timing Details</a> <sup>[1424]</sup> <a href="#">General Ordering</a> <sup>[1426]</sup> <a href="#">Sequence diagrams</a> <sup>[1248]</sup>
<b>Cancel</b>	Click on this button to close the dialog without saving any data you have entered.	

#### Toolbox icon



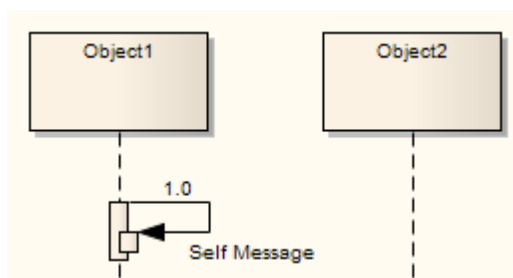
### Notes

- You can also use the Message connector as an Information Flow, and realize information flows on the Message

### Learn more

- [Message Examples](#)<sup>[1423]</sup>
- [Co-Region Notation](#)<sup>[1427]</sup>
- [Realize an Information Flow](#)<sup>[1414]</sup>

#### 6.5.19.1.1 Self-Message

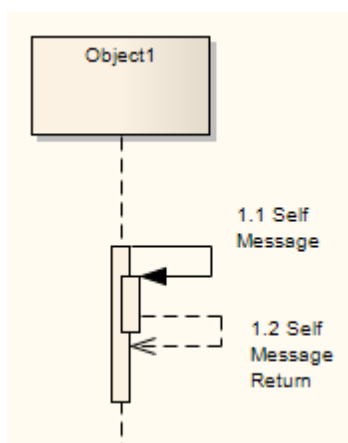


A Self-Message reflects a new process or method invoked within the calling lifeline's operation. It is a specification of a Message, typically in a Sequence diagram.

Self-Message Calls indicate a nested invocation; new activation levels are added with each Call.

#### Self-Message as Return

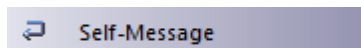
It is possible to depict a return from a Self Message call.



#### Create a Self Message return

Step	Action
1	Create a second Self Message at the end of execution.
2	Double-click on the Message name to open the Message Properties dialog.
3	Select the <b>Is Return</b> checkbox.
4	Raise the Activation level of the return.

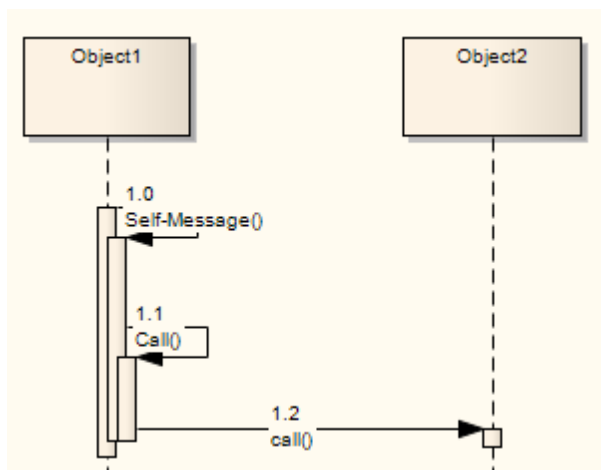
#### Toolbox icon



#### Learn more

- [Sequence Diagram](#) <sup>[1249]</sup>
- [Message](#) <sup>[1417]</sup>
- [Raise the Activation Level](#) <sup>[1256]</sup>
- [Call](#) <sup>[1422]</sup>

#### 6.5.19.1.2 Call



A Call is a type of Message connector that extends the level of activation from the previous Message. All Self-Messages create a new activation level, but this focus of control usually ends with the next Message (unless activation levels are manually adjusted). Self-Message Calls, as depicted above by the first Call, indicate a nested invocation; new activation levels are added with each Call. Unlike a regular Message between elements, a Call between elements continues the existing activation in the source element, implying that the

Call was initiated within the previous Message's activation scope.

#### Toolbox icon

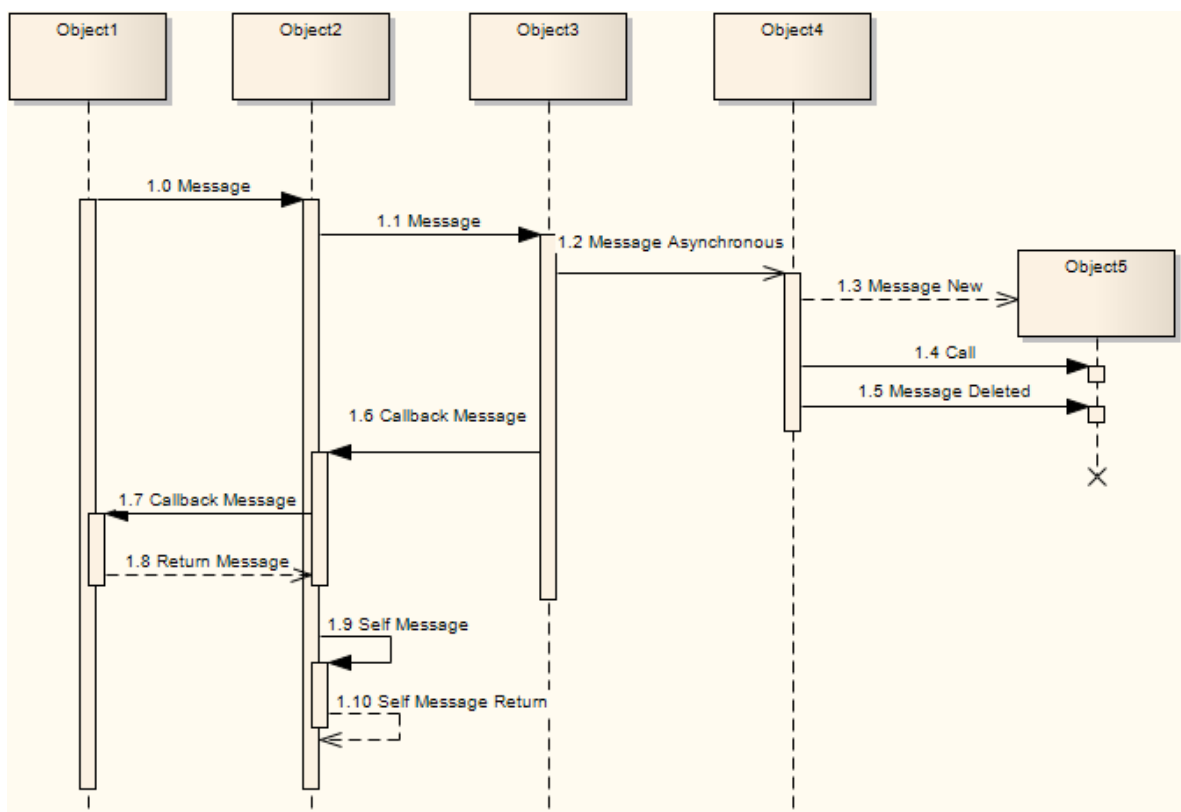


#### Learn more

- [Message \(Sequence Diagram\)](#)<sup>[1418]</sup>
- [Self-Message](#)<sup>[1421]</sup>
- [Sequence Element Activation](#)<sup>[1255]</sup>

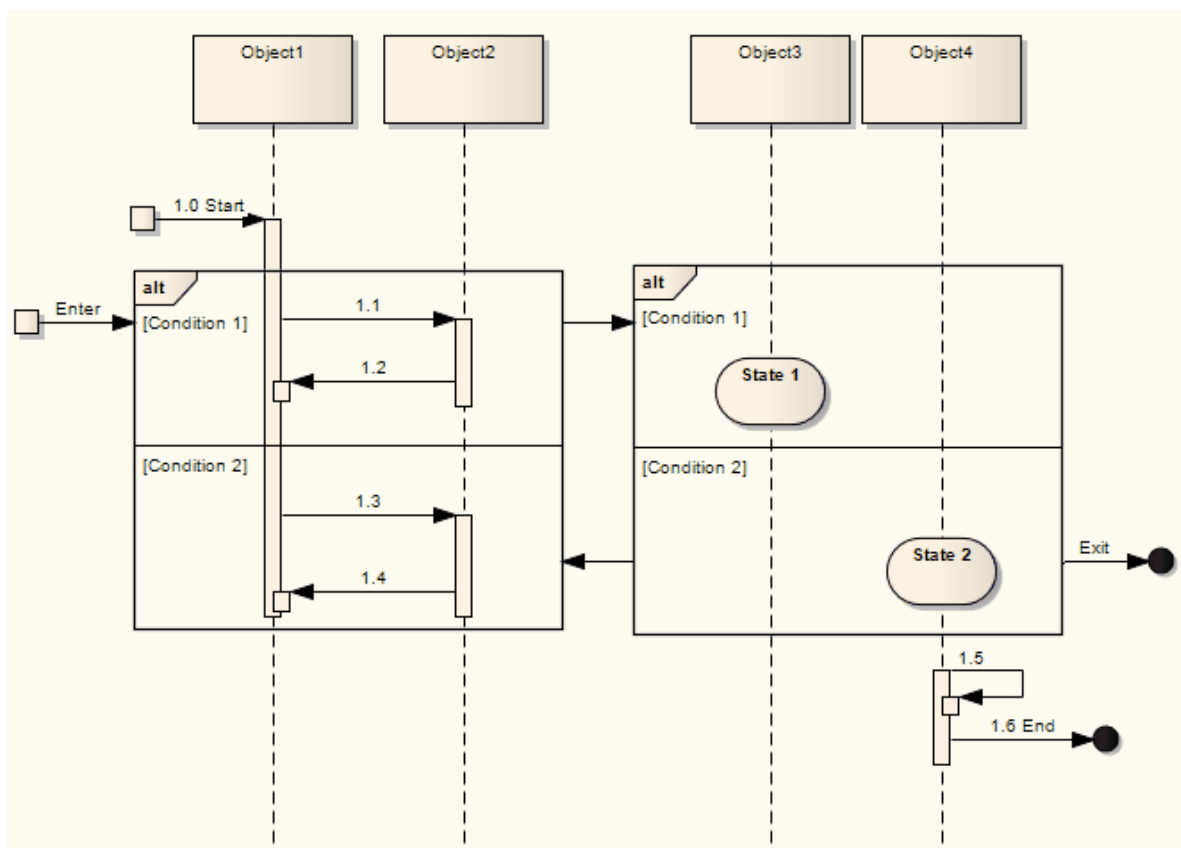
### 6.5.19.1.3 Message Examples

The following are different types of Messages available on Sequence Diagrams. Note that Messages on Sequence diagrams can also be modified with Shape Scripts.



#### Other Sequence Messages

The following are examples of Messages that are not part of the sequence described by the diagram.



#### Learn more

- [Messages](#) <sup>[1418]</sup>
- [Sequence Diagrams](#) <sup>[1249]</sup>
- [Shape Scripts](#) <sup>[1582]</sup>

#### 6.5.19.1.4 Change the Timing Details

It is possible to change the timing details of a Message in a Sequence diagram.

**Access** Right-click on the Message | Timing Details

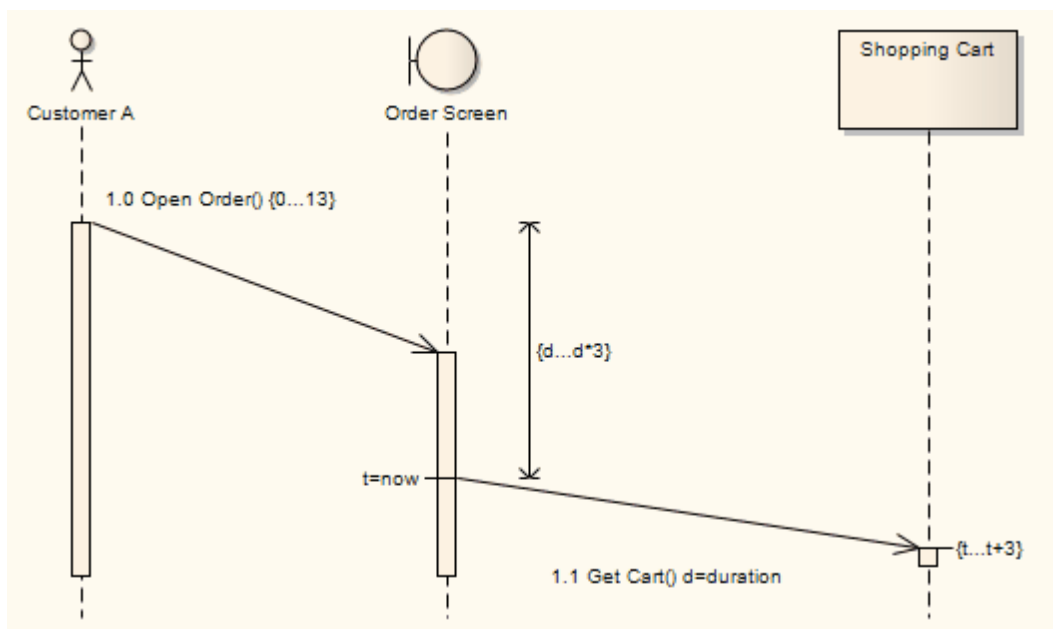
#### Change Timing

Field	Action
<b>Duration Constraint</b>	Indicate the minimum and maximum limits on how long a message can last.
<b>Duration Constraint Between Messages</b>	Indicate the minimum and maximum interval between sending or receipt of the previous message at the current message's source Lifeline, and sending the current message.



Field	Action
<b>Duration Observation</b>	Capture the duration of a message.
<b>Timing Constraint</b>	Indicate the minimum and maximum time at which the message should arrive at the target.
<b>Timing Observation</b>	Capture the point at which the message was sent.

See the OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 511*).



In this diagram, on the *Open Order* Message:

- **Duration Constraint** has been set to **0...13**

On the *Get Cart* Message:

- **Duration Constraint Between Messages** has been set to **d...d\*3**
- **Duration Observation** has been set to **d=duration**
- **Timing Constraint** has been set to **t...t+3**
- **Timing Observation** has been set to **t=now**

By typing a value in the **Duration Constraint** field, you enable the Message angle to be adjusted. After clicking on the **OK** button on the Timing Details dialog, click on the head of the Message connector and drag the connector up or down to change the angle. You cannot extent the angle beyond the life line of the connecting sequence object or create an angle of less than 5 degrees.

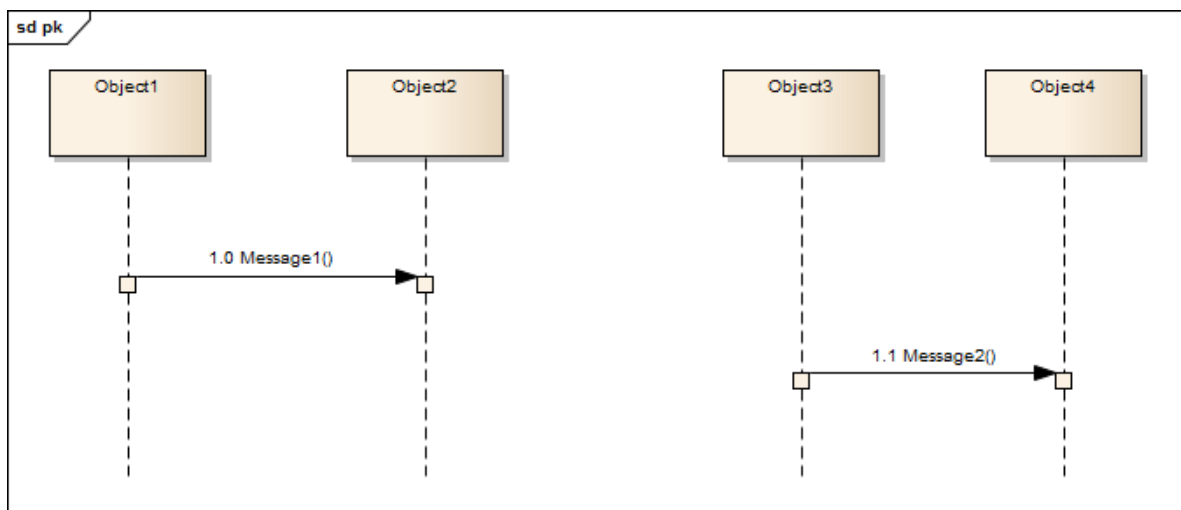
You can also create the **Duration Constraint Between Messages** line by dragging the General Ordering arrow up to the point at which the previous message joins the source Lifeline for the current message. A dialog displays on which you enter the value for the constraint. Having created the line, you can move it to any point within half way along the current message and half way along the previous message, to avoid overlap with other message timing details. You can edit or delete the value either **through the Timing Details** dialog or by right-clicking on the line itself and selecting the appropriate context menu option.

#### Learn more

- [Sequence Diagram](#)<sup>[1249]</sup>
- [Message](#)<sup>[1417]</sup>
- [General Ordering](#)<sup>[1426]</sup>

#### 6.5.19.1.5 General Ordering

In a Sequence diagram, the workflow is represented by the sequence of Messages down the diagram. Messages near the top of the diagram are passed before Messages lower down the diagram.

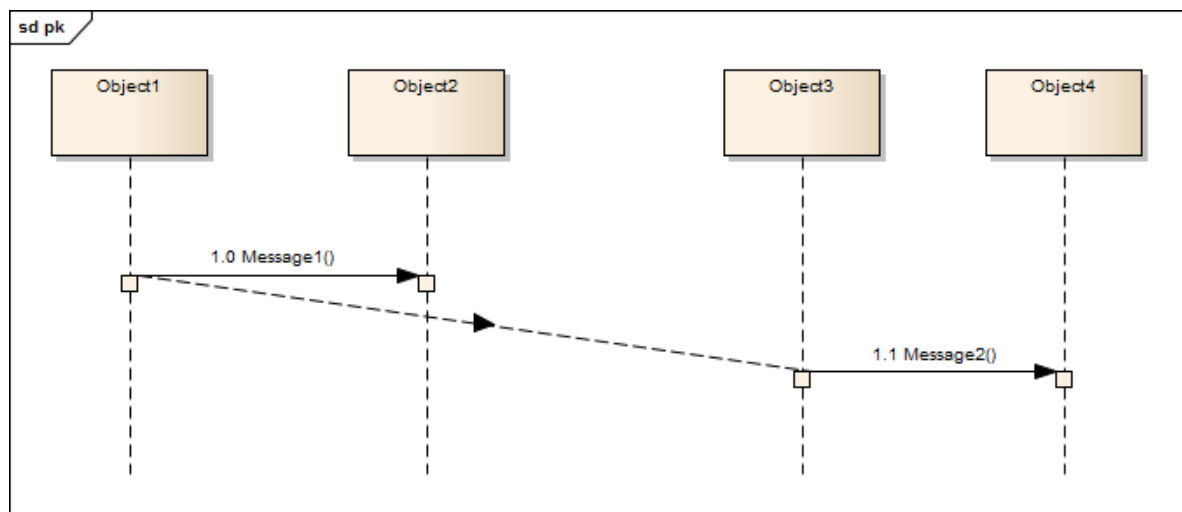


In the diagram, *Message 1* is earlier than *Message 2*. However, in a complex diagram, or when representing finely timed operations or parallel processing, this might not be apparent. You can reinforce the sequence using a **General Ordering arrow**.

Click on the Message arrow. A small arrow displays at the source anchor point.



Click on this arrow and drag it to the start of the next Message in sequence (*Message 2* in the example). The General Ordering arrow displays, indicating that the second Message follows the first.



You can have more than one General Ordering arrow issuing from or targeting a Message, if necessary.

#### 6.5.19.1.5.1 Asynchronous Signal Message

You define a Message as an asynchronous signal message by displaying the Message Properties dialog and setting the **Synch** field to **Asynchronous** and the **Kind** field to **Signal**. A *synchronous* message cannot be used to convey signals, so setting the **Synch** field to **Synchronous** disables the **Kind** field.

**Return Value**, **Assign To** and the **Operations** button, which are not applicable to asynchronous *signals*, are disabled.

The **Operations** button changes to a **Signal** button, which you click on to associate the asynchronous signal message with a Signal element in the model. You can type the arguments corresponding to the Signal attributes into the **Argument(s)** field.

When you click on the **Signal** button, the Select Signal dialog displays, through which you locate and select the required Signal element. (The Select Signal dialog is a variation of the Select <Item> dialog.)

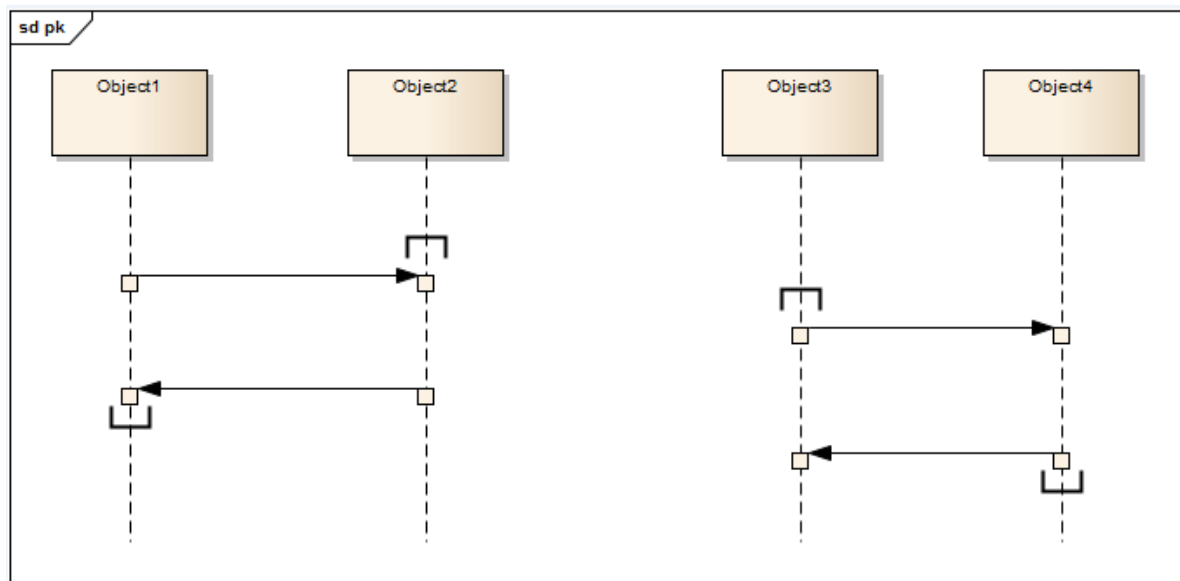
#### Learn more

- [Message Properties Dialog](#) <sup>[1418]</sup>

#### 6.5.19.1.6 Co-Region Notation

*Co-Region notation* can be used as a short hand for parallel combined fragments. You can add this notation to a Sequence diagram using the **Co-Region** submenu, which you display by right-clicking on a connector in a Sequence diagram and selecting the **Co-Region** context menu option. There are four sub-options available:

- **Start at head**
- **End at head**
- **Start at tail**
- **End at tail**



### 6.5.19.2 Message (Communication Diagram)

A Message in a Communication diagram is equivalent in meaning to a Message in a Sequence diagram. It implies that one object uses the services of another object, or sends a message to that object. Communication Messages in Enterprise Architect are always associated with an Association connector between object instances. Always create the Association first, then add a Message to the connector.

Messages can be dragged into a suitable position by clicking and dragging on the message text.

Communication Messages are ordered to reflect the sequencing of the diagram. The numbering scheme should reflect the nesting of each event. A sequencing scheme could be:

```
1
2, 2.1, 2.2, 2.3
3.
```

This would indicate the single sequence of events 2.1, 2.2 and 2.3 occurs within an operation initiated by event 2. This is the default pattern applied by Enterprise Architect.

Alternatively, the sequence could be:

```
1
2, 2.1, 2.1.1, 2.1.1.1
    2.2, 2.2.1, 2.2.1.1
3
```

This would indicate that two sequences of events can be initiated by event 2, and 2.1 and 2.2 are separate sequences, not consecutive events in one sequence. You can set the sequence pattern and order using the Message Properties dialog and the Sequence Communications dialog.

If the target object is a Class or has its instance classifier set, the drop-down list of possible message names includes the exposed operations for the base type.

#### Learn more

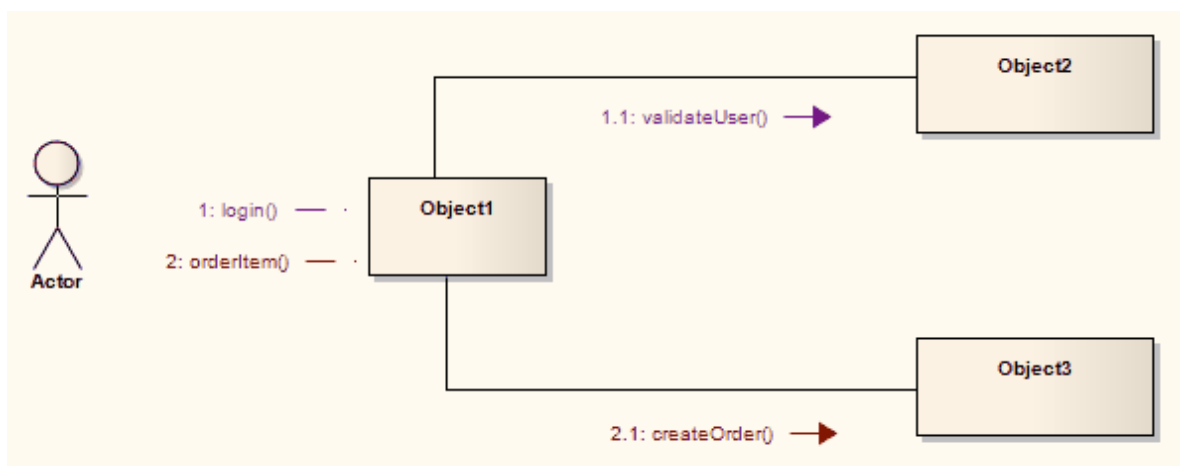
- [Communication Diagram](#) 

- [Create a Communication Message](#) <sup>[1429]</sup>
- [Re-Order Messages](#) <sup>[1430]</sup>

### 6.5.19.2.1 Create a Communication Message

#### Create a Communication Message

Step	Action
1	Open a diagram (one of: Communication, Analysis, Interaction Overview, Object, Activity or State Machine).
2	Add the required objects.
3	Add an Association relationship between each pair of objects that communicate.
4	Right-click on an Association to display the context menu.
5	Select the option to add a Message from one object to the other.
6	When the Message Properties dialog displays, type in a name and any other required details.
7	Click on the <b>OK</b> button. The Message is added, connected to the Association and Object instances.
8	Move the Message to the required position.



Learn more

- [Communication Message](#)<sup>[1428]</sup>
- [Message \(Sequence Diagram\)](#)<sup>[1418]</sup>

**6.5.19.2.2 Re-Order Messages**

When constructing your Communication diagram, it is frequently necessary to create or delete Message 'groups' and to re-order the sequence of Messages. There are two dialogs that help you perform these tasks: the Message Properties dialog and the Sequence Communications dialog.

**Organize Message Groups**

If you have several Messages in the form 1.1, 1.2, 1.3, 1.4, for example, but would like to start a new numbering group on, say, the third Message (that is, 1.1, 1.2, **2.1**, 2.2, 2.3 ), you can change a Message in the series to a *Start Group* message.

Step	Action
<b>1</b>	Double-click on a Message <i>name</i> . The Message Properties dialog displays.
<b>2</b>	To make the selected Message the start of a new group, select the <b>Start New Group</b> checkbox.
<b>3</b>	If required, in the <b>Notes</b> field, type an explanatory note. You can format the text using the Notes toolbar at the top of the field.
<b>4</b>	Click on the <b>OK</b> button to save changes.

**Sequence Messages**

In larger and more complex diagrams, you might have to use deeper levels of Messages in a group; for example, 1, 1.2, 1.2.1, 1.2.1.1. You might also have to change the sequence of Messages, making Message 1.3, for example, into Message 1.1.

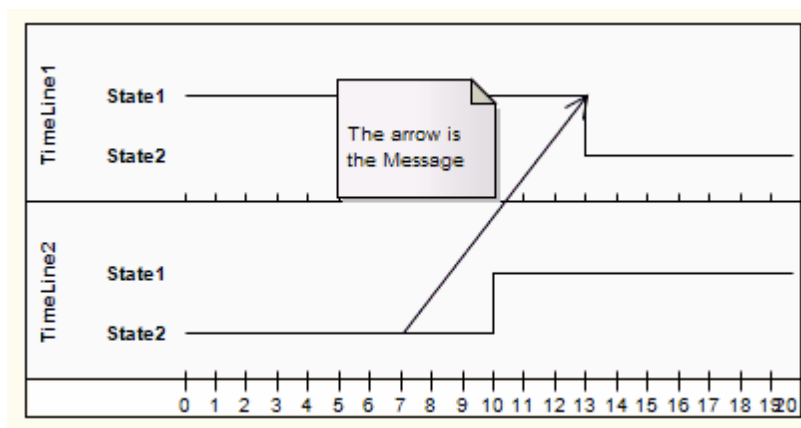
Step	Action
<b>1</b>	Either: <ul style="list-style-type: none"> <li>• Select the <b>Diagram   Advanced   Sequence Messages</b> menu option</li> <li>• Click on the diagram background and select the <b>Sequence Communication Messages</b> context menu option or</li> <li>• Right-click on a Message and select the <b>Sequence Communication Messages</b> context menu option</li> </ul>

Step	Action
	The Communication Messages dialog displays.
2	Click on the Message to adjust and, at the bottom of the dialog, click on the: <ul style="list-style-type: none"> <li>• <b>Move Up</b> or <b>Move Down</b> (Hand) buttons to move the Message up or down the sequence (e.g. Message 1.2 to Message 1.1 or 1.3)</li> <li>• <b>Move Left</b> or <b>Move Right</b> (Hand) buttons to move the Message up or down a level (e.g. Message 1.2.1 to Message 1.2 or Message 1.2.1.1)</li> </ul>
3	Repeat step 2 until the Message sequence and levels match your requirements. You might have to adjust other Message numbers (in group, sequence or level) to accommodate the changes you have made.
4	Click on the <b>OK</b> button to save changes.

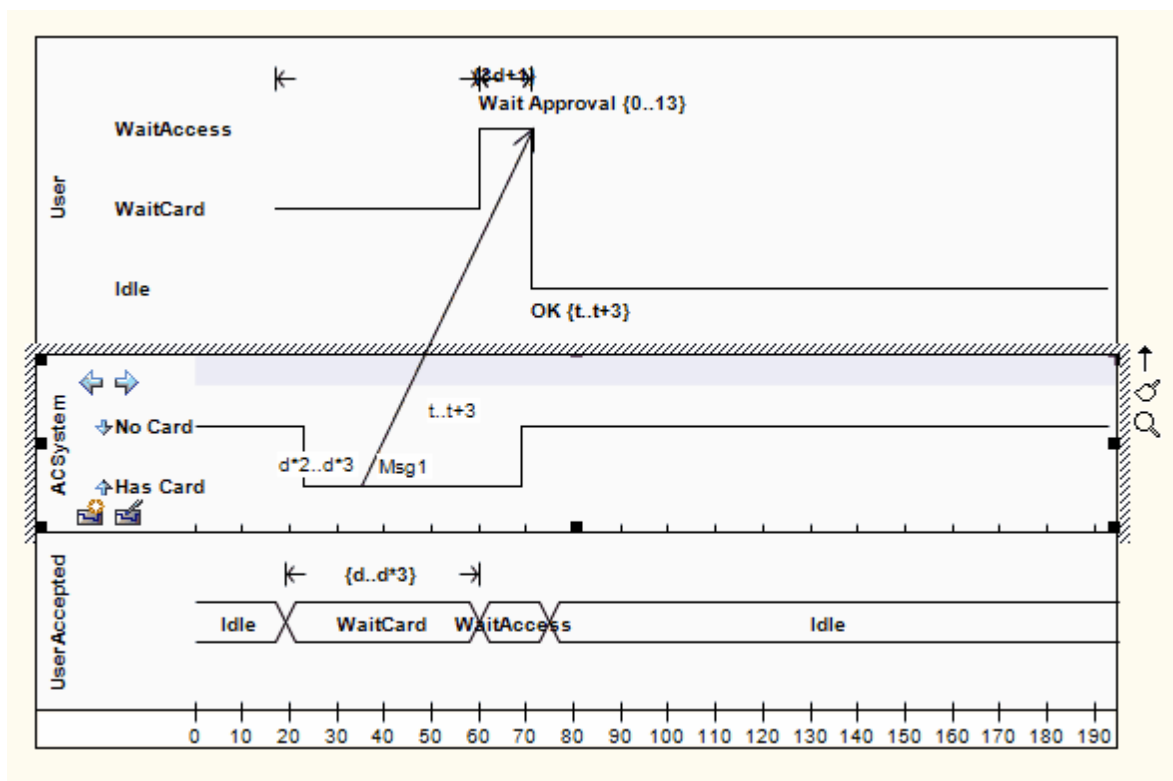
#### Learn more

- [Communication Message](#) 

### 6.5.19.3 Message (Timing Diagram)

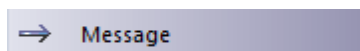


**Messages** are the communication links between Lifelines in a Timing diagram. In the case of a Timeline, a Message is a connection between two Timeline objects.



See UML Superstructure Specification, v2.1.1, figures 14.30 and 14.31, p. 520.

#### Toolbox icon



#### Learn more

- [Timing Diagram](#) <sup>[1225]</sup>
- [Lifeline](#) <sup>[1321]</sup>
- [Create a Timing Message](#) <sup>[1432]</sup>

#### 6.5.19.3.1 Create a Timing Message

You can create a Timing Message between two Lifeline objects (State or Value) on a Timing diagram, each with existing transition points.

#### Create a Message between Lifelines

Step	Action
1	Click on the Message icon on the Timing Relationships page of the Toolbox ( <b>More tools   Timing</b> ).
2	Click on the source Lifeline at the point at which the Message will start, and drag the cursor to the transition point on the destination Lifeline where the Message will end.



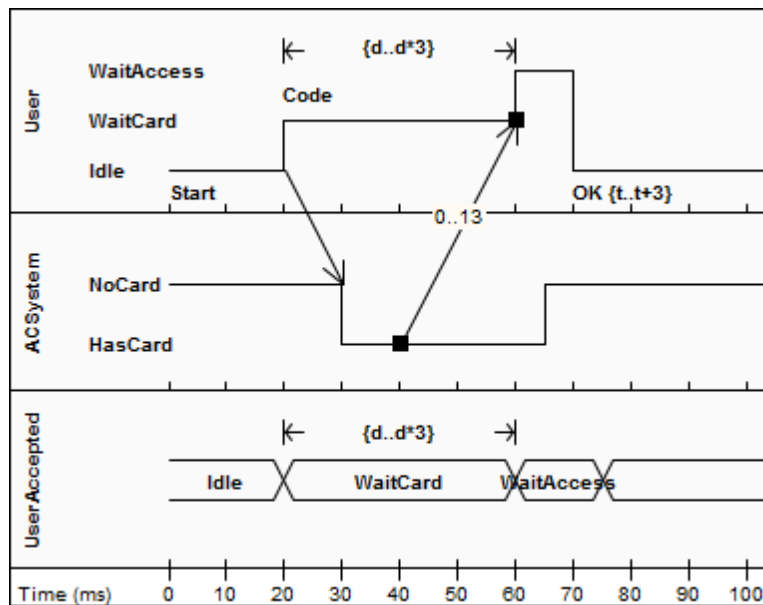
Step	Action
	A new Timing Message is created between these two points.
<b>3</b>	Double-click on the new Message to open the Timing Message dialog. Review or complete the dialog as indicated in the Dialog Fields table.

#### Dialog Fields

Field/Button	Action
<b>Start</b>	Identifies the Lifeline from which the Message originates.
<b>End</b>	Identifies the Lifeline on which the Message terminates.
<b>Start Time</b>	Shows the time after the timeline begins at which the Message starts. You can change this if you need to.
<b>End Time</b>	Shows the time after the timeline begins at which the Message ends. You can change this if you need to, but the time must correspond to a transition point on the target Lifeline.
<b>Name</b>	(Optional) Type in a name for the Message.
<b>Time Observation</b>	(Optional) Type any text to act as a label providing information on when the Message is sent.
<b>Duration Observation</b>	(Optional) Type any text to act as a label providing information on the interval of a Lifeline at a particular state, begun from receipt of the Message.
<b>Transition To</b>	The state in the target Lifeline that the Message terminates on. If necessary, you can click on the drop-down arrow and select a different state to transition to. The head of the Message moves accordingly.
<b>Event</b>	(Optional) Type in the name of any event that triggers the transition.
<b>Time Constraint</b>	(Optional) Type in the maximum time it can take to transmit the Message.
<b>Duration Constraint</b>	(Optional) Type in the maximum time the Lifeline can remain in the changed state after receipt of the Message.

Field/Button	Action

This diagram shows an example of a configured Message:



(See *UML Superstructure Specification*, v2.1.1, figures 14.30 and 14.31, p. 520.)

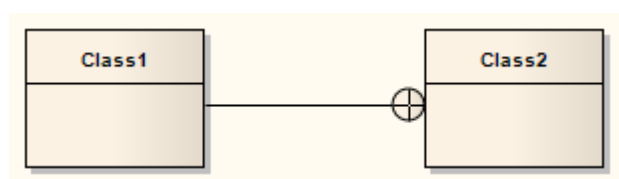
### Notes

- You can move the source end of the Message freely along the source timeline; however, the target end (arrow head) must attach to a transition
- If you create a new Message and do not give it a target transition, it automatically finds and attaches to the nearest transition; if you move the target end, it drags the transition with it

### Learn more

- [Timing Diagram](#) <sup>[1225]</sup>
- [Message \(Timing Diagram\)](#) <sup>[1431]</sup>
- [State Lifeline](#) <sup>[1335]</sup>
- [Value Lifeline](#) <sup>[1355]</sup>

## 6.5.20 Nesting



### Description

The Nesting Connector is an alternative graphical notation for expressing containment or nesting of elements within other elements. It is most appropriately used for displaying Package nesting in a Package diagram.

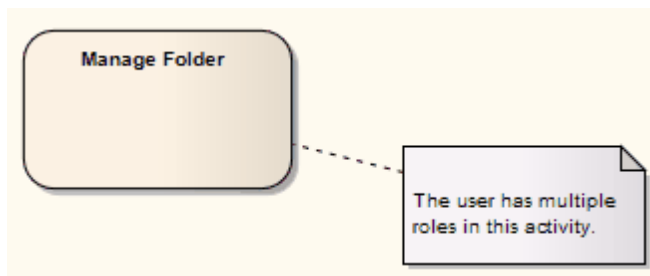
### Toolbox icon



### Learn more

- [Package Diagram](#) f1182

## 6.5.21 Notelink

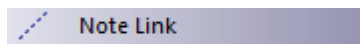


### Description

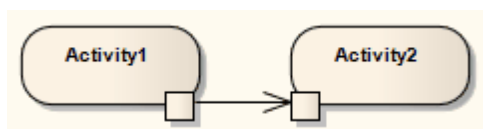
A Notelink connector connects a Note to one or more other elements of any other type.

Both Note and Notelink are available in any category of the Toolbox, in the Common page. You can also select them from the UML Elements toolbar.

### Toolbox icon



## 6.5.22 Object Flow



### Description

*Object Flows* are used in Activity diagrams and State Machine diagrams. When used in an Activity diagram, an Object Flow connects two elements, with specific data passing through it. To view sample Activity

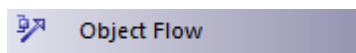
diagrams using Object Flows, see the *Object Flows in Activity Diagrams* topic.

In State Machine diagrams, an Object Flow is a specification of a state flow or transition. It implies the passing of an Object instance between elements at run-time.

You can insert an Object Flow from the State or Activity pages of the Toolbox, or from the drop-down list of all relationships located in the header toolbar. You can also modify a transition connection to an Object Flow by selecting the **ObjectFlow** checkbox on the connection Properties dialog.

See the *Control Flow* topic for information on setting up Guards and Weights on Object Flows.

#### Toolbox icon



#### Learn more

- [Activity Diagram](#) <sup>[1199]</sup>
- [Control Flow](#) <sup>[1403]</sup>
- [Object Flows in Activity Diagrams](#) <sup>[1436]</sup>
- [Object Node](#) <sup>[1325]</sup>

#### OMG UML Specification:

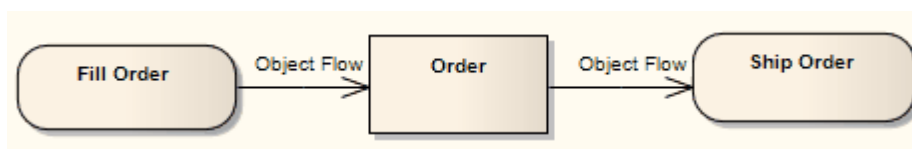
The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 389*) states:

*An object flow is an activity edge that only passes object and data tokens.*

### **6.5.22.1 Object Flows in Activity Diagrams**

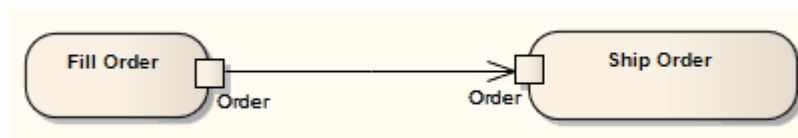
In Activity diagrams, there are several ways to define the flow of data between objects.

The following diagram depicts a simple Object Flow between two actions, *Fill Order* and *Ship Order*, both accessing order information.



See *UML Superstructure Specification, v2.1.1, figure 12.110, p. 391*.

This explicit portrayal of the data object *Order*, connected to the Activities by two Object Flows, can be refined by using the following format. Here, Action Pins are used to reflect the order.



See *UML Superstructure Specification*, v2.1.1, figure 12.110, p. 391.

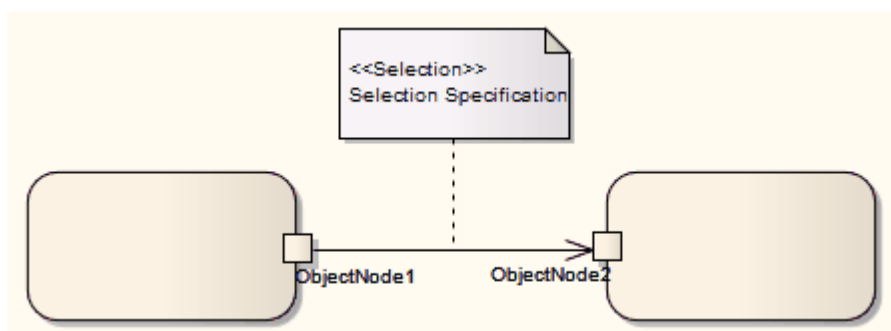
The following diagram is an example of multiple Object Flows exchanging data between two actions.



See *UML Superstructure Specification*, v2.1.1, figure 12.111, p. 391.

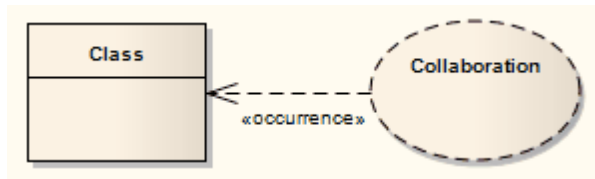
Selection and transformation behavior, together composing a sort of query, can specify the nature of the Object Flow's data access. Selection behavior determines which objects are affected by the connection. Transformation behavior might then further specify the value of an attribute pertaining to a selected object.

Selection and transformation behaviors can be defined by attaching a note to the Object Flow. To do this, right-click on the Object Flow and select the **Attach Note or Constraint** context menu option. A dialog lists other flows in the diagram, to which you can select to attach the note, if the behavior applies to multiple flows. To comply with UML 2, preface the behavior with the notation «*selection*» or «*transformation*».



See *UML Superstructure Specification*, v2.1.1, figure 12.112, p. 392.

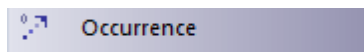
### 6.5.23 Occurrence



#### Description

An *Occurrence* relationship indicates that a Collaboration represents a classifier, in a Composite Structure diagram. An Occurrence connector is drawn from the Collaboration to the classifier.

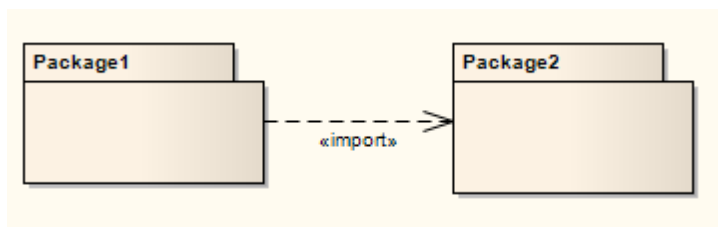
#### Toolbox icon



#### Learn more

- [Collaboration](#) <sup>[1367]</sup>
- [Composite Structure Diagram](#) <sup>[1188]</sup>

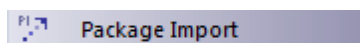
### 6.5.24 Package Import



#### Description

A Package Import relationship is drawn from a source Package to a Package whose contents are to be imported. Private members of a target Package cannot be imported. The relationship is typically used in a Package diagram.

#### Toolbox icon



#### Learn more

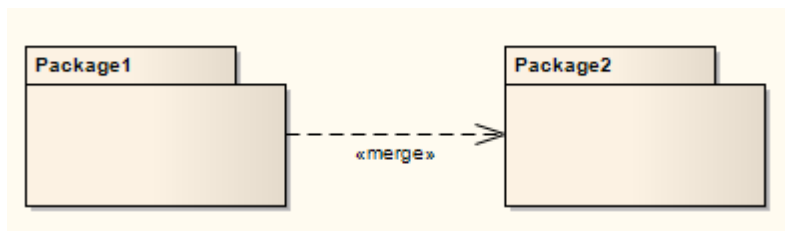
- [Package](#) <sup>[1382]</sup>
- [Package Diagram](#) <sup>[1182]</sup>

**OMG UML Specification:**

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 112*) states:

*A package import is a relationship between an importing namespace and a package, indicating that the importing namespace adds the names of the members of the package to its own namespace. Conceptually, a package import is equivalent to having an element import to each individual member of the imported namespace, unless there is already a separately-defined element import.*

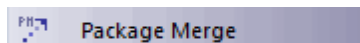
### 6.5.25 Package Merge

**Description**

In a Package diagram, a Package Merge indicates a relationship between two Packages whereby the contents of the target Package are merged with those of the source Package. Private contents of a target Package are not merged. The applicability of a Package Merge addresses any situation where multiple packages contain identically-named elements, representing the same thing. A Package Merge merges all matching elements across its merged Packages, along with their relationships and behaviors. Note that a Package Merge essentially performs generalizations and redefinitions of all matching elements, but the merged Packages and their independent element representations still exist and are not affected.

The Package Merge serves a graphical purpose in Enterprise Architect, but creates an ordered Package relationship applied to related Packages (which can be seen under the Link tab in the Package's Properties dialog). Such relationships can be reflected in XML exports or Enterprise Architect Automation Interface scripts for code generation or other Model Driven Architecture (MDA) interests.

Package Merge relationships are useful to reflect situations where existing architectures contain functionalities involving like elements, which are merged in a developing architecture. Merging doesn't affect the merged objects, and supports the common situation of product progression.

**Toolbox icon****Learn more**

- [Package Diagram](#) <sup>[1182]</sup>
- [Package](#) <sup>[1382]</sup>

**OMG UML Specification:**

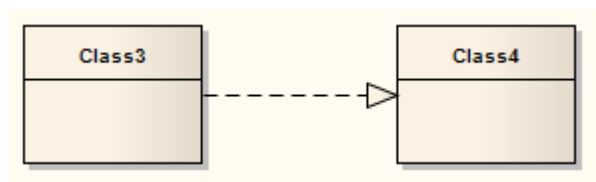
The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 113-114*) states:

A package merge is a directed relationship between two packages that indicates that the contents of the two packages are to be combined. It is very similar to Generalization in the sense that the source element conceptually adds the characteristics of the target element to its own characteristics resulting in an element that combines the characteristics of both.

This mechanism should be used when elements defined in different packages have the same name and are intended to represent the same concept. Most often it is used to provide different definitions of a given concept for different purposes, starting from a common base definition. A given base concept is extended in increments, with each increment defined in a separate merged package. By selecting which increments to merge, it is possible to obtain a custom definition of a concept for a specific end. Package merge is particularly useful in meta-modeling and is extensively used in the definition of the UML metamodel.

Conceptually, a package merge can be viewed as an operation that takes the contents of two packages and produces a new package that combines the contents of the packages involved in the merge. In terms of model semantics, there is no difference between a model with explicit package merges, and a model in which all the merges have been performed.

### 6.5.26 Realization



#### Description

A source object implements or Realizes its destination object. Realize connectors are used in a Use Case, Component or Requirements diagram to express traceability and completeness in the model. A business process or Requirement is realized by one or more Use Cases, which in turn are realized by some Classes, which in turn are realized by a Component, and so on. Mapping Requirements, Classes and such across the design of your system, up through the levels of modeling abstraction, ensures the big picture of your system remembers and reflects all the little pictures and details that constrain and define it.

You can also define template binding parameters for a Realize connector between a binding Class and a parameterized Class.

#### Toolbox icon



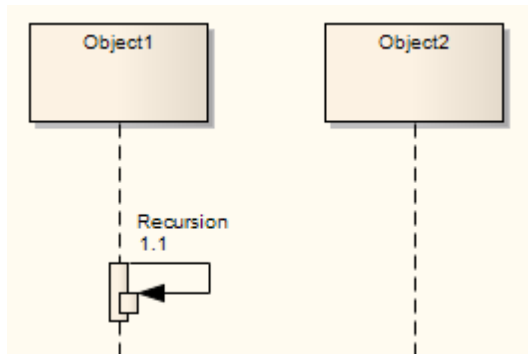
#### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 131) states:

*A Realization signifies that the client set of elements are an implementation of the supplier set, which serves as the specification. The meaning of 'implementation' is not strictly defined, but rather implies a more refined or elaborate form in respect to a certain modeling context. It is possible to specify a mapping between the specification and implementation elements, although it is not necessarily computable.*



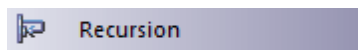
### 6.5.27 Recursion



#### Description

A Recursion is a type of Message used in Sequence diagrams to indicate a recursive function.

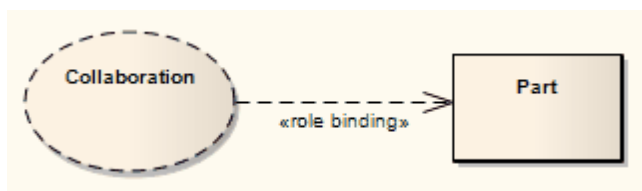
#### Toolbox icon



#### Learn more

- [Message](#) <sup>[1418]</sup>
- [Sequence Diagram](#) <sup>[1249]</sup>

### 6.5.28 Role Binding

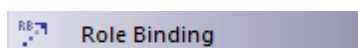


#### Description

Role Binding is the mapping between a Collaboration Use's internal roles and the respective Parts required to implement a specific situation, typically in a Composite Structure diagram. The associated Parts can have properties defined to enable the binding to occur, and the Collaboration to take place.

A Role Binding connector is drawn between a Collaboration and the classifier's fulfilling roles, with the Collaboration's internal binding roles labeled on the classifier end of the connector.

#### Toolbox icon



### Learn more

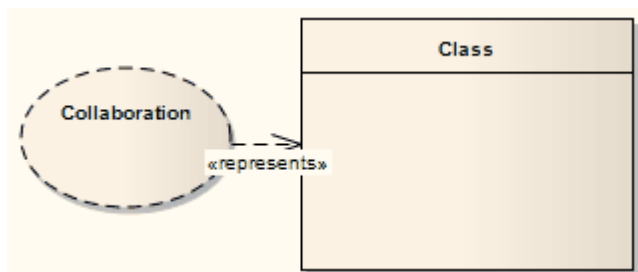
- [Collaboration Use](#) <sup>[1368]</sup>
- [Composite Structure Diagram](#) <sup>[1188]</sup>
- [Collaboration](#) <sup>[1367]</sup>

### OMG UML Specification:

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 174) states:

*A mapping between features of the collaboration type and features of the classifier or operation. This mapping indicates which connectable element of the classifier or operation plays which role(s) in the collaboration. A connectable element may be bound to multiple roles in the same collaboration use (that is, it may play multiple roles).*

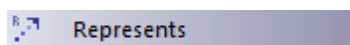
## 6.5.29 Represents



### Description

The Represents connector indicates that a Collaboration is used in a classifier, typically in a Composite Structure diagram. The connector is drawn from the Collaboration to its owning classifier.

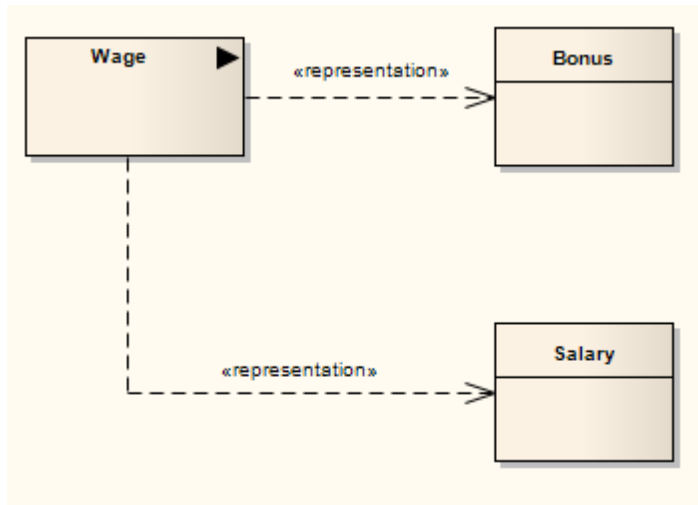
### Toolbox icon



### Learn more

- [Collaboration](#) <sup>[1367]</sup>
- [Composite Structure Diagram](#) <sup>[1188]</sup>

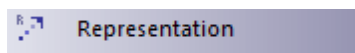
### 6.5.30 Representation



#### Description

The Representation relationship is a specialization of a Dependency, connecting Information Item elements that represent the same idea across models, typically in an Analysis diagram. For example, Bonus and Salary are both a representation of the Information Item Wage.

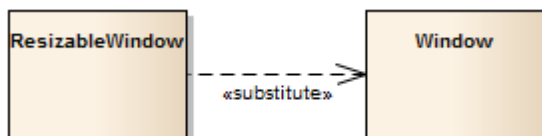
#### Toolbox icon



#### Learn more

- [Dependency](#) <sup>[1404]</sup>
- [Information Item](#) <sup>[1376]</sup>
- [Analysis diagram](#) <sup>[1801]</sup>

### 6.5.31 Substitution

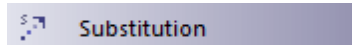


#### Description

A Substitution is a relationship between two Classifiers, signifying that the **substituting** Classifier complies with the **contract** specified by the **contract** Classifier. This implies that instances of the substituting Classifier are runtime-substitutable, where instances of the contract Classifier are expected. Above, the Class named *ResizableWindow* has a Substitution connector to the Class named *Window*, meaning that wherever you are asked for a window you can use a resizable window.

The Substitution relationship is a subtype of a Dependency relationship.

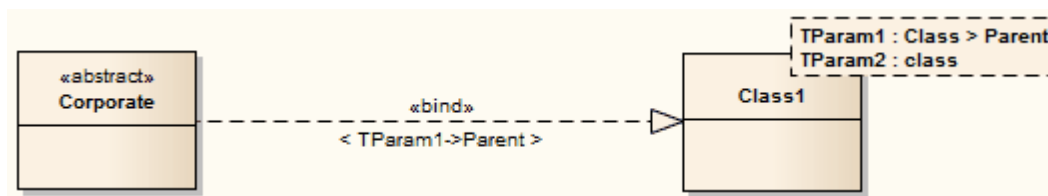
#### Toolbox icon



#### Learn more

- [Class Diagram](#) <sup>[1184]</sup>
- [Dependency](#) <sup>[1404]</sup>

### 6.5.32 Template Binding



#### Description

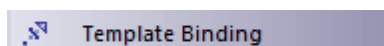
You create a Template Binding connector between a binding Class and a parameterized Class. You then define a binding expression on that connector. However, if the binding Class requires a Generalization, Realization or Association relationship with the parameterized Class, you can define the binding expression on that relationship instead.

You can create a Template Binding connector using:

- The **Template Binding** icon on the Class Relationships page of the Diagram Toolbox
- The Quicklinker arrow next to the source Class element
- The Templates tab of the binding Class element Properties dialog; here, you create the Template Binding relationship by clicking the **Add** button under the Binding(s) panel, specifying the connector type, and selecting the target parameterized Class from the Select <Item> dialog

Each of these methods creates the connector itself. For the first two methods you then double-click on the connector to display the connector Properties dialog, on which you select the Binding tab to define *parameter substitutions* as the binding expression. The third method takes you to this dialog and tab automatically.

#### Toolbox icon



#### Learn more

- [Parameterized Class](#) <sup>[1365]</sup>
- [Parameter Substitution](#) <sup>[1445]</sup>

**OMG UML Specification:**

The OMG UML specification (*UML Superstructure Specification*, v2.1.1, p. 622) states:

*A template is a parameterized element ... used to generate other model elements using TemplateBinding relationships. The template parameters for the template signature specify the formal parameters that will be substituted by actual parameters (or the default) in a binding.*

**6.5.32.1 Parameter Substitution**

Once a Template Binding (or other binding) relationship exists, you can add parameter substitutions to identify the formal parameters that are replaced, and the actual parameters that replace them, in the binding expression.

**How to**

To define a parameter substitution

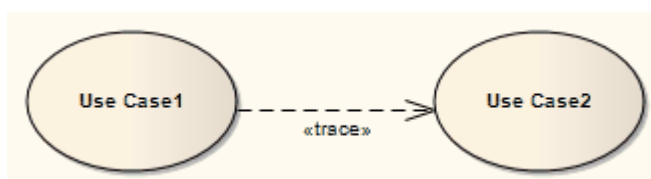
Step	Action
<b>1</b>	Display the Binding tab, in the Template Binding connector Properties dialog. The <b>Target</b> field identifies the target parameterized Class.
<b>2</b>	Click on the <b>Add</b> button below the Parameter Substitution(s) panel. The next available row in the panel is enabled for editing, and the word <i>&lt;none&gt;</i> is displayed in the <b>Formal</b> column.
<b>3</b>	Click on the field and on the drop-down arrow that is now displayed. A list of the template parameters from the target Class displays; click on the required parameter.
<b>4</b>	Click on the ( ... ) button in the corresponding <b>Actual</b> field for the parameter. If the template parameter: <ul style="list-style-type: none"> <li>Does not have a constraint, a short context menu displays offering the choice of typing a free-text value into the <b>Actual</b> field, or selecting a classifier from the Select Classifier dialog</li> <li>Has a constraint defined, the Select Classifier dialog displays automatically, showing the available classifiers</li> </ul>
<b>5</b>	Locate and select the required classifier to replace the parameter in the binding expression. If you do not define an <b>Actual</b> classifier <i>and</i> the template parameter has a default value defined, that default is used in the expression.
<b>6</b>	To edit existing parameter substitutions, click on them and make the required changes as indicated in steps 3 and 4.

Step	Action
7	Click on the <b>Apply</b> and/or <b>OK</b> button. The parameter substitutions display as a label underneath the connector.

#### Learn more

- [Template Binding](#)<sup>[1444]</sup>

### 6.5.33 Trace



#### Description

The Trace relationship is a specialization of an Abstraction, connecting model elements or sets of elements that represent the same concept across models. Traces are often used to track requirements and model changes, typically in a Traceability diagram, or in a Class, Use Case, Object or Composite Structure diagram.

As changes can occur in both directions, the order of this Trace is usually ignored. The relationship's properties can specify the trace mapping, but the trace is usually bi-directional, informal and rarely computable.

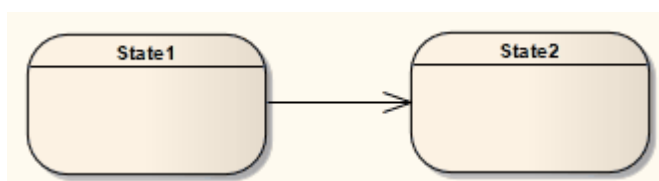
#### Toolbox icon



#### Learn more

- [Abstraction](#)<sup>[1391]</sup>
- [Example Traceability Diagram](#)<sup>[743]</sup>

### 6.5.34 Transition



**Description**

If you need to define the logical movement from one State to another in a State Machine diagram, you can drag a *Transition* connector from the Toolbox onto the diagram. You control the Transition through the connector Properties dialog.

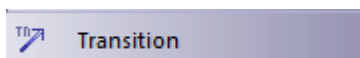
Field	Action	See also
<b>Guard</b>	Type in the expression to be evaluated <b>after</b> an Event is dispatched but <b>before</b> the corresponding Transition is triggered.  If the guard is true at that time, the Transition is enabled; otherwise, it is disabled.	
<b>Effect is a Behavior</b>	Convert the <b>Effect</b> field from a free-text field to the definition of a specific Activity or behavior.  The Select <Item> dialog displays, prompting you to select the Activity or behavior element from the model.	<a href="#">Select &lt;Item&gt; Dialog</a> <sup>[994]</sup>
<b>Effect</b>	Either: <ul style="list-style-type: none"> <li>Type a description of the effect of the Transition, or</li> <li>If you have selected the <b>Effect is a Behavior</b> check box, select an Activity or behavior to be performed during the Transition (to change this subsequently, click on the ( ... ) button to redisplay the Select &lt;Item&gt; dialog)</li> </ul>	
<b>Trigger</b>		
<b>Name</b>	Specify the name of the trigger; either: <ul style="list-style-type: none"> <li>Type the name, or</li> <li>Select an existing trigger in the model from the Select &lt;Item&gt; dialog, which you display by clicking on the ( ... ) button</li> </ul>	
<b>Type</b>	Specify the type of trigger: <ul style="list-style-type: none"> <li><b>Call</b> - specifies that the event is a <b>CallEvent</b>, which sends a message to the associated object by invoking an operation</li> <li><b>Change</b> - specifies that the event is a <b>ChangeEvent</b>, which indicates that the transition is the result of a change in value of an attribute</li> <li><b>Signal</b> - specifies that the event is a <b>SignalEvent</b>, which corresponds to the receipt of an asynchronous signal instance</li> <li><b>Time</b> - corresponds to a <b>TimeEvent</b>; which specifies a moment in time</li> </ul> <p>Code generation for State Machines expects a specification value for any of the four types.</p>	

Field	Action	See also
<b>Specification</b>	Specify the event instigating the Transition; either: <ul style="list-style-type: none"> <li>Type the event (time or change), or</li> <li>Select an existing specification in the model using the Select &lt;Item&gt; dialog, which you display by clicking on the ( ... ) button</li> </ul>	
<b>New</b>	Clear the fields ready to begin defining a new trigger.	
<b>Save</b>	Save the newly created or edited trigger.	
<b>Delete</b>	Remove the selected trigger from the list.	
<b>&lt;trigger list&gt;</b>	List the existing triggers, which might or might not have names and types, and which can include triggers created in older models.	

### Notes

- Fork and Join segments can have neither triggers nor guards
- You can identify hidden triggers and locate triggers in the Project Browser, using the **Find Triggers Associated** option on the Transition connector context menu; if one trigger exists for the Transition it is immediately highlighted in the Project Browser, if more than one trigger exists the Element Usage dialog displays - select the required trigger and click on the **Open** button to highlight the trigger in the Project Browser
- You can define a self-Transition as an **Internal Transition**, and represent the connector and its properties in a compartment of the State element

### Toolbox icon



### Learn more

- [State Machine Diagrams](#) <sup>f203</sup>
- [Internal Transition](#) <sup>f448</sup>

### OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.1.1, p. 568*) states:

*A transition is a directed relationship between a source vertex and a target vertex. It may be part of a compound transition, which takes the state machine from one state configuration to another, representing the complete response of the state machine to an occurrence of an event of a particular type.*





### 6.5.34.1 Internal Transition

If you need to define an **internal Transition** in a State, you can do so by creating an external **self-Transition** connector (where the **Source** and **Target** are the same State) and then changing the connector **kind** property. The self-Transition connector is then removed from the diagram and the internal Transition displays in a **compartment** inside the State element.

**Access**    **Open the State Machine diagram containing the State element**

#### Define an Internal Transition

Step	Action	See also
1	On the State element, create a Transition connector issuing from and terminating in the element (a self Transition).  Right-click on the connector and select the <b>Properties</b> context menu option to display the Properties dialog.	
2	Select the Constraints page and define any guard, effect and trigger for the Transition.	<a href="#">Transition</a> 
3	On the Advanced page, click on the drop-down arrow in the value field for the <b>kind</b> property and select <b>internal</b> .	
4	Click on the <b>OK</b> button. The Transitions display in the same compartment as internal activities ( <i>exit</i> , <i>do</i> , <i>entry</i> ).  	

#### Notes

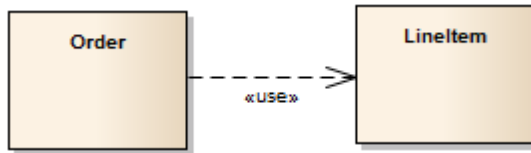
- To view or edit the properties of the internal Transition, double-click on the entry in the compartment within the State
- If you need multiple internal transitions, including those with the same Trigger but different guards, you create them separately with each Transition having its own guard

#### OMG UML Specification

The OMG UML specification (*UML Superstructure Specification, v2.4.1, p. 583*) states:

*An internal transition executes without exiting or re-entering the state in which it is defined. This is true even if the state machine is in a nested state within this state.*

### 6.5.35 Usage



#### Description

A Usage is a Class diagram relationship in which one element requires another element for its full implementation or operation. The diagram above shows that the Class *Order* requires the Class *LineItem* for its full implementation.

The Usage relationship is a subtype of a Dependency relationship.

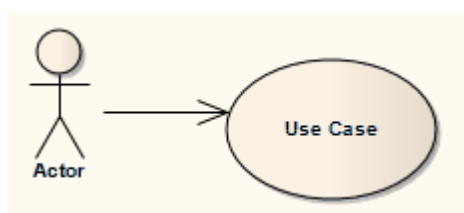
#### Toolbox icon



#### Learn more

- [Class Diagram](#) 1184
- [Dependency](#) 1404

### 6.5.36 Use



#### Description

A Use relationship indicates that one element requires another to perform some interaction. The Use relationship does not specify how the target supplier is used, other than that the source client uses it in definition or implementation.

You typically use the Use relationship in Use Case diagrams to model how Actors use system functionality (Use Cases).

#### Notes

- It is more usual (and correct UML) to have an Association between an Actor and a Use Case

- The **Usage** relationship, used in Class diagrams, is a different relationship

#### Toolbox icon



#### Learn more

- [Use Case Diagram](#) <sup>[1201]</sup>
- [Association](#) <sup>[1393]</sup>
- [Usage](#) <sup>[1450]</sup>

## 6.6 UML Stereotypes

UML supports *stereotypes*, which are an inbuilt mechanism for logically extending or altering the meaning, display, characteristics or syntax of a basic UML model elements. You can apply stereotypes to a range of model element types, including:

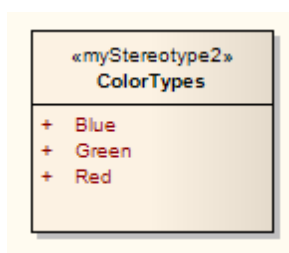
- Elements (such as Classes and Objects)
- Relationships (such as Dependencies and Associations)
- Association Ends
- Attributes and Operations
- Operation Parameters

Different model elements have different stereotypes associated with them. You can create and use your own stereotypes in three different ways:

- To create a new object **type** based on a basic UML model element type, to be imported as part of a Profile into any model and made available for use through the Diagram Toolbox; examples of extended element types already provided in Enterprise Architect include a Table element (which is a stereotyped Class element) and Boundary, Control and Entity elements (which are are stereotyped Object elements)
- To customize the appearance or property of an **instance** of a model element of a specific type; these stereotypes are applied only through the Properties dialog of the object, within the model in which they are created, although you can transport custom stereotype definitions between models as Reference Data
- As a simple label on an element, to identify the role or nature of the object that the element represents

For further definitions of stereotypes, see the OMG UML specification (*UML Superstructure Specification*, v2.1.1, section 18.3.8, pp. 667-672).

Where a stereotype does not affect appearance, it is generally indicated by name on the base UML object shape. In the example below, «myStereotype2» is the stereotype name. Some of the built-in stereotypes are also represented by icons; see *Stereotype Visibility*.



Where the stereotype causes the element to be drawn differently, or is used to define a new type of object, the element shape can be quite different, as illustrated by the three Robustness diagram stereotypes:



You apply a new appearance or shape by associating the stereotype with either a metafile (image file) and fill, border and text colors, or a Shape Script that defines the shape, dimensions and text of the object.

#### Learn more

- [Apply Stereotypes](#) <sup>[1453]</sup>
- [Stereotype Selector](#) <sup>[1455]</sup>
- [Stereotype Visibility](#) <sup>[1456]</sup>
- [Standard Stereotypes](#) <sup>[1457]</sup>
- [Custom Stereotypes](#) <sup>[1461]</sup>
- [Stereotypes with Alternative Images](#) <sup>[1459]</sup>
- [MDG Technology SDK](#) <sup>[1483]</sup>
- [Add Stereotypes and Metaclasses](#) <sup>[1488]</sup> (in an MDG Technology Profile)
- [Shape Scripts](#) <sup>[1582]</sup>

### 6.6.1 Apply Stereotypes

During the course of your modeling, you may decide that an existing object needs a stereotype applied. Enterprise Architect allows new stereotypes to be applied to objects by themselves or in combination with other stereotypes. You will do this through the **Stereotype** field on:

- The object's Properties **dialog** or
- For an element, the element Properties **window**

**Access** **double-click on object > General**  
**click on element | Element | Properties Window (Alt+1)**

#### Apply stereotype to object, using Stereotype field on Properties dialog

Use any *one* of these steps:

Step	Action	See also
1	Type the stereotype(s) to apply as a comma-separated list.  The system checks the loaded and enabled Profiles and Technologies, and the Stereotypes table, for a stereotype with a matching name and base type. If it finds a	

Step	Action	See also
	<p>match, it links to that stereotype and applies the effects. If it doesn't find a match, it creates a new stereotype name in the Stereotypes table and links to that - the stereotype acting as a simple label.</p> <p>Therefore:</p> <ul style="list-style-type: none"> <li>• If you don't want the element's stereotype to match with an identically named stereotype located by the system, either: <ul style="list-style-type: none"> <li>• Disable the Technology that owns the stereotype</li> <li>• Unload the Profile that owns the stereotype, or</li> <li>• Delete the stereotype from the stereotypes table</li> </ul> </li> <li>• If you do want the stereotype to match to a specific stereotype in a Profile or Technology, load the Profile or enable the Technology.</li> </ul>	
2	Click on the drop-down arrow and select the required (single) stereotype from the list.	
3	Click on the ( ... ) button at the right of the field, and use the Stereotype Selector dialog.	<a href="#">Stereotype Selector</a> <small>[1455]</small>

#### Apply stereotype to element using Stereotype field on Properties window

Stereotype

Use any *one* of these steps:

Step	Action	See also
1	<p>Type the stereotype(s) to apply as a comma-separated list.</p> <p>The system checks the loaded and enabled Profiles and Technologies, and the Stereotypes table, for a stereotype with a matching name and base type. If it finds a match, it links to that stereotype and applies the effects. If it doesn't find a match, it creates a new stereotype name in the Stereotypes table and links to that - the stereotype acting as a simple label.</p> <p>Therefore:</p> <ul style="list-style-type: none"> <li>• If you don't want the element's stereotype to match with an identically named stereotype located by the system, either: <ul style="list-style-type: none"> <li>• Disable the Technology that owns the stereotype</li> <li>• Unload the Profile that owns the stereotype, or</li> <li>• Delete the stereotype from the stereotypes table</li> </ul> </li> <li>• If you do want the stereotype to match to a specific stereotype in a Profile or Technology, load the Profile or enable the Technology.</li> </ul>	

Step	Action	See also
2	Click on the drop-down arrow and select the required stereotype from the list.	
3	Select the <b>browse other stereotypes...</b> option at the end of the drop-down list to use the Stereotype Selector dialog.	<a href="#">Stereotype Selector</a> <sup>[1455]</sup>

#### Learn more

- [Properties Dialog](#)<sup>[956]</sup>
- [Properties Window](#)<sup>[992]</sup>
- [Custom Stereotypes](#)<sup>[1461]</sup>

## 6.6.2 Stereotype Selector

If you want to apply more than one stereotype to a UML object, from **multiple sources** such as Profiles or the Custom Stereotypes list, you can select the stereotypes from the Stereotype Selector dialog. This dialog also helps you to identify existing, valid **individual** stereotypes, and to create new stereotypes. The new stereotypes, at this point, are simple labels; if you want them to impose an effect on the object, locate them on the Stereotypes tab of the UML Types dialog and define the effect.

**Access**    **Right click object | Properties > General: Stereotype ( ... )**

#### Select stereotypes to apply or remove

Field/ Button	Action
<b>Profile</b>	Click on the drop-down arrow and choose the required stereotype source - an integrated MDG Technology or the base EAUML, or your Customized Stereotypes list (select the blank entry).
<b>Stereotypes</b>	Select the checkbox against each required stereotype. If you no longer want to use a stereotype, deselect the checkbox.
<b>New</b>	Click on this button to create a new (but undefined) stereotype. A prompt displays for the stereotype name.
<b>OK</b>	Click on this button to apply the selection.

#### Notes

- If you have selected more than one stereotype, the **Stereotypes** field of the Properties dialog only shows one of them; the full list **is** shown on the object in the diagram
- The appearance of a stereotype on an object in a diagram is influenced by the stereotype visibility settings on the Diagram Properties dialog

#### Learn more

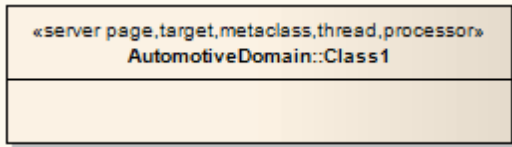

- [Custom Stereotypes](#)<sup>[1461]</sup>
- [Apply Stereotypes](#)<sup>[1453]</sup>
- [Stereotype Visibility](#)<sup>[1456]</sup>

### 6.6.3 Stereotype Visibility

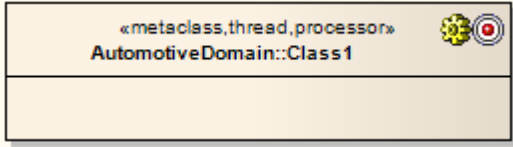
When you apply one or more stereotypes to an object, the display of that object in a **diagram** defaults to showing the stereotype names in a string within guillemets (« »); multiple names are separated by commas. Some stereotypes are associated with small icons that display in the top right corner of the element; these icons are built into the system, and cannot be deleted or added to. In both cases, you can modify the visibility of the text or icon stereotype indicators in a diagram, using the Diagram Properties dialog.

**Access** **double-click diagram background > Elements**  
**double-click diagram background > Features**

#### Set Stereotype Visibility Options

Field/Button	Action
<b>Show Element Stereotypes</b>	<p>Select this checkbox on the Elements tab to show all <b>element</b> stereotypes in the current diagram; for example (with <b>Use Stereotype icons</b> (below) <b>not</b> selected):</p>  <p>Deselect this checkbox to hide all element stereotype names (and icons).</p> 
<b>Use Stereotype Icons</b>	<p>Select this checkbox on the Elements tab to display icons instead of text, for those element stereotypes that have icons defined.</p> <p>Stereotypes that do not have associated icons are still represented by the stereotype names; for example.</p>



Field/Button	Action
	 <p>The icons represent the stereotypes «server page» and «target».</p>
<b>Show Stereotypes</b>	Select this checkbox on the Features tab to show all <b>attribute and operation</b> stereotypes in the current diagram. This option does not affect the display of element stereotypes.

**Notes:**

- In the **Project Browser**, the object name is preceded by the stereotype name(s) within guillemets, and multiple names are indicated by the first stereotype name followed by an ellipsis (...); you can hide the stereotype name by deselecting the Project Browser **Show Stereotypes** checkbox (**Tools | Options | General**)

**Learn more**

- [UML Stereotypes](#) <sup>1452</sup>

**6.6.4 Standard Stereotypes**

Below is a list of standard element stereotypes (as provided in the *EABase.eap* base model), each enclosed by guillemets (« »):

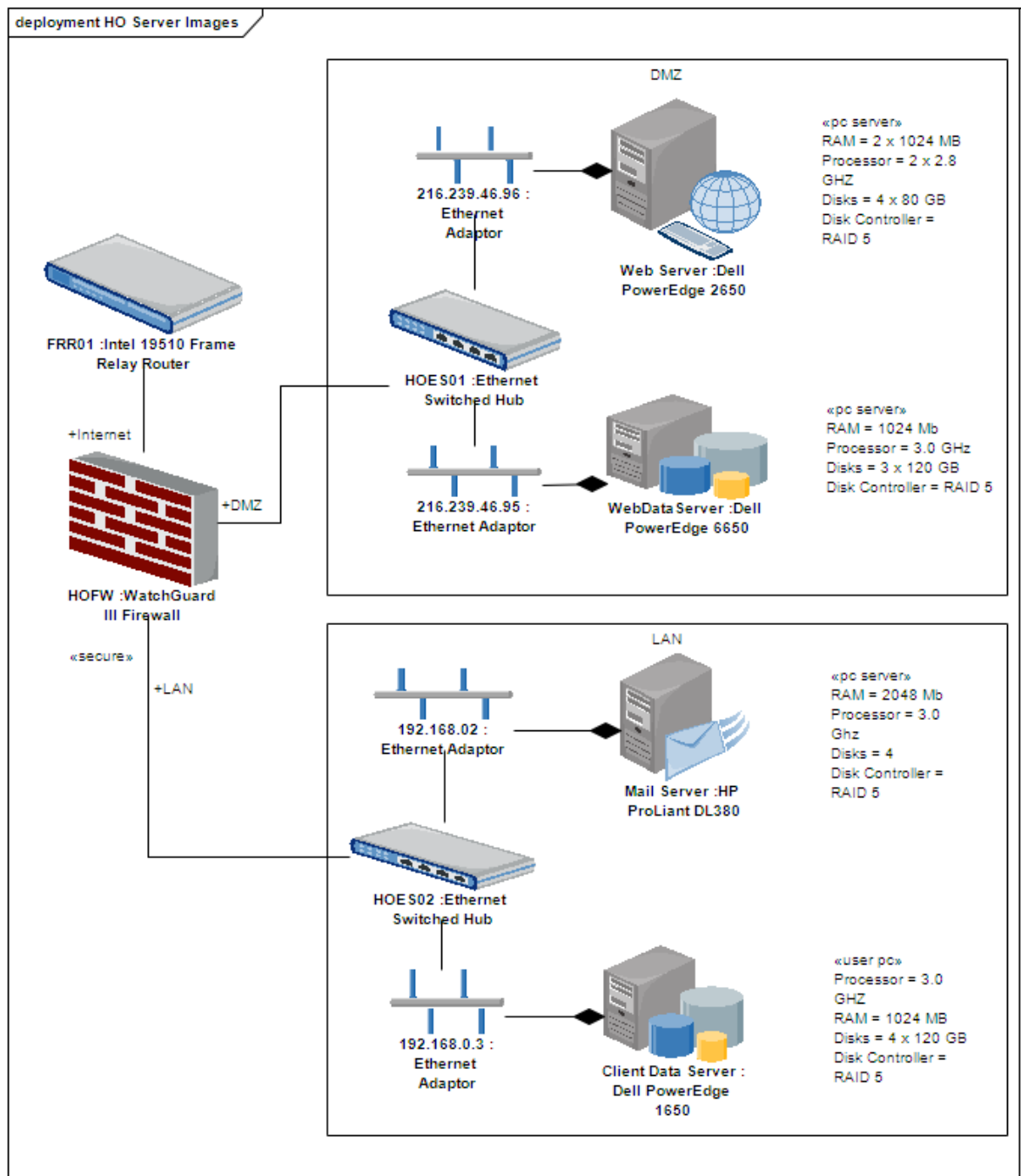
Stereotype	Base Class
«access»	Dependency
«become»	Flow
«call»	Usage
«copy»	Flow
«create»	Message
«derive»	Abstraction
«destroy»	Message

Stereotype	Base Class
«document»	Abstraction
«executable»	Abstraction
«facade»	Package
«file»	Abstraction
«framework»	Package
«friend»	Dependency
«global»	AssociationEnd
«implementation»	Class
«implementation»	Generalization
«import»	Dependency
«instantiate»	Usage
«invariant»	Constraint
«library»	Abstraction
«local»	AssociationEnd
«metaclass»	Class
«parameter»	AssociationEnd
«postcondition»	Constraint

Stereotype	Base Class
«powertype»	Class
«precondition»	Constraint
«process»	Classifier
«refine»	Abstraction
«requirement»	Comment
«responsibility»	Comment
«self»	AssociationEnd
«send»	Usage
«stub»	Package
«table»	Abstraction
«thread»	Classifier
«trace»	Abstraction
«type»	Class
«utility»	Classifier

### 6.6.5 Stereotypes with Alternative Images

If you want to represent an element using an image (for example, depict a hardware component using a 3-D box, or even using an image of the unit itself), you can do so using a stereotype that has been associated with a metafile. When the stereotype is applied to a Class or other element that supports alternative graphical format, the element is drawn using the image instead of the standard UML shape. For example, in this Deployment diagram, the Component elements all have alternative images.



### Notes

- You cannot change the representation of elements that include Lifelines, such as those in Sequence diagrams; the standard representation is important in the use and function of those elements

### Learn more

- [Custom Stereotypes](#) <sup>[1461]</sup>

### 6.6.6 Custom Stereotypes

A custom **Stereotype** applies a different appearance or characteristic to a basic UML model component or feature. You can apply a custom stereotype in two different ways:


- To change the appearance or property of **an instance** of a model component of a specific type; these stereotypes are defined on the Stereotypes tab of the UML Types dialog and applied through the Properties dialog of the object, within the model in which they are created, although you can transport custom stereotype definitions between models as Reference Data
- As a simple label on an element, to identify the role or nature of the object that an element represents; these stereotypes are simply names typed into the **Stereotype** field of the object Properties dialog, and do not affect the element display unless they are subsequently edited to have an effect

The more obvious changes you can make are to the shape, dimensions and appearance of the object, which you can apply by associating a metafile (image file) and customized colors with the stereotype, or by attaching a Shape Script to the stereotype. When you have defined and saved the stereotype, you can then apply it to any new or existing object of the base class with which it is associated.

**Access**   **Settings | UML Types > Stereotypes**

#### Maintain custom stereotypes

Field/Button	Usage	See also
<b>Stereotype</b>	Type or select the name of the stereotype.	
<b>Group name</b>	(Optional) Type a plural name under which to group the stereotype features for attributes and operations; the name will be shown on diagrams in the attributes and operations compartments.	
<b>Base Class</b>	Click on the drop-down arrow and select the name of a pre-existing object type so that the stereotyped element will inherit the base characteristics of that type.	
<b>Notes</b>	(Optional, but recommended) Type any notes concerning the stereotype (not the elements to which the stereotype is to be applied).	
<b>New</b>	Click on this button to clear the above fields to create a new stereotype definition.	
<b>Save</b>	Click on this button to save a new or edited stereotype definition.	
<b>Delete</b>	Click on this button to delete a stereotype definition from the model.	
<b>Override Appearance</b>		

Field/Button	Usage	See also
<b>None</b>	Select to retain the default element appearance for this stereotype.	
<b>Metafile</b>	Select to associate the stereotype with an image metafile (.emf or .wmf) to apply that image when the stereotype is used.	
<b>Shape Script</b>	Select to associate the stereotype with a custom shape, created using the Shape Scripting language.	<a href="#">Shape Scripts</a> 
<b>Assign</b>	Click on this button to either: <ul style="list-style-type: none"> <li>• Display the browser to locate the .emf or .wmf metafile to associate with the stereotype, or</li> <li>• Open the Shape Editor create the Shape Script to be associated with the stereotype</li> </ul>	
<b>Edit</b>	If a Shape Script is already associated with the stereotype, click on this button to open the Shape Editor to update the Shape Script.	
<b>Remove</b>	Remove the associated metafile or Shape Script from the stereotype.	
<b>Default Colors</b>		
<b>Fill</b>	Click on the drop-down arrow and select or define the default background color of the elements to be refined by the stereotype.  This color will be applied to all occurrences of any element to which the stereotype has been applied; if the color is subsequently changed, the change is immediately applied to all occurrences of any element to which the stereotype was applied (as for changes to any other property of the stereotype).  However, on elements created with the stereotype, the default color might be overridden by other color definitions of a higher priority that have been applied to the element.	
<b>Border</b>	Click on the drop-down arrow and select or define the default color of the borders of the elements to be refined by the stereotype.	
<b>Font</b>	Click on the drop-down arrow and select or define the default color of the text of the elements to be refined by the stereotype.	
<b>Reset</b>	Reset the default colors to those of the base element with which the stereotype is associated.	

Field/Button	Usage	See also

### Notes

- You can transport custom stereotype definitions between models, using the **Export Reference Data** and **Import Reference Data** options
- You can also create Stereotype **elements** that extend basic UML model element **types** to create new model element types; you can re-use these extended model elements in other projects, by incorporating them into a Profile (usually within an MDG Technology) and importing this into the various target projects

### Learn more

- [UML Stereotypes](#) <sup>1452</sup>
- [Apply Stereotypes](#) <sup>1453</sup>
- [Developing Profiles](#) <sup>1485</sup>
- [Export Reference Data](#) <sup>376</sup>
- [Import Reference Data](#) <sup>380</sup>

## 6.7 Design Patterns

**Design patterns** are a group of collaborating Objects/Classes that can be abstracted from a general set of modeling scenarios. They are also known as **parameterized collaborations**.

Design patterns are an excellent means of achieving re-use and building in robustness. As patterns are discovered in any new project, the basic pattern template from previous engagements can be re-used with the appropriate variable names modified for the current project.

Patterns generally describe how to solve an abstract problem, and it is the task of the pattern user to modify the pattern elements to meet the demands of the current engagement.

Before using a pattern it must first be created as a standard UML diagram and then saved as an XML pattern file. This XML file can then be imported as a resource that can be used in any model.

### Sparx-Created GoF Patterns

To help you start using design patterns in Enterprise Architect, Sparx Systems provides you with an MDG technology for the patterns described in the book *Design Patterns - Elements of Reusable Object-Oriented Software* by Gamma et al., referred to as the 'Gang of Four' or GoF. These patterns are made available through a set of Toolbox pages.

### Notes

- You can transport all Patterns listed in the Resources window between projects, using the **Export Reference Data** and **Import Reference Data** options

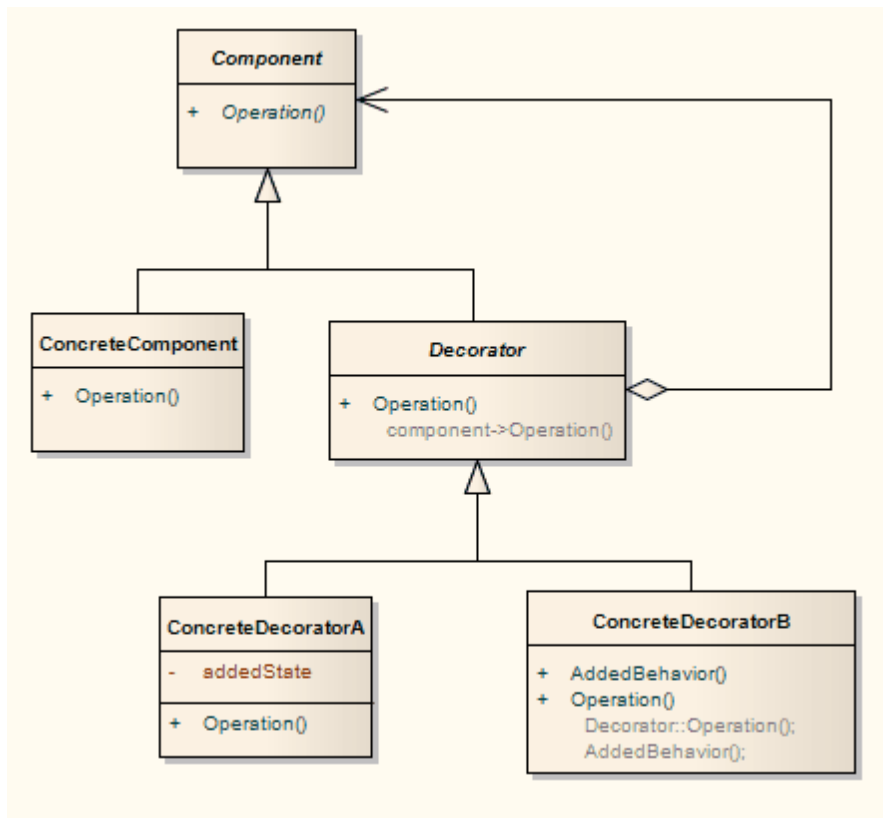
### Learn more

- [GoF Patterns](#) <sup>[2285]</sup>
- [Create a Pattern](#) <sup>[1464]</sup>
- [Import a Pattern](#) <sup>[1466]</sup>
- [Use a Pattern](#) <sup>[1467]</sup>
- [Export Reference Data](#) <sup>[376]</sup>
- [Import Reference Data](#) <sup>[380]</sup>

### 6.7.1 Create a Pattern

To create a Design Pattern you first must model the Pattern as a standard UML diagram within Enterprise Architect. The following diagram was created from an example in the GoF book *Design Patterns - Elements of Reusable Object-Oriented Software* by Gamma et al.





Access [Select required diagram](#) | [Diagram](#) | [Advanced](#) | [Save UML Pattern](#)

### Reference

On the Save Diagram as UML Pattern dialog, complete the fields as indicated below:

Field/Button	Action	See also
<b>Pattern Name</b>	Type the Pattern name.	
<b>Filename</b>	Type a directory path and .XML filename into which to save the Pattern.	
<b>Category</b>	Type the Category under which the Pattern should be listed in <b>UML Patterns</b> (required).	
<b>Version</b>	Type the Pattern version number.	
Notes	Type any notes on the Pattern.	
<b>Actions</b>	Select the appropriate checkboxes to select the actions for the	<a href="#">Use a Pattern</a> <sup>1467</sup>

Field/Button	Action	See also
	<p>elements that are contained in the Pattern; these actions are performed when the Pattern is used.</p> <p>The available actions are:</p> <ul style="list-style-type: none"> <li>• <b>Create</b>: Creates the Pattern element directly without modification</li> <li>• <b>Merge</b>: Merges the Pattern element with an existing element, enabling the existing element to take on the role of the selected Pattern element</li> <li>• <b>Instance</b>: Creates the Pattern element as an instance of an existing element</li> <li>• <b>Type</b>: Creates the Pattern element types as an existing element</li> </ul> <p>If your Pattern includes an Object element, you would use <b>Instance</b> to set the classifier of the Object to one of the Classes in the diagram onto which you are dropping the Pattern.</p> <p>If your Pattern includes a Property (Port or Part) you would use <b>Type</b> to set the type of the Property to one of the Classes in the diagram onto which you are dropping the Pattern.</p>	
<b>OK</b>	<p>Click on this button twice to save the Pattern.</p> <p>Once saved you can load the Pattern into Enterprise Architect, into the Resources window.</p>	<a href="#">Import a Pattern</a> <small>[1466]</small>

### Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Diagrams** permission to save a diagram as a Pattern
- If your source diagram contains information flows, the Information Items Conveyed and Information Flows Realized data is not copied into the Pattern
- To change the name of one of the elements, double-click on the element to display the Edit dialog; from this dialog you can also add comments detailing the element's purpose
- Patterns can not be created for Sequence diagrams.

### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Modeling Languages | Patterns | Create a Pattern**

## 6.7.2 Import a Pattern

Before using a previously created Design Pattern file in a model, you must first import it into the model; it is then available from the Resources window and optionally from the Toolbox.

**Access**   **Project | Resources (Alt+6) | Right-click on UML Patterns | Import UML Pattern**

**Import a Design Pattern**

Step	Action
1	On the Select UML Pattern Import Filename dialog, locate and click on the XML file to import.
2	Click on the <b>Open</b> button to import the Pattern.  The imported Pattern is placed in the appropriate category as defined in the XML file; if the category does not already exist under <i>UML Patterns</i> , a new one is created.

**Patterns in MDG Technologies**

A number of Technologies provide their own Patterns, and some Technologies are designed principally as a vehicle for making specific Patterns available to the model, such as the MDG Technology for Gang of Four Patterns. Such Patterns are provided through the Diagram Toolbox pages for the Technology. If you want to use such Patterns, check that the appropriate Technology has been loaded and enabled in the model.

**Learn more**

- [Create a Pattern](#) <sup>[1464]</sup>
- [Using MDG Technologies](#) <sup>[1475]</sup>

**Learning Center topics**

- (Alt+F1) | **Enterprise Architect** | **Modeling Languages** | **Patterns** | **Import a Pattern**


**6.7.3 Use a Pattern**

Using a Design Pattern, you can rapidly create template solutions for code structures that perform the same type of task in other situations, and use items defined in the Pattern with the UML model.


**Access**   **Project** | **Resources** (Alt+6) | **UML Patterns** | **Right-click on Pattern**

**Use a Pattern previously imported into the model**

Step	Action	See also
1	Open the diagram into which to add the Pattern.	
2	Select the Resources window.	

Step	Action	See also
3	Expand the <i>UML Pattern</i> folder and find the Pattern to add.	
4	<p>Either:</p> <ul style="list-style-type: none"> <li>• Select the <b>Add Pattern to Diagram</b> context menu option or</li> <li>• Drag and drop the Pattern from the Resources window onto the diagram</li> </ul> <p>You can also view the Pattern details in read-only mode by selecting the <b>View Pattern Details</b> context menu option.</p>	<a href="#">Add Pattern Dialog</a> 
5	Once the appropriate selections have been made, click on the <b>OK</b> button to import the Pattern into the model, recreating the original diagram with new GUIDs.	

#### Change the default of the Pattern element

Step	Action	See also
1	From the Add Pattern dialog select the individual element in the Pattern Element panel.	
2	<p>Click on the ( ... ) button to display the Edit dialog.</p> <p>The specific method for changing the element name is dependant upon the entry in the <b>Action</b> column of the Pattern Elements panel.</p>	
3	<p>If the <b>Action</b> entry for the element is <b>Create</b>, then in the <b>Default</b> field in the Edit dialog delete the existing value and type your own, user-defined value.</p> <p>Click on the <b>OK</b> button.</p> <p>The element default is updated on the Add Pattern dialog.</p>	
4	<p>If the <b>Action</b> entry for the element is <b>Merge</b>, in the Edit dialog click on the ( ... ) button to browse to an existing element classifier.</p> <p>The Select &lt;Item&gt; dialog displays.</p>	
5	<p>Locate and select an existing element classifier.</p> <p>You can restrict the number of choices by selecting the elements from a specific namespace; to do this, click on the <b>In Namespace</b> drop-down arrow and select a namespace.</p>	<a href="#">Using Classifiers</a> 

Step	Action	See also

#### Learn more

- [Import a Pattern](#) 

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect** | **Modeling Languages** | **Patterns** | **Use a Pattern**

### 6.7.4 Add Pattern Dialog

The Add Pattern dialog displays when you are using or editing a Design Pattern element.

#### Reference

Panel	Action
<b>Preview</b>	Display a preview of the Pattern.  Click on the <b>Preview</b> link to open a view of the Pattern and drag the sides into as large a picture as you require.
<b>Pattern Elements</b>	Access the individual elements contained in the Pattern. From here you can: <ul style="list-style-type: none"><li>• Select the action for the individual element (<i>Create</i>, <i>Merge</i>, <i>Instance</i> or <i>Type</i>, as applicable for each element) by clicking on the drop-down arrow, or</li><li>• Modify the default of the Pattern element or - for a merged element - choose the namespace, by clicking on the ( ... ) button on the right of the <b>Default</b> entry</li></ul>
<b>Element Notes</b>	Display the comments that describe the element in the Pattern.  Highlight an element in the Pattern Elements panel to view the notes.

**Part**

---

**VII**









## 7 Extending UML Models

Sometimes a modeling problem cannot be adequately expressed using the base UML model elements or, similarly, an area of work falls into a specialized domain that requires a tailored modeling approach and/or program language support. To meet such requirements, you can **extend** the capabilities of UML to develop new modeling constructs, using **MDG Technologies** to combine and deploy a wide range of **extension mechanisms** such as:

- UML Profiles
- Stereotypes
- Shape Scripts
- Tagged Values
- Constraints
- Patterns
- Customized Code and Transformation Templates, and
- Grammars

Using the **MDG Technology Creation Wizard**, you can quickly and easily integrate the extensions into a technology and rapidly tailor UML and Enterprise Architect to address a particular modeling domain not explicitly covered in the original UML specification, but using extension mechanisms that are still part of the Specification.

### Learn more

- [Using MDG Technologies](#)  <sup>1475</sup>
- [Using UML Profiles](#)  <sup>1472</sup>
- [Tagged Value Types](#)  <sup>1621</sup>
- [Shape Scripts](#)  <sup>1582</sup>
- [Developing Programming Languages](#)  <sup>1581</sup>
- [Code Template Framework](#)  <sup>1631</sup>
- [Grammar Framework](#)  <sup>1705</sup>
- [Creating an MDG Technology](#)  <sup>1527</sup>

## 7.1 Using UML Profiles

A **UML Profile** is a collection of **extensions** (**stereotypes** and their associated **Tagged Values**) to basic UML model elements, which together describe some particular modeling problem and which you can create and use to develop modeling constructs in a specific domain. The stereotypes and Tagged Values of a Profile are applied to elements, attributes, methods, method parameters, connectors and connector ends to create new, tailored model elements. A Profile is distributed and used as a component of an MDG Technology.

The deployed technology automatically generates a page of elements and relationships in the Diagram Toolbox, for each of the UML Profiles within the technology. When you drag the elements and connectors from the Toolbox onto the current diagram, the stereotype, Tagged Values and default values, notes and metafile (if one is specified) are automatically applied to the new element. You can also drag and drop Profile attributes and operations onto **existing** Classes, so that they are immediately added with the specified stereotype and Tagged Values.

### Learn more

- [MDG Technologies](#) <sup>[1475]</sup>
- [Add Profile Objects to a Diagram](#) <sup>[1472]</sup>
- [Tagged Values in Profiles](#) <sup>[1473]</sup>
- [Synchronize Tagged Values and Constraints](#) <sup>[1473]</sup>
- [MDG Technology SDK](#) <sup>[1483]</sup>

### 7.1.1 Add Profile Objects to a Diagram

After a technology has been imported into your project, the profiled objects (elements and connectors) and features (attributes and operations) are available from the technology pages of the Diagram Toolbox. The way in which you add the Profile objects to a diagram is no different from the way in which you use the standard UML objects on the system.

**Access** **Diagram | Diagram Toolbox (Alt+5): More tools | <technology name>**

### Use the Profile Objects

Action	Description
<b>Add a Profile-based element to a diagram</b>	Click on the element in the Toolbox page and drag it onto the diagram.
<b>Add a Profile-based connector to a diagram</b>	Click on the connector in the Toolbox page, then click on the source element in the diagram and drag it to the target.
<b>Add a Profile-based attribute or operation to a diagram</b>	Click on the attribute or operation in the Toolbox page, and drag it onto the host element on the diagram.  The system prompts you to enter a name for the feature.

### Learn more



- [Diagram Toolbox](#)<sup>[792]</sup>
- [MDG Technologies](#)<sup>[1475]</sup>

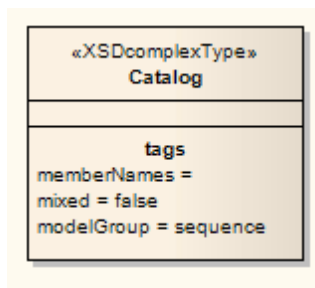
### 7.1.2 Tagged Values in Profiles

Stereotypes within a profiled element or connector can define one or more associated **Tagged Values**. When you drag a profiled element or connector from the Diagram Toolbox onto a diagram, any associated Tagged Values are automatically added to the new element or connector. Tagged Values in profiled objects are an excellent way to further extend the versatility of your UML modeling.

As an example, the UML Profile for XSD (XML Schema) provides the *XSDComplexType* stereotype to extend a Class; this stereotype has the Tagged Values:

- **memberNames**
- **mixed** and
- **modelGroup**

When you create a Complex Type element, the Tagged Values are added and are visible in the *tags* compartment of the element (including those that have no value set).



When you select the element, the Tagged Values **window** displays all the associated tags.

- The values of tags imported in a Profile **override** the values of equivalent tags in the UML Types dialog; if the initial value of the tag from the Profile is **not set**, the value of the tag shown in the element will be **blank**, even if there are default values for the tag in the UML Types dialog
- Tags that have default profile values are automatically set
- Where Tagged Values in the profiled element have a *values* section (for example, *values="element | attribute | both" default="both"*) you can select the non-default values from a drop-down list
- Where no *value* exists, you can add a value as free text; you would do this for a profile tag that has no initial value, to use a default value from the UML Types dialog

#### Learn more

- [Predefined Structured Types](#)<sup>[1622]</sup>
- [Tagged Values](#)<sup>[1134]</sup>

### 7.1.3 Synchronize Tagged Values and Constraints

When you create an element, attribute, operation or connector from a profiled object, the Tagged Values and constraints are added from the Profile stereotype. Subsequently, you might update the constraints or Tagged Values of a particular stereotype in the Profile, in which case the items already created in the model would not have those additional constraints or Tagged Value tags and notes.

Similarly, you might have manually added the stereotype to a set of objects, which automatically adds the Tagged Values but not the constraints associated with that stereotype, and now want the objects to receive

the constraints.

You can apply the updated or missing Tagged Values and constraints using the **Synchronize Stereotype** function. This operates on any profiled element in your model, from any technology that is integrated with or imported into Enterprise Architect.

**Access** **Diagram | Diagram Toolbox | More tools | <technology name>** then  
**right-click icon for profiled element/connector/feature | Synchronize Stereotype**

#### Synchronize objects using the Technology Toolbox pages

Step	Action
1	On the Synch Profiled Elements dialog, click on the <b>OK</b> button.  All elements, features or connectors created with the selected profiled object icon are updated, across the model.  The items that have been modified, and the changes that were made, are listed in the <b>Actions</b> field.
2	When the update is complete, click on the <b>Cancel</b> button.

#### Alternative - Single Object Update

You can quickly synchronize the tags and constraints of a **single** element **in a diagram**. To do this:

Step	Action
1	Drag the updated profiled element from the Diagram Toolbox page onto the element in the diagram. A short context menu displays.
2	Select the <b>Apply «stereotype name»</b> context menu option.  The diagram element is updated with any tags and constraints from the profiled element that it does not already have.

#### Notes

- The **Synchronize Stereotype** context menu option displays when a Diagram Toolbox icon represents a profiled element or and connector; it does not display for basic UML object icons
- You can review any changes by displaying the element Properties dialog and by opening the Tagged Values window and clicking on an appropriate profiled element
- Removing a stereotype from an object automatically removes any Tagged Values assigned by that stereotype






## 7.2 Using MDG Technologies

An MDG Technology is a vehicle for providing access to the resources of either a commercially-available technology or a technology that you have created yourself. Such resources include a wide range of facilities and tools, such as UML Profiles, code modules, scripts, Patterns, images, Tagged Value Types, report templates, linked document templates, Toolbox pages and Learning Center pages.

Using Enterprise Architect, you can develop models based on the standard UML specifications, and you can extend the core UML structures using UML-supported mechanisms such as Tagged Values, Stereotypes, Profiles and Design Patterns. These facilities are within the Enterprise Architect core technologies, and you can activate and use further Model Driven Generation (MDG) Technologies that are either integrated with the system or available from external locations.

If your systems or work domain require further specialization you, as a Technology Developer, can use Enterprise Architect to develop your own customized modeling languages and solutions.

### Obtain and use Technologies

Source of Technology	See also
<p>Core technologies - Enterprise Architect itself contains a:</p> <ul style="list-style-type: none"> <li>• Basic UML 2 technology as an implementation of UML 2.4.1 structural and behavioral modeling, and</li> <li>• Core Extensions technology that applies profiles and stereotypes to provide extended modeling of aspects such as Requirements, User interface and Data Modeling</li> </ul>	
Additional technologies are included in the Enterprise Architect Install directory, <i>MDGTechnologies</i> subfolder.	<a href="#">Extensions - MDG Technologies</a> 
You can import technologies from external sources into the <i>APPDATA</i> folder (% <i>APPDATA%\Sparx Systems\EA\MDGTechnologies</i> ) for your own use, or into the Resources window for other project users to access.	<a href="#">Import MDG Technologies to Model Resources</a> 
You can transfer technologies into the <i>MDGTechnologies</i> subfolder; these technologies are available when you restart Enterprise Architect (on Vista/Windows 7 systems you might have to increase your access permissions to do this).	
You can access and activate MDG Technologies in remote system folders or web sites, from Enterprise Architect.	<a href="#">Access Remote Technologies</a> 
Technology Developers can create new MDG Technologies and deploy them to the project team either through the <i>MDGTechnologies</i> subfolder or from a remote folder or website.	<a href="#">Creating an MDG Technology</a> 
To see which technologies are available within Enterprise Architect, and activate the ones you require, use the MDG Technologies dialog ( <b>Settings   MDG Technologies</b> ).	<a href="#">Manage MDG Technologies</a> 

Source of Technology	See also
<p>Having made the MDG Technologies available, you can manage their availability to users and you can work with them.</p> <p>You also have the facility to turn <b>off</b> or <b>disable</b> the Enterprise Architect <b>Basic UML 2</b> and <b>Core Extensions</b> technologies and facilities, so that you can apply the Enterprise Architect facilities and features exclusively to one or more selected MDG Technologies.</p>	<p><a href="#">Work with MDG Technologies</a> <sup>[1476]</sup></p>

### Notes

- In developing **new** technologies, you would be familiar with the modeling structures and concepts of the core system and extension mechanisms as they impact and are used by the people you are designing the technology for; that is, the system as described in the modeling sections of this *User Guide*

### Learn more - General

- [Modeling Basics](#) <sup>[750]</sup>
- [Standard UML Models](#) <sup>[1179]</sup>
- [Requirement Models](#) <sup>[1726]</sup>
- [Domain Based Models](#) <sup>[1789]</sup>

### Learn more - Technology Development

- [Extending UML Models](#) <sup>[1471]</sup>

## 7.2.1 Work with MDG Technologies

Any MDG Technology listed on the MDG Technologies dialog can be enabled, which makes their interface profiles and Toolbox pages available for your use.

When you **enable** an MDG Technology, any Technology-specific diagram types are **added** to the New Diagram dialog lists, and the Technology's Toolbox pages are added to those available through the **More tools** menus in the Toolbox.

If you set an MDG Technology to **Active**, it becomes the main technology for the model. Only one Technology can be active at a time. The Technology's validation configuration is set, and whilst common Toolbox pages are visible at all times, the Technology's Toolbox pages **override** any parallel Enterprise Architect Toolbox pages; for example, the ICONIX *Class* pages would override the Enterprise Architect *Class* pages.

You create Technology-specific diagrams and populate them with elements and connectors in the same way as for standard Enterprise Architect diagrams.

### Learn more

- [Create New Diagrams](#) <sup>[822]</sup>
- [Manage MDG Technologies](#) <sup>[1477]</sup>

## 7.2.2 Manage MDG Technologies

You use the MDG Technologies dialog to manage the MDG Technologies accessible to the project and available to project users. The dialog lists the technologies held in a number of locations accessed by the project, such as the **APPDATA** folder and the Enterprise Architect **Install directory**. You can set these technologies to being available for use or disabled, as you require. MDG Technologies are deployed as .XML files.

[Access](#)   **Settings | MDG Technologies**

### Configure availability of Technologies

Field/Option/ Button	Action	See also
<b>Technology</b>	<p>Lists all MDG Technologies currently accessible to the project, in alphabetical order.</p> <p>If you click on a Technology name, the upper right panel of the dialog displays the technology:</p> <ul style="list-style-type: none"> <li>• Name</li> <li>• Version number</li> <li>• Logo (if defined), and</li> <li>• Location of the deployed XML file, which can be: <ul style="list-style-type: none"> <li>• Internal to Enterprise Architect</li> <li>• An extension</li> <li>• In the Install directory (just the file name)</li> <li>• In the <i>APPDATA</i> folder (filename followed by <i>(in APPDATA)</i>)</li> <li>• In the model</li> </ul> </li> </ul> <p>The lower right panel displays a description of the Technology, in many cases providing the manufacturer's web site address and a support contact.</p>	
<b>Enabled</b>	<p>Select this checkbox against each Technology that you want to be available for use in the project. When an MDG Technology is enabled:</p> <ul style="list-style-type: none"> <li>• The Technology is added to the list of available options in the <b>Profile</b> field of the <b>Default Tools</b> toolbar, so that you can apply the interface profiles of the MDG Technology</li> <li>• At least one set of Toolbox pages for the MDG Technology is automatically added to the <b>Diagram Toolbox</b>; you can access the added Toolbox pages through the <b>More Tools</b> option</li> <li>• Any MDG Technology-specific <b>diagram templates</b> are added to the <b>New Diagram</b> dialog for selection; when selected, these display the diagram-specific Toolbox pages</li> </ul>	

Field/Option/ Button	Action	See also
	<p>Clear the checkbox against a Technology to make it unavailable to the project users.</p> <p>If you disable an MDG Technology that was in use, its Toolbox pages, diagram types and quick-links are <b>omitted</b> from the Diagram Toolbox, <b>More tools</b> option, Default Tools toolbar, diagrams and New Diagram dialog in the user interface.</p>	
<b>All</b>	Click on this button to <b>select</b> the <b>Enabled</b> checkbox of every Technology listed on the dialog.	
<b>None</b>	<p>Click on this button to <b>clear</b> the <b>Enabled</b> checkbox of every Technology listed on the dialog.</p> <p>If you click on this button, scroll to the top of the list and select the <b>Basic UML 2 Technology</b> and <b>Core Extensions</b> checkboxes to re-enable the UML and Extended Toolbox pages and diagram types.</p>	
<b>Set Active</b>	<p>Click on this button when you have clicked on one preferred Technology, to make that Technology the default interface to Enterprise Architect. This displays an asterisk against the Technology name in the <b>Technology</b> panel, and selects the Technology in the <b>Profile</b> field of the Default Tools toolbar. If the MDG Technology has not yet been enabled, this button also enables it.</p> <p>Making a Technology the active Technology can change the way Enterprise Architect windows are displayed and override the Toolbox pages with pages specific to the Technology.</p>	
<b>Advanced</b>	Click on this button to add MDG Technologies in folders and websites remote from Enterprise Architect.	<a href="#">Access Remote MDG Technologies</a> [1479]
<b>Remove</b>	<p>(Enabled only for Technologies imported directly into the model.)</p> <p>Click on this button to remove the selected Technology from the list, from the Resources window and from the model.</p>	<a href="#">Resources</a> [1173]
<b>OK</b>	Click on this button to close the dialog, save your changes and put them into effect.	
<b>Cancel</b>	Click on this button to close the dialog and abort the changes you have made.	

**Notes**

- If you change the **Enabled** setting of an MDG Technology, or if you change the list of external paths, click on the **OK** button to reload all enabled technologies; you do not need to restart Enterprise Architect for the changes to take effect
- To work exclusively in a selected MDG Technology, or a small number of Technologies, you can enable just those Technologies (and perhaps set one of them to Active) and then *deselect* the **Basic UML 2 Technology** checkbox (and, if necessary, the **Core Extensions** checkbox)

#### Learn more


- [Import MDG Technologies To Model](#) 

### 7.2.3 Access Remote MDG Technologies

When you are working on your model, you can use MDG Technologies local to your system, or you can access Technologies you have identified in folders and websites remote from the system. You essentially 'bookmark' these remote Technologies for continued use, and then delete the link when you do not want to use them any more.

#### Access **Settings | MDG Technologies: Advanced**

#### Specify the location of a remote MDG Technology

Step	Action	See also
1	On the MDG Technologies - Advanced dialog, click on the <b>Add</b> button. A short context menu displays, offering the options: <ul style="list-style-type: none"> <li>• <b>Add Path</b></li> <li>• <b>Add URL</b></li> </ul>	<a href="#">Manage MDG Technologies</a> 
2	To specify an MDG Technology <i>in a directory folder</i> , select the <b>Add Path</b> option. The Browse for Folder dialog displays. Browse for the MDG Technology folder, click on it, and click on the <b>OK</b> button; go to step 4.	
3	To specify an MDG Technology <i>on a web site</i> , select the <b>Add URL</b> option. The Input dialog displays. In the <b>Enter Value</b> field, type or copy-and-paste the MDG Technology URL and click on the <b>OK</b> button.	
4	The folder path or URL for the MDG Technology displays in the Path panel. The Technology is available	

#### Notes

- To **remove** an MDG Technology listed in the MDG Technologies - Advanced dialog, click on the folder path or URL and click on the **Remove** button; the path or URL is deleted

## 7.2.4 Import MDG Technologies to Model

If you locate or create an MDG Technology that is of use to your project, you can import it into the project either:

- For **your own** use in the model; that is, import the technology into the %APPDATA%\Sparx Systems\EA\MDGTechnologies folder on your workstation, or
- To be available to **all users** of the model, through the Resources window for the model

To import an MDG Technology you must have a suitable MDG Technology XML file. If the MDG Technology includes references to any metafiles, they should be in the same directory as the MDG Technology XML file.

On startup, Enterprise Architect scans both the *APPDATA* folder and the Enterprise Architect Install directory *MDGTechnologies* subfolder for technology files, to make them available through the MDG Technologies dialog and, for model Technologies, the Resources window. Technologies imported to the *APPDATA* folder are indicated by the text *Location: Technology.xml*.

**Access**   **Tools | MDG Technology Import**  
**Project | Resources | Right-click MDG Technologies folder | Import Technology**

### Import a technology

Step	Action
1	<p>On the Copy Technology to Application Data dialog, in the <b>Filename</b> field, type the path and filename of the MDG Technology file to import, or browse for it using the ( ... ) button.</p> <p>When you enter the filename, the MDG Technology name and version display in the <b>Technology</b> and <b>Version</b> fields, and any notes display in the <b>Notes</b> field.</p>
2	<p>Select the appropriate radio button for the type of import you want to perform:</p> <ul style="list-style-type: none"> <li>• <b>Import to Model</b></li> <li>• <b>Import to User</b></li> </ul>
3	<p>Click on the <b>OK</b> button.</p> <ul style="list-style-type: none"> <li>• (If you selected the <b>Import to User</b> option) If the <i>APPDATA</i> folder does not yet exist, Enterprise Architect creates it</li> <li>• If the MDG Technology already exists, Enterprise Architect displays a prompt to overwrite the existing version and import the new one</li> </ul> <p>Once the import to <b>APPDATA</b> is complete, you must restart Enterprise Architect; the MDG Technology is then listed in the MDG Technologies dialog.</p>

### Notes

- This feature is not available in the Desktop edition of Enterprise Architect



- To remove an MDG Technology which has been added to APPDATA, locate the appropriate XML file in the %APPDATA%\Sparx Systems\EA\MDGTechnologies folder and remove it
- Consider the fact that some MDG Technologies can be large and might impose some delays on the workstation as they load each time a user connects to the model
- To remove an MDG Technology from the Resources window and the model, either:
  - Right-click on the Technology name and select the **Remove Technology** menu option, or
  - Click on the Technology name in the Manage MDG Technologies dialog and click on the **Remove** button

#### Learn more

- [Manage MDG Technologies](#) <sup>[1477]</sup>
- [Resources](#) <sup>[1173]</sup>

## 7.2.5 Extensions - MDG Technologies

Enterprise Architect is the core for a range of Model Driven Generation (MDG) extensions to its modeling capabilities, using more specialized, niche frameworks and profiles.

Extensions	See also
A number of technologies are already integrated with the Enterprise Architect installer.	<a href="#">ArchiMate</a> <sup>[1926]</sup> <a href="#">BPEL</a> <sup>[1870]</sup> <a href="#">BPMN</a> <sup>[1845]</sup> <a href="#">Data Flow Diagrams</a> <sup>[1797]</sup> <a href="#">Eriksson-Penker Extensions</a> <sup>[1929]</sup> <a href="#">ICONIX</a> <sup>[2282]</sup> <a href="#">Mind Mapping</a> <sup>[1794]</sup> <a href="#">SoaML</a> <sup>[2449]</sup> <a href="#">SOMF 2.1</a> <sup>[2454]</sup> <a href="#">Strategic Modeling</a> <a href="#">Systems Modeling Language (SysML)</a> <sup>[2294]</sup>
Enterprise Architect provides support for: <ul style="list-style-type: none"> <li>• Downloading MDG Technologies from external system files or websites, or</li> <li>• Creating your own easily with the Enterprise Architect MDG Technology Wizard</li> </ul>	<a href="#">Access Remote MDG Technologies</a> <sup>[1479]</sup> <a href="#">MDG Technology SDK</a> <sup>[1483]</sup>
Sparx Systems also market a number of MDG products, as follows: <ul style="list-style-type: none"> <li>• MDG Technology For:                             <ul style="list-style-type: none"> <li>• Zachman Framework</li> <li>• The Open Group Architecture Framework (TOGAF)</li> <li>• Unified Profile for DoDAF and MODAF (UPDM)</li> </ul> </li> </ul>	

Extensions	See also
<ul style="list-style-type: none"> <li>• Data Distribution Service (DDS)</li> <li>• Python (for Enterprise Architect versions 4.5 to 5.0, integrated in later versions) (* free product! *)</li> <li>• CORBA (* free product! *)</li> <li>• Java Beans (* free product! *)</li> <li>• Testing (* free product! *)</li> <li>• MDG Integration For: <ul style="list-style-type: none"> <li>• Eclipse 3.3</li> <li>• Visual Studio 2005, 2008 &amp; 2012</li> </ul> </li> <li>• MDG Link For <ul style="list-style-type: none"> <li>• Eclipse</li> <li>• Visual Studio.NET</li> <li>• Microsoft Visio (* free product! *)</li> <li>• Telelogic DOORS</li> </ul> </li> </ul> <p>Over time, this list is being extended to include further products.</p>	
<p>Sparx Systems provide different editions of Enterprise Architect tailored for systems engineering and business engineering, or both together.</p> <p>These editions incorporate several of the above MDG Technologies and other Add-Ins.</p>	<a href="#">Enterprise Architect Editions</a> <small>[18]</small>
<p><b>Product Information:</b> For the latest list of available Add-Ins and an introduction to each product, including details of pricing, purchasing and download options, see the Sparx Systems website.</p> <p>When you purchase one of the Add-Ins, you receive one or more license keys and instructions on obtaining, installing and registering the product.</p> <p>The information page for most products provides a link to download the product User Guide in .pdf format.</p> <p>The product User Guide can also be displayed as a .chm file online within the product itself; to access this online help in Enterprise Architect, select the <b>Extensions   &lt;productname&gt;   Help</b> menu option.</p>	<a href="#">Products Page on the Sparx Systems website</a>

#### Learn more

- [Creating MDG Technologies](#) [1545]



Extension Facilities

Facility	Description	See also
<b>MDG Technologies</b>	An MDG Technology is a vehicle for providing access to the resources of a commercially-available technology or one that you have created yourself. Such resources include a wide range of facilities and tools, such as UML Profiles, code modules, scripts, Patterns, images, Tagged Value Types, report templates, linked document templates, Toolbox pages and Learning Center pages.	<a href="#">MDG Technologies - Creating</a> <sup>[152]</sup>
<b>Profiles</b>	<p>Profiles are a means of extending UML; you use them to build models in particular domains.</p> <p>A Profile is a collection of additional stereotypes and Tagged Values that extend or are applied to elements, attributes, methods and connectors, which together describe some particular modeling problem and facilitate modeling constructs in that domain.</p>	<a href="#">Using UML Profiles</a> <sup>[1472]</sup>  <a href="#">Developing Profiles</a> <sup>[1485]</sup>
<b>Stereotypes</b>	<p>Stereotypes are an inbuilt mechanism for logically extending or altering the meaning, display and syntax of a model element. Different model elements have different standard stereotypes associated with them.</p> <p>The same principles apply when you customize your own stereotypes, either through the UML Types dialog to qualify an element of an existing type, or as elements that extend a specific metaclass to define a new element type.</p>	<a href="#">UML Stereotypes</a> <sup>[1452]</sup>  <a href="#">Custom Stereotypes</a> <sup>[1461]</sup>  <a href="#">Create UML Profiles</a> <sup>[1485]</sup>
<b>Design Patterns</b>	<p>Patterns are groups of collaborating Objects/Classes that can be abstracted from a general set of modeling scenarios (that is, parameterized collaborations).</p> <p>They generally describe how to solve an abstract problem, and are an excellent means of achieving re-use and building in robustness.</p>	<a href="#">Design Patterns</a> <sup>[1464]</sup>  <a href="#">Create a Pattern</a> <sup>[1464]</sup>
<b>Shape Scripts</b>	<p>A Shape Script is a script that applies a custom shape and orientation to an element or connector, in place of that object's standard UML notation. Each script is associated with a particular stereotype, and is drawn for every object having that stereotype.</p> <p>Where you redefine the properties of a standard UML object to create a new object, you can apply a new shape to the object as well.</p>	<a href="#">Shape Scripts</a> <sup>[1582]</sup>
<b>Tagged Value Types</b>	<p>You use Tagged Values to add further properties to a model element. You can apply them at three levels:</p> <ul style="list-style-type: none"> <li>As a standard Tagged Value associated with the model element</li> <li>As a <b>customized</b> Tagged Value based on a <b>standard</b> Tagged Value Type</li> <li>As a <b>customized</b> Tagged Value based on a <b>customized</b> Tagged Value Type</li> </ul>	<a href="#">Tagged Values</a> <sup>[1134]</sup> <a href="#">Tagged Value Types</a> <sup>[1621]</sup> <a href="#">Create Custom Tagged Value Type</a> <sup>[1626]</sup>

Facility	Description	See also
<b>Code Template Frameworks</b>	Within Enterprise Architect, you can modify the way code is generated or transformed, including generating code for behavioral models, by customizing the templates that control these actions. You can also incorporate these templates in a technology, to add the customized generation and transformation to the facilities of that technology.	<a href="#">Code Template Framework</a> <sup>[1631]</sup>

## 7.3.2 Developing Profiles

Profiles are collections of extensions, based on stereotypes that are applied to UML elements, connectors and features. The stereotypes can have attributes to specifically define Tagged Values that further extend the characteristics of the stereotyped element or connector. Profiles are stored as XML files with a specific format; to apply the extensions of a Profile, you add its XML file as a component of an MDG Technology, and deploy the technology; that is:

1. Create a model in which to develop the MDG Technology, and within this create a Profile package in which you define your Profile(s)
2. Save the Profile as an XML file, with a specific format.
3. Call the XML file into an MDG Technology, using the MDG Technology Creation Wizard.
4. Deploy the MDG Technology (and hence Profile) on your system.

### Learn more

- [MDG Technologies - Creating](#)<sup>[1527]</sup>
- [Create UML Profiles](#)<sup>[1485]</sup>
- [Quick Linker](#)<sup>[1515]</sup>
- [Customize Toolbox Profiles](#)<sup>[1560]</sup>
- [Create Diagram Profiles](#)<sup>[1570]</sup>
- [Tagged Value Types](#)<sup>[1621]</sup>
- [Using UML Profiles](#)<sup>[1472]</sup>
- [UML Profiles in the Resources Window](#)<sup>[1526]</sup>

### 7.3.2.1 Create Stereotype Profiles

When you are creating a Profile to define a new modeling solution, you initially create a package with the «*profile*» stereotype. You then consider the number of model elements (and hence Stereotype elements) you will need to create. If you are going to create:

- A small number of Stereotype elements, you can manage them on a single child diagram within the Profile package, and save the **diagram** as the Profile
- A large number of Stereotype elements, create them on as many child diagrams as are convenient (one Stereotype per diagram if you prefer) and save the **package** as the Profile

Every Stereotype element extends at least one *Metaclass* element. The Stereotype elements use the Profile name as their namespace. When you have created your Profile, you can incorporate it into an MDG Technology.

The process of creating a Profile and applying it to your models comprises a number of steps. Some of

these steps are necessary only if you want the Profile to apply a specific meaning, display, appearance or syntax to a type of model element.



### Create a Profile

Step	Description	See also
1	Create a Profile package in a technology development model.	<a href="#">Create a Profile Package</a> [1487]
2	Add Stereotype and Metaclass elements to the child diagram(s) of the Profile package.	<a href="#">Add Stereotypes and Metaclasses</a> [1488]
3	Define Tagged Values for the Stereotype elements.	<a href="#">Define Stereotype Tagged Values</a> [1492]
4	Define constraints for the Stereotype elements.	<a href="#">Define Stereotype Constraints</a> [1500]
5	Add an Enumeration element to define a drop-down list of values for a Tagged Value on the Stereotype element.	<a href="#">Add An Enumeration to a Stereotype</a> [1493]
6	Add Shape Scripts for the Stereotype elements.	<a href="#">Add Shape Scripts</a> [1501]
7	Set the default appearance for each stereotyped model element.	<a href="#">Set Default Appearance</a> [1502]
8	Include Quick Linker definitions in the Profile.	<a href="#">Add Quick Linker Definition To Profile</a> [1523]
9	Save either the package or the diagram as the Profile, and export it.	<a href="#">Export Profile</a> [1523] <a href="#">Save Profile Options</a> [1525]
10	Incorporate the Profile into an MDG Technology and deploy the technology.	<a href="#">Create MDG Technologies</a> [1545] <a href="#">Add a Profile</a> [1548] <a href="#">Deploy An MDG Technology</a> [1580]

### Notes

- A Profile package can contain several diagrams and many elements and connectors, but **no other packages**; do not use nested packages in a Profile
- If you are creating a Profile to form part of an MDG Technology, note that you define the special Toolbox pages, diagrams and Learning Center pages for the Technology in **separate** Profiles

Learn more




- [Customize Toolbox Profiles](#) 
- [Create Custom Diagram Profiles](#) 

**7.3.2.1.1 Create a Profile Package**

The first stage in creating a UML Profile to define new model elements is to create a package that has the stereotype «*profile*» in your technical development model.

**Access** **Project Browser model context menu | Add | Add Diagram**

Create a Profile Package

Step	Description	See also
1	On the New Diagram dialog, click on <b>UML Structural</b> in the <b>Select From</b> field, and <b>Package</b> in the <b>Diagram Types</b> field. Click on the <b>OK</b> button. The new diagram opens in the Diagram View.	<a href="#">Add New Diagrams</a> 
2	Open the Profile page of the Toolbox ( <b>More tools   UML   Profile</b> ).	<a href="#">Profile Toolbox</a> 
3	Drag the <i>Profile</i> item onto the Package diagram. The New Model Package dialog displays.	
4	In the <b>Package Name</b> field, type a name for the Profile and select the <b>Automatically add new diagram</b> checkbox. Click on the <b>OK</b> button. The New Diagram dialog displays.	
5	In the <b>Name</b> field, type the diagram name, then click on <b>UML Structural</b> in the <b>Select From</b> field and <b>Class</b> in the <b>Diagram Types</b> field.	<a href="#">Add New Diagrams</a> 
6	Click on the <b>OK</b> button. The system creates a package with the stereotype « <i>profile</i> » and a child Class diagram. Depending on your system set-up, the Properties dialog for the package might display. If you wish, you can add any basic Package details you want to assign to the package, such as version, phase, or notes.	
7	On the diagram, double-click on the Profile Package to open the child diagram. You now use this child diagram to add Stereotype elements to the Profile.	<a href="#">Add Stereotypes and</a>

Step	Description	See also
		<a href="#">Metaclasses</a> 

#### Learning Center topics

- (Alt+F1) | [Enterprise Architect](#) | [Modeling Languages](#) | [Build a Profile](#) | [Create a Profile](#)

#### 7.3.2.1.2 Add Stereotypes and Metaclasses


When you are extending the UML to develop a domain-specific toolset, you start by creating a **Profile package** for the stereotypes you intend to customize. This package has at least one **child Class diagram**, and it is on this child diagram that you specify:

- The types of object that you are extending, represented by **Metaclass** elements, and
- The way in which each object is extended, represented by **Stereotype** elements

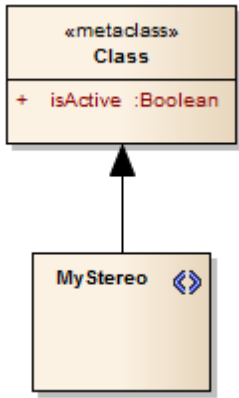
You can qualify the effect of a Stereotype on a Metaclass using a range of other tools, including:

- Shape Scripts in the Stereotype
- Tagged Values, defined by attributes in the Stereotype element
- Structured Tagged Value Classes, defined using attributes in the Stereotype element
- Enumerations, defined using attributes in the Stereotype element
- Tagged Value **connectors**, to identify possible values for a Tagged Value in an element generated with a Stereotype
- Constraints on the Stereotype element
- Special attributes, that define specific default behavior of stereotyped elements, such as the initial size and color of the element
- Modifying the default appearance of the Stereotype element

#### Add Metaclasses and Stereotypes to a Profile

Step	Description	See also
1	Open the child diagram of the Profile Package.	<a href="#">Create a Profile Package</a> 
2	<p>Drag the <i>Metaclass</i> element from the Profile page of the Toolbox onto the diagram.</p> <p>The Extend Metaclass dialog displays, listing the types of object you can extend, namely:</p> <ul style="list-style-type: none"> <li>• Core UML elements, and attributes and operations</li> <li>• Core connectors</li> <li>• Abstract metatypes such as Action types, ConnectorEnd and Gate, and</li> <li>• Stereotypes</li> </ul>	

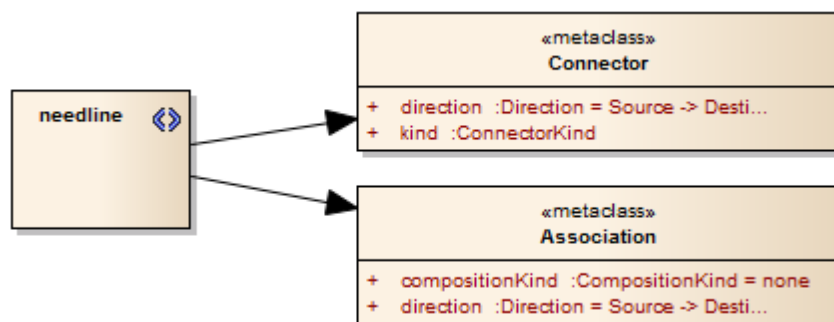


Step	Description	See also
	<p>On the Core Elements tab, you can include the set of system-defined extended elements such as ActivityRegion, Change and User, by selecting the <b>Include Extended</b> checkbox.</p> <p>On the Stereotypes tab, to specify the technology containing the stereotypes that you want to extend, click on the drop-down arrow in the top field and select the technology name.</p>	<a href="#">Create Stereotypes Extending non-UML Objects</a> <sup>[149]</sup>
3	<p>Scroll through the selected list and tick one or more object types that you want to extend.</p> <p>If you want to select all objects on a tab, click on the <b>All</b> button.</p>	
4	<p>Click on the <b>OK</b> button.</p> <p>For each checkbox that you have selected, a new <b>Metaclass</b> element is created on the diagram.</p>	
5	<p>Drag a <i>Stereotype</i> element from the Toolbox onto the diagram.</p> <p>If the Properties dialog does not display, double-click on the element on the diagram.</p>	
6	In the <b>Name</b> field, type a name for the stereotype.	
7	Click on the <b>OK</b> button.	
8	Click on the <i>Extension</i> relationship in the Toolbox and drag the connection from the Stereotype element to the Metaclass element that it will extend.	
9	<p>Your diagram now resembles the one shown below:</p>  <pre> classDiagram     class Class {         +isActive : Boolean     }     class MyStereo     MyStereo -- &gt; Class </pre> <p>The diagram shows a metaclass box labeled «metaclass» Class. It has a compartment with the attribute + isActive : Boolean. Below it is a stereotype box labeled MyStereo with a blue double-headed arrow icon. A solid line with an open arrowhead points from the MyStereo box to the Class box, representing an extension relationship.</p>	

Step	Description	See also
10	<p>Optionally, you can now add to your Stereotype element:</p> <ul style="list-style-type: none"> <li>• Stereotype tags</li> <li>• Enumeration tags</li> <li>• Structured Tagged Values</li> <li>• Tagged Value connectors</li> <li>• Special attributes</li> <li>• Constraints and/or</li> <li>• Shape Scripts</li> </ul> <p>You can also define the default appearance of the element or connector as required.</p>	<p><a href="#">Define Stereotype Tagged Values</a> <small>[1492]</small></p> <p><a href="#">Add An Enumeration to a Stereotype</a> <small>[1493]</small></p> <p><a href="#">Define a Structured Tagged Value</a> <small>[1495]</small></p> <p><a href="#">Use the Tagged Value Connector</a> <small>[1498]</small></p> <p><a href="#">Special Attributes</a> <small>[1503]</small></p> <p><a href="#">Define Stereotype Constraints</a> <small>[1500]</small></p> <p><a href="#">Add Shape Scripts</a> <small>[1501]</small></p> <p><a href="#">Set Default Appearance</a> <small>[1502]</small></p>

### Notes

- If you intend to extend a large number of model elements, rather than putting all of them on one diagram you can create additional child Class diagrams under the Profile package and add different types of Metaclass elements to different diagrams; in this case you save the **package** as the Profile, not the individual **diagrams**
- If you want to have a stereotype extending more than one metaclass, create one Stereotype element with an Extension connector to each of the Metaclass elements, as shown below:



- Stereotype elements must have unique names, but Metaclass elements can have the same name (for example, there can be several Action Metaclasses, each with a different *ActionKind* attribute)

### Learn more

- [Add New Diagrams](#) [822]

Learning Center topics

- (Alt+F1) | **Enterprise Architect** | **Modeling Languages** | **Build a Profile** | **Create a New Stereotype**



**7.3.2.1.3 Create Stereotypes Extending non-UML Objects**

A Profile is typically defined by extending core UML object types to create your own modeling language or technology; however, you can also extend non-UML objects defined by another existing technology such as Archimate, BPMN, or SysML.

Extending a non-UML object allows inheritance of the following properties from the existing stereotype:

- Tagged Values
- Shape Scripts
- Stereotype colors
- Metatype properties

Create a Stereotype extending a non-UML Object

Step	Description	See also
1	Identify the base UML type of the non-UML object that you want to extend.  You can reveal an element's base UML type by deleting its Stereotypes. For example, create a SysML1.3 Block and, using the <b>Properties window</b> , delete the Block element's Stereotype. The element's type will change from Block to Class.	
2	In the Project Browser, locate the Package with the <<profile>> Stereotype and open its child diagram.  If you do not have an existing <<profile>> Package, use the <b>MDG Technology Builder</b> option in the <b>Model Wizard</b> dialog to create a new technology, then open the diagram from the newly created <<profile>> Package.	<a href="#">Using the Profile Helpers</a> 
3	Drag the <b>Metaclass</b> icon from the Profile page of the Diagram Toolbox onto the diagram.  The Extend Metaclass dialog displays.	<a href="#">Profile Toolbox</a> 
4	Select the Stereotypes tab.	
5	From the drop-down list, select the Profile to extend (for example, <b>SysML1.3</b> ) and select the checkbox next to the non-UML Stereotype to extend (for example, <b>block</b> ).  Click on the <b>OK</b> button.  The appropriate Stereotype element is added to the Profile diagram.	
6	Add a new Stereotype by dragging the <b>Add Stereotype</b> Profile Helper from the Diagram Toolbox.	

Step	Description	See also
	<p>This will be the custom Stereotype that extends the non-UML type added to the diagram in step 5.</p> <p>When defining this Stereotype, add the Metaclass identified in step 1. For example, if you intend to extend a <b>SysML Block</b>, you would add the <b>Class</b> Metaclass.</p> <p>When you have finished, the Stereotype element and Metaclass element are displayed on the Profile Diagram.</p>	<a href="#">Add Stereotypes and Metaclasses using Profile Helpers</a> <sup>[1531]</sup>
7	Draw a <b>Generalize</b> connector from the custom Stereotype added in step 6 to the non-UML Stereotype element added in step 5.	
8	Save the diagram as a Profile.	<a href="#">Export a Profile</a> <sup>[1523]</sup>
9	Define a Toolbox Profile that has items for each of your Stereotypes.	<a href="#">Create Toolbox Profiles using the Profile Helpers</a> <sup>[1538]</sup>
10	Incorporate the saved Profiles into an MDG Technology.	<a href="#">Create MDG Technology File</a> <sup>[1545]</sup>

### Notes

- When using a Shape Script to customize the Stereotype's appearance you can use the `drawparentshape()` method to render the shape that is defined for the non-UML object being extended

### Learn more

- [Create Stereotype Profiles using the Profile Helpers](#) <sup>[1529]</sup>
- [Drawing Methods](#) <sup>[1591]</sup>

#### 7.3.2.1.4 Define Stereotype Tagged Values

You can define additional meta-information for a stereotype by adding various types of Tagged Value, which you identify as **attributes** of the Stereotype element. The simplest Tagged Values are those for which you type plain text into the **Value** field. For more complex Tagged Values, such as enumerations and Structured Tagged Values, see the *Learn more* section below.

**Access** Click on Stereotype element: **F9 > General**

#### Define Tagged Values for a Stereotype element

Field/Button	Description	See also
<b>New</b>	Click on this button to clear the dialog fields ready for creating a new attribute.	
<b>Name</b>	Type the name of the <b>tag</b> you are assigning to the Stereotype element.	
<b>Type</b>	Click on the drop-down arrow and select the attribute type.	
<b>Initial</b>	(Optional.) Type the initial value of the tag.	
<b>Notes</b>	Type a description of the tag.	
<b>Save</b>	Click on this button to save the new attribute details.	
<b>Close</b>	Click on this button to close the Attribute Properties dialog.	

#### Learn more

- [Tagged Value Types](#) <sup>[1621]</sup>
- [Add An Enumeration to a Stereotype](#) <sup>[1493]</sup>
- [Define a Structured Tagged Value](#) <sup>[1495]</sup>
- [Use the Tagged Value Connector](#) <sup>[1498]</sup>
- [Define Stereotype Tags with Predefined Tag Types](#) <sup>[1499]</sup>

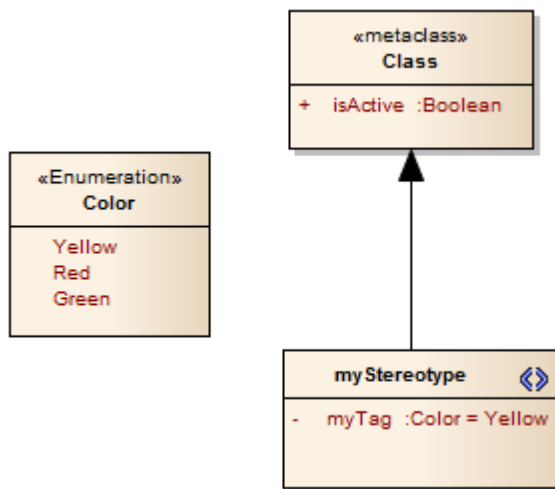
#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Modeling Languages | Build a Profile | Define Tagged Values**
- (Alt+F1) | **Enterprise Architect | Modeling Languages | Build a Profile | Define Structured Tagged Values 1**
- (Alt+F1) | **Enterprise Architect | Modeling Languages | Build a Profile | Define Structured Tagged Values 2**

##### **7.3.2.1.4.1 Add an Enumeration to a Stereotype**

Enumeration elements can be used to generate a drop-down list of values for a Tagged Value associated with a Stereotype element. The list is displayed, and the value selected, in the Tagged Values window.

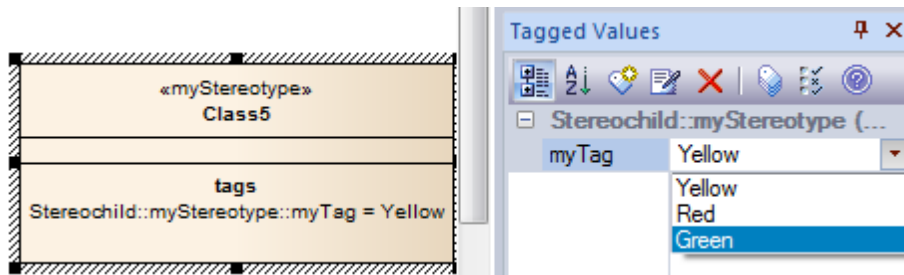
In the example below, the enumeration *Color* provides the drop-down list of values (Yellow, Red, Green) for the *myTag* Tagged Value on the *myStereotype* element.



#### Add an Enumeration to the Stereotype

Step	Description	See also
1	Open the Profile package child diagram.	
2	In the Toolbox, select <b>More tools   Profile</b> . The Profile pages of the Toolbox display.	
3	Drag an <i>Enumeration</i> item from the Toolbox onto the diagram. If the Properties dialog does not display, double-click on the element on the diagram.	
4	In the <b>Name</b> field, type the name of the new Enumeration element.	
5	Click on the Details tab and on the <b>Attributes</b> button. The Attributes Properties dialog displays.	
6	In the <b>Name</b> field, type the name of the Enumeration attribute (for example, <b>Yellow</b> ).	
7	In the <b>Type</b> field, click on the drop-down arrow and select the appropriate type (such as <b>int</b> or <b>string</b> ).	
8	Click on the <b>Save</b> button and the <b>New</b> button, and repeat steps 6 to 8 for additional attributes, to define the other values for the drop-down list.	
9	When you have defined all the values, click on the <b>Close</b> button.	

Step	Description	See also
10	Right-click on the <i>Stereotype</i> element and select the <b>Attributes</b> context menu option. The Attribute Properties dialog displays for the <b>stereotype</b> .	
11	In the <b>Name</b> field type a name for the attribute.	
12	In the <b>Type</b> field click on the ( ... ) button and select the name of the <b>Enumeration element</b> from the Select <Item> dialog.	
13	In the <b>Initial</b> field type the name of the required Enumeration attribute that defines the <b>default</b> value.	
14	Click on the <b>Save</b> and <b>Close</b> buttons.  You have now generated a drop-down list for setting the value of the tag in the Tagged Values window. When the Profile is in use, the Tagged Value for an element created with the stereotype might appear as shown:	



#### Learn more

- [Tagged Values](#) 

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Modeling Languages | Build a Profile | Define Enumeration Tagged Values 1**
- (Alt+F1) | **Enterprise Architect | Modeling Languages | Build a Profile | Define Enumeration Tagged Values 2**

#### 7.3.2.1.4.2 Define a Structured Tagged Value

If you want to define a property that has a number of components, such as an address, you can use a Structured Tagged Value. This consists of a set of related Tagged Values in a sequence that together define the property. For example, the Structured Tagged Value for the street address has the component Tagged Values:


PropertyNo - 448  
 Street - My Street  
 Town - Creswick  
 AreaCode - 3363

When you initially display this in the Tagged Value window or *tags* compartment of an element, the values of the tags are displayed in a string, such as:

**448, My Street, Creswick, 3363**

You can then expand the Structured Tagged Value to list the component tag names and values. You create a Structured Tagged Value in a profile, using an unstereotyped Class. Any attribute owned by a Stereotype element in the profile that is typed by such a Class will define the Structured Tagged Value.

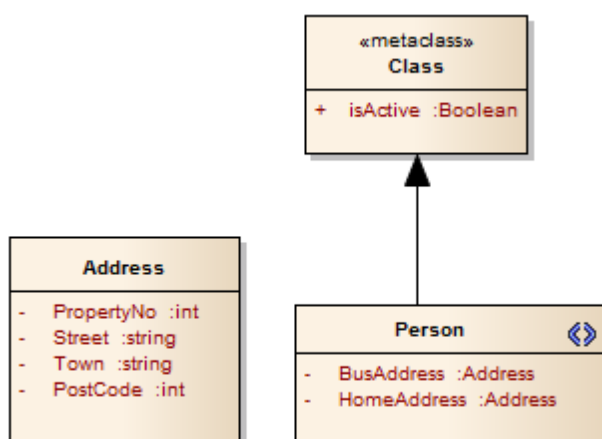
#### Create a Structured Tagged Value Class

Step	Description	See also
1	In your Profile Package, open the child Class diagram.	<a href="#">Create a Profile Package</a> 
2	In the Toolbox, select <b>More tools   Class</b> .  The contents of the Class page of the Toolbox display.	
3	Drag a Class item from the Toolbox onto the diagram.  If the Properties dialog does not display, double-click on the element on the diagram.	
4	In the <b>Name</b> field, type the name of the new Class element.	
5	Select the Details page from the Properties hierarchy, and click on the <b>Attributes</b> button.  The Attributes dialog displays.	
6	In the <b>Name</b> field, type the name of the Structured Tag attribute (for example, <b>PropertyNo</b> ).	
7	In the <b>Type</b> field, click on the drop-down arrow and select the appropriate type (such as <b>int</b> or <b>string</b> ).	
8	Click on the <b>Save</b> button, and repeat steps 6 to 8 for each remaining component tag attribute (for example: <b>Street</b> , <b>Town</b> , <b>AreaCode</b> ).	
9	When you have defined all the component tags, click on the <b>Close</b> button.	

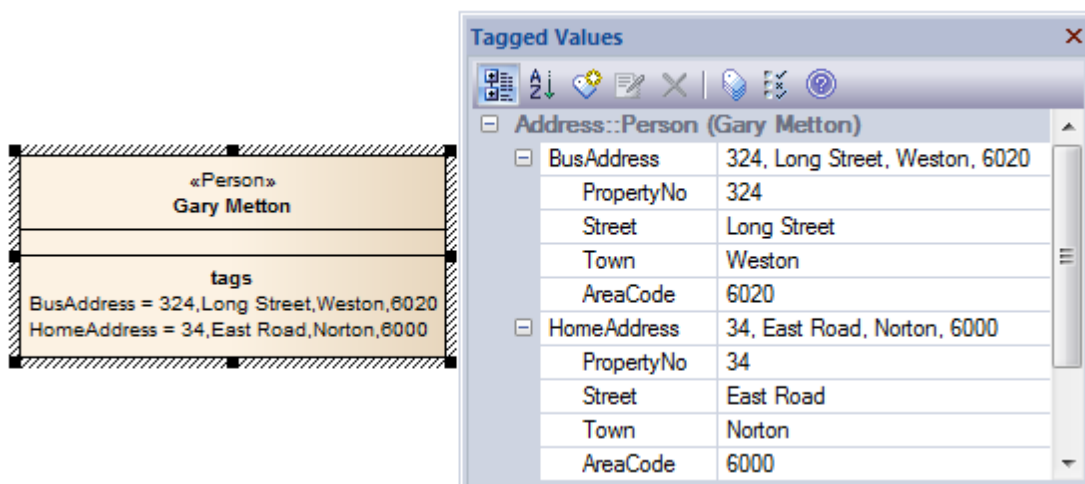


Step	Description	See also
10	Right-click on the <i>Stereotype</i> element and select the <b>Attributes</b> context menu option. The Attributes dialog displays for the <i>stereotype</i> .	
11	In the <b>Name</b> field type a name for the attribute (for example: <b>HomeAddress</b> ).	
12	In the <b>Type</b> field click on the ( ... ) button and select the name of the Structured Tagged Value Class element from the Select <Item> dialog, as the attribute's classifier.	<a href="#">Select &lt;Item&gt; Dialog</a> <sup>[994]</sup>
13	Click on the <b>Save</b> and <b>Close</b> buttons. You have now generated the components of the Structured Tagged Value to be maintained in the Tagged Values window for any element derived from this part of the profile.	<a href="#">Tagged Values</a> <sup>[1134]</sup>
14	Continue defining the profile, then save the diagram or package as a profile and either export it for use or add it to an MDG Technology file.	<a href="#">Export a Profile</a> <sup>[1523]</sup> <a href="#">Create MDG Technologies</a> <sup>[1545]</sup>

### Example



These elements when, as a Profile, are saved, exported, imported and used, provide the Structured Tagged Values for the home and business addresses in an element of the stereotype *Person*.



### Notes

- The process of applying a Structured Tagged Value through a profile is an alternative to applying the Tagged Value through an **Add-In** broadcast; see the *Learn more* topics below

### Learn more

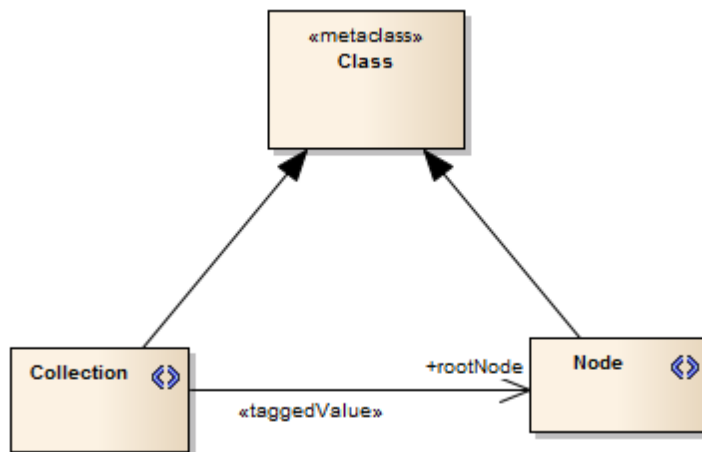
- [Tagged Value Broadcasts](#) [3084]
- [Predefined Structured Types](#) [1622]

#### 7.3.2.1.4.3 Use the Tagged Value Connector

A common situation when creating a profile is where instances of one stereotype need to reference elements with another stereotype applied. For example, an element that defines a Collection might have a Tagged Value called **rootNode** to identify the Root of that Collection, which will be a Class with the stereotype <<Node>>. In the Tagged Values window, the user would click on the selection button ( ... ) against the **rootNode** Tagged Value; when the Select <Item> dialog displays, the user can locate all Nodes in the current model, and select one of these elements as the value of the tag.

To achieve this, you use the **Tagged Value connector**, from the Profile pages of the Toolbox. A Tagged Value connector defines a reference-type (that is, RefGUID) Tagged Value owned by the source stereotype; the Tagged Value name is the name of the **target role** of this connector, and the Tagged Value is limited to referencing elements with the stereotype of the target element.

The following diagram demonstrates how you might use the connector to create the example above. A Profile defines two stereotypes: «Collection» and «Node» (both of which extend the Metaclass Class). The «Collection» stereotype owns a Tagged Value connector with the target role *rootNode*, pointing to the «Node» stereotype. You enter the target role name on the Target Role page of the connector Properties dialog.



#### Learn more

- [Select <Item> Dialog](#) <sup>[994]</sup>
- [Target Role](#) <sup>[1132]</sup>
- [Source Role](#) <sup>[1130]</sup>

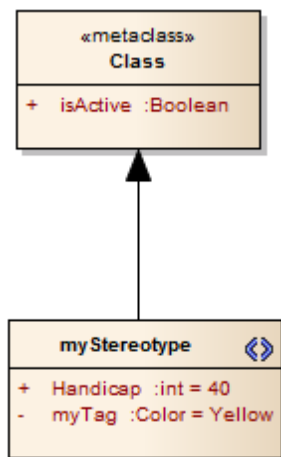
#### **7.3.2.1.4.4 With Predefined Tag Types**

Tagged Values define a wide range of properties and characteristics of a model element, and some of these properties have complex or structured values. For example, you might want your user to select a value between upper and lower limits (using 'Spin' arrows), set a date and time, select a color from a palette, or work through a checklist.

You create these complex Tagged Values from any of a number of predefined structured Tagged Value types and filters, some of which you might have created yourself (**Settings | UML Types > Tagged Value Types**); the attribute you create in the Stereotype element has the same name as the Tagged Value Type.

#### Assign Tagged Values to Stereotypes

Having created a structured Tagged Value, you assign it to the Stereotype element in the same way as for simple Tagged Values, by creating an attribute in the Stereotype element with the name of the Tagged Value Type. For example, to make the Tagged Value *Handicap* appear in a stereotype, create an attribute named *Handicap*. Depending on the tag type, you can set the default value for the tag by giving the attribute an *Initial* value.



#### Learn more

- [Predefined Structured Types](#) <sup>[1622]</sup>
- [Create Structured Tagged Values](#) <sup>[1622]</sup>
- [Define Stereotype Tagged Values](#) <sup>[1492]</sup>

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect** | **Modeling Languages** | **Build a Profile** | **Define Typed Tagged Values**

#### 7.3.2.1.5 Define Stereotype Constraints

If you need to define the conditions and rules under which the Stereotype element operates and exists, you can do this by setting Constraints on the element. Typical constraints are pre- and post- conditions, which indicate things that must be true before the element is created or accessed and things that must be true after the element is destroyed or its action is complete.

You can show the constraints for an element directly on the diagram, using the Feature Visibility function.

**Access**    **Double-click Stereotype element > Constraints**

#### Define constraints for a stereotype

Field/Button	Description	See also
<b>New</b>	Click on this button to clear the fields ready to create a new constraint.	
<b>Constraint</b>	Type the value of the constraint.	
<b>Type</b>	Click on the drop-down arrow and select the appropriate type ( <b>Pre-condition</b> , <b>Post-condition</b> or <b>Invariant</b> ).	

Field/Button	Description	See also
<b>Status</b>	Click on the drop-down arrow and select the appropriate status.	
Notes	Type any additional information required.	
<b>Save</b>	Click on this button to save the constraint data.	
<b>OK</b>	Click on this button to close the dialog.	

#### Learn more


- [Feature Visibility](#)<sup>[845]</sup>

#### 7.3.2.1.6 Add Shape Scripts

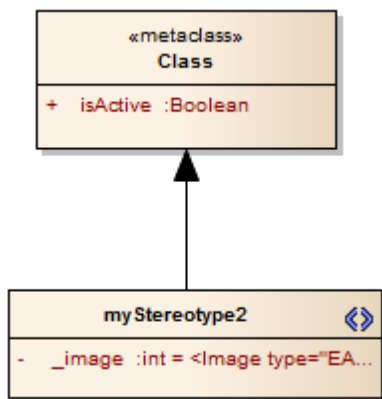
UML elements and connectors each have a standard appearance, in terms of shape, color and labeling. It is possible to change the appearance of a type of element or connector in a number of ways, using a Shape Script to define the exact feature you want to impose on the default - or main - shape. If you want to standardize the appearance, to apply to many elements, you attach the Shape Script to an attribute of a Stereotype element in a UML Profile (such as an MDG Technology UML Profile).

**Access** **Profile Stereotype element: ( F9 ) > General**

#### Add a Shape Script to a Stereotype element

Step	Description	See also
<b>1</b>	In the <b>Name</b> field, type <b>_image</b> .	
<b>2</b>	Click on the  button next to the <b>Initial Value</b> field. The Shape Editor dialog displays.	<a href="#">Shape Editor</a> <sup>[1584]</sup>
<b>3</b>	Enter the Shape Script in the Shape Editor dialog. When you have finished writing the Shape Script, click on the <b>OK</b> button and then the <b>Close</b> button.	<a href="#">Write Scripts</a> <sup>[1585]</sup>

The Stereotype element now resembles this example:



### Notes

- Your Shape Script might include externally-defined images; in this case the Shape Script would include the image method, specifying the image file name prefixed with the technology name
- If you are creating a Shape Script for an Association Class, note that the Shape Script is applied to both the Class part and the Association part; therefore, you might have to include logic in the shape main that tests the type of the element so that you can give separate drawing instructions for Class and for Association

Such logic is not necessary in the:

- shape source or shape target, which are ignored by Classes, or the
- decoration shapes which are ignored by Associations
- You can also apply Shape Scripts to elements on an ad hoc basis, attaching the Shape Script to a stereotype defined on the UML Types dialog (**Settings > UML Types**)

### Learn more

- [Create UML Profiles](#) <sup>1485</sup>
- [Shape Scripts](#) <sup>1562</sup>

### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Modeling Languages | Build a Profile | Defining an Image**

#### 7.3.2.1.7 Set Default Appearance

If you want to define a **simple** default appearance for a stereotyped element or connector, you can select the Stereotype element that defines it and just set any or all of the:

- Background/fill color
- Border color
- Border line width
- Font type, style (bold, italic) and size, or
- Font color

To set these, you use the Default Appearance dialog.

**Access** Click on Stereotype element: F4

#### Notes

- When you save the Profile defining the stereotyped elements and connectors, select the **Color and Appearance** checkbox on the Save UML Profile dialog

#### Learn more

- [Set an Element's Default Appearance](#)<sup>[927]</sup>
- [Add Stereotypes and Metaclasses](#)<sup>[1488]</sup>
- [Saving Profiles](#)<sup>[1523]</sup>

### 7.3.2.1.8 Special Attributes

It is possible to define a number of special features and behaviors of a stereotyped model element, such as the icon to represent it in the Project Browser and Diagram Toolbox, the default location of any image files associated with the stereotype, the dimensions of the element in a diagram, or whether the appearance is defined by a Shape Script. You define these features in your Profile using special attributes that can be applied to either the:








- Stereotype elements or
- Metaclass elements, referring to the stereotypes that extend them

**Access** Click on profile element: F9 > General

#### Set the attribute(s)

Field/Button	Description	See also
<b>Name</b>	Type the name of the attribute (as listed in the tables below).	
<b>Initial</b>	Type or select the initial value of the attribute.	
<b>Save</b>	Click on this button to save your changes. (You do not need to set any other values.)	
<b>Close</b>	Click on this button to close the dialog.	

#### Stereotype element Attributes

Attribute	Meaning	See also
<b>_defaultAttributeType</b>	<p>Defines the default type of the new attributes created from the Diagram Toolbox. Use this in a Stereotype element that extends an <i>Attribute</i> Metaclass, and set the attribute type as the initial value.</p> <p>If you do not provide this, the system creates attributes with the default type <b>int</b>.</p>	
<b>icon</b>	<p>Contains the bitmap file location of the 16x16-pixel icon displayed beside all elements defined by the Stereotype, in the Project Browser. This does not apply to <i>Package</i> elements. The icon is also automatically used as the Diagram Toolbox image wherever the stereotyped element is listed.</p> <p>For a transparent background, you can use light grey - RGB (192,192,192).</p> <p>For this attribute to work correctly, also set the <i>_metatype</i> attribute (see below).</p>	
<b>_image</b>	<p>Identifies a Shape Script definition, the script for which is created in the <b>Initial Value</b> field.</p> <p>For this attribute to take effect, you need to set the <b>Alternate Image</b> option when you save the Profile.</p>	<a href="#">Add Shape Scripts</a>  <sup>[1501]</sup>  <a href="#">Export a Profile</a>  <sup>[1523]</sup>
<b>_instanceMode</b>	Define what happens when an instance is created of a stereotyped element.	<a href="#">Define Creation of Instance</a>  <sup>[1510]</sup>
<b>_instanceOwner</b>		
<b>_instanceType</b>		
<b>_metatype</b>	Defines stereotypes as metatypes, so that the identity of an element as a custom, stereotyped element is hidden.	<a href="#">Define a Stereotype as a Metatype</a>  <sup>[1508]</sup>
<b>_sizeY</b>	<p>Sets the initial height of the element, in pixels, at 100% zoom.</p> <p>For this attribute to take effect, you need to set the <b>Element Size</b> option when you save the Profile.</p>	<a href="#">Export a Profile</a>  <sup>[1523]</sup>
<b>_sizeX</b>	<p>Sets the initial width of the element, in pixels, at 100% zoom.</p> <p>For this attribute to take effect, you need to set the <b>Element Size</b> option when you save the Profile.</p>	<a href="#">Export a Profile</a>  <sup>[1523]</sup>
<b>_strictness</b>	Defines the degree to which a stereotyped element can have more than one stereotype applied to it.	<a href="#">Define Multiple-Stereotype Level</a>  <sup>[1509]</sup>



**Metaclass element Attributes**

Attribute	Meaning	See also
<b>_AttInh</b>	If set to <b>1</b> , sets the <b>Inherited Features: Show Attributes</b> checkbox to selected on each new stereotyped model element.	<a href="#">Feature Visibility</a> <sup>[845]</sup>
<b>_AttPkg</b>	If set to <b>1</b> , sets the <b>Attribute Visibility: Package</b> checkbox to selected on each new stereotyped model element.	
<b>_AttPri</b>	If set to <b>1</b> , sets the <b>Attribute Visibility: Private</b> checkbox to selected on each new stereotyped model element.	
<b>_AttPro</b>	If set to <b>1</b> , sets the <b>Attribute Visibility: Protected</b> checkbox to selected on each new stereotyped model element.	
<b>_AttPub</b>	If set to <b>1</b> , sets the <b>Attribute Visibility: Public</b> checkbox to selected on each new stereotyped model element.	
<b>compositeKind</b>	When applied to an Association, defines whether the source or target end is an aggregate or composite. Permitted values are: <ul style="list-style-type: none"> <li>• <b>None</b></li> <li>• <b>Aggregate at Source</b></li> <li>• <b>Aggregate at Target</b></li> <li>• <b>Composite at Source</b></li> <li>• <b>Composite at Target</b></li> </ul>	<a href="#">Aggregation</a> <sup>[1392]</sup>
<b>_ConInh</b>	If set to <b>1</b> , sets the <b>Show Element Compartments: Inherited Constraints</b> checkbox to selected on each new stereotyped model element.	<a href="#">Feature Visibility</a> <sup>[845]</sup>
<b>_Constraint</b>	If set to <b>1</b> , sets the <b>Show Element Compartments: Constraints</b> checkbox to selected on each new stereotyped model element.	
<b>_defaultDiagram Type</b>	Defines the type of child diagram created when an element is made composite.	<a href="#">Define Child Diagram Type</a> <sup>[1512]</sup>
<b>direction</b>	Automatically created when any type of connector Metaclass element is dragged from the Profile toolbox page onto a diagram. You can set a value for this attribute in preference to using the <i>_SourceNavigability</i> or <i>_TargetNavigability</i> attributes (below).	
<b>_HideStyle</b>	Set the <b>Initial Value</b> field to a comma-separated list of stereotypes to hide those stereotypes by setting the <b>Hide Stereotyped Features</b> filter for each new stereotyped model	<a href="#">Feature Visibility</a> <sup>[845]</sup>

Attribute	Meaning	See also
	element.	
<b>_isVertical</b>	Set to <b>True</b> for a stereotyped <i>ActivityPartition</i> to make the default Activity Partition orientation vertical.	<a href="#">Partition</a> <sup>[1325]</sup>
<b>_lineStyle</b>	Sets the line style of a stereotyped connector; the <b>Initial Value</b> of the attribute can be one of: <ul style="list-style-type: none"> <li>• <b>direct</b></li> <li>• <b>auto</b></li> <li>• <b>custom</b></li> <li>• <b>bezier</b></li> <li>• <b>treeH</b> (horizontal)</li> <li>• <b>treeV</b> (vertical)</li> <li>• <b>treeLH</b> (lateral horizontal)</li> <li>• <b>treeLV</b> (lateral vertical)</li> <li>• <b>orthogonalS</b> (orthogonal, square corners)</li> <li>• <b>orthogonalR</b> (orthogonal, rounded corners)</li> </ul>	<a href="#">Connector Styles</a> <sup>[1114]</sup>
<b>_makeComposite</b>	Makes each stereotyped element a composite element when it is created.	<a href="#">Define Composite Elements</a> <sup>[1511]</sup>
<b>_OpInh</b>	If set to <b>1</b> , sets the <b>Inherited Features: Show Operations</b> checkbox to selected on each new stereotyped model element.	<a href="#">Feature Visibility</a> <sup>[845]</sup>
<b>_OpPkg</b>	If set to <b>1</b> , sets the <b>Operation Visibility: Package</b> checkbox to selected on each new stereotyped model element.	
<b>_OpPri</b>	If set to <b>1</b> , sets the <b>Operation Visibility: Private</b> checkbox to selected on each new stereotyped model element.	
<b>_OpPro</b>	If set to <b>1</b> , sets the <b>Operation Visibility: Protected</b> checkbox to selected on each new stereotyped model element.	
<b>_OpPub</b>	If set to <b>1</b> , sets the <b>Operation Visibility: Public</b> checkbox to selected on each new stereotyped model element.	
<b>_PType</b>	If set to <b>1</b> , sets the <b>Show element type (Port or Part only)</b> checkbox to selected on each new stereotyped model element.	
<b>_ResInh</b>	If set to <b>1</b> , sets the <b>Show Element Compartments: Inherited Responsibilities</b> checkbox to selected on each new stereotyped model element.	

Attribute	Meaning	See also
<b>_Responsibility</b>	If set to <b>1</b> , sets the <b>Show Element Compartments: Responsibilities</b> checkbox to selected on each new stereotyped model element.	
<b>_Runstate</b>	If set to <b>1</b> , sets the <b>Hide Object Runstate in current diagram</b> checkbox to selected on each new stereotyped model element.	
<b>_SourceAggregation</b>	<b>Deprecated.</b> See <b>compositeKind</b> , above.	
<b>_SourceMultiplicity</b>	Sets the multiplicity of the source element, such as <b>1..*</b> or <b>0..1</b> .	<a href="#">Source Role</a> <sup>[1130]</sup>
<b>_SourceNavigability</b>	If the connector is non-navigable, set this attribute to <b>Non-Navigable</b> .  If other values are more appropriate, use the <i>direction</i> attribute (above).	
<b>_subtypeProperty</b>	Specifies the fully qualified name of the Tagged Value that is used to generate a popup submenu each time an element with the stereotype is created from the Toolbox.  The Tagged Value is an enumeration and the submenu consists of a command for each enumeration literal. The Tagged Value is initialized with whichever command is selected on the submenu; if none is selected (such as if the user clicks off the submenu) then the default value is used as normal.  For example, if you create a <i>BPMN 2 Activity</i> element, a submenu displays listing the task types such as <b>BusinessRule</b> , <b>Manual</b> and <b>Receive</b> . Selecting one of these values sets it as the value of the <i>taskType</i> Tagged Value.  The Tagged Value is effectively the Activity's subtype; in the BPMN 2 profile, in the format <i>profile::stereotype::tag</i> , the <b>subtypeProperty</b> for the Activity stereotype would be:  <i>BPMN2.0::Activity::taskType</i> .	<a href="#">BPMN 2.0 Activity</a> <sup>[1904]</sup>
<b>_Tag</b>	If set to <b>1</b> , sets the <b>Show Element Compartments: Tags</b> checkbox to selected on each new stereotyped model element.	<a href="#">Feature Visibility</a> <sup>[845]</sup>
<b>_tagGroupings</b>	Maps the Tagged Values into the tag groups (below) displayed in the Tagged Values window, in the form:  <i>tagName1=groupName1;tagName2=groupName2;</i>  This facility currently is available for <b>object</b> types only, not for other types such as attributes.	
<b>_tagGroups</b>	Defines a comma-separated list of required groups in the order	<a href="#">Define Tag Groupings</a> <sup>[1514]</sup>

Attribute	Meaning	See also
	<p>in which they are to be displayed in the Tagged Values window. For example:</p> <p><i>groupName1,groupName2,groupName3</i></p> <p>This facility currently is available for <b>object</b> types only, not for other types such as attributes.</p>	
<b>_tagGroupStates</b>	<p>Maps <b>_tagGroups</b> displayed in the Tagged Values window to the state of <b>open</b> or <b>closed</b>, in the form:</p> <p><i>groupName1=open;groupName2=closed;</i></p> <p>This facility currently is available for <b>object</b> types only, not for other types such as attributes.</p>	
<b>_TagInh</b>	If set to <b>1</b> , sets the <b>Show Element Compartments: Inherited Tags</b> checkbox to selected on each new stereotyped model element.	<a href="#">Feature Visibility</a> <sup>[845]</sup>
<b>_TargetAggregation</b>	<b>Deprecated.</b> See <b>compositeKind</b> , above.	
<b>_TargetMultiplicity</b>	Sets the multiplicity of the target element, such as <b>1..*</b> or <b>0..1</b> .	<a href="#">Target Role</a> <sup>[1132]</sup>
<b>_TargetNavigability</b>	<p>If the connector is non-navigable, set this attribute to <b>Non-Navigable</b>.</p> <p>If other values are more appropriate, use the <i>direction</i> attribute (above).</p>	

#### Notes

- Where an attribute is set to **1** to turn a feature on, setting it to **0** turns the feature off

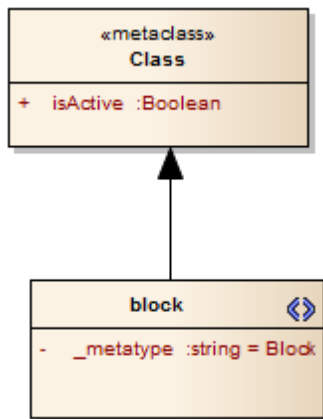
#### Learning Center topics

- (Alt+F1) | [Enterprise Architect](#) | [Modeling Languages](#) | [Build a Profile](#) | [Defining an Image](#)

#### 7.3.2.1.8.1 Define a Stereotype as a Metatype

If you want to hide the identity of a custom element as a stereotyped UML element, you can set the **\_metatype** special attribute in the Stereotype element that defines it. The *\_metatype* attribute also makes custom element types appear in contexts where only Enterprise Architect's inbuilt types would normally appear; for example, in the lists of element types in the Relationship Matrix.

In the following example from SysML, *block* is defined as a Stereotype element that extends a UML Class.



However, a SysML user is not interested in UML **Classes**, only in SysML **Blocks**. If you set the `_metatype` attribute to **Block**, any element created from that stereotype, while behaving like a stereotyped Class in most contexts will:

- Show *Block* <name> rather than *Class* <name> as the title of its Properties dialog
- Be auto-numbered as *Block1* not *Class1* on creation, and
- Appear as *Block* not *Class* in many other contexts throughout Enterprise Architect

#### Learn more

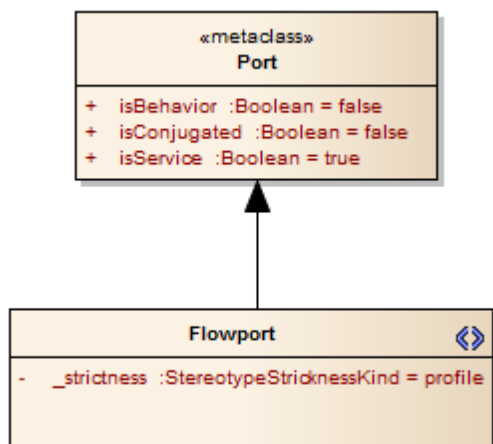
- [Stereotype and Metaclass Attributes](#) 1503

#### 7.3.2.1.8.2 Define Multiple-Stereotype Level

An element can have more than one stereotype applied to it. You can define the **level** to which multiple stereotypes can be applied, by creating the `_strictness` special attribute in the defining Stereotype element. The type of the attribute is *StereotypeStrictnessKind*, with one of four values in the **Initial Value** field:

- **profile**, which states that an element of this type cannot be given more than one stereotype from the same Profile
- **technology**, which states that an element of this type cannot be given more than one stereotype from the same technology
- **all**, which states that an element of this type cannot have multiple stereotypes at all, or
- **none**, which is the default behaviour and states that there are no restrictions on the use of multiple stereotypes

The following example is from SysML and shows that a `«flowPort»` cannot have any other stereotype applied to it.



#### Learn more

- [Stereotype and Metaclass Attributes](#) [1503]

#### 7.3.2.1.8.3 Define Creation of Instance

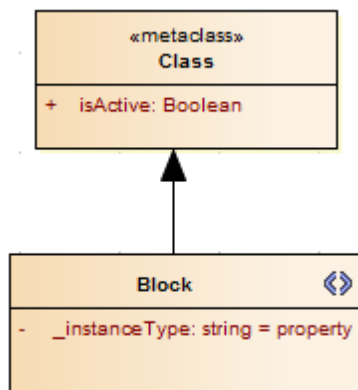
A stereotyped element can be the classifier of instances created from it. You can define **how an instance is created** from that stereotyped element, by adding special attributes to the defining Stereotype. The attributes **modify the text** on the Paste As dialog that displays when a stereotyped element is dragged out of the Project Browser onto a diagram.

#### Attributes

Attribute	Meaning	See also
<b>_instanceMode</b>	<p>Changes the second option for the <b>Paste as</b> field on the dialog to either:</p> <ul style="list-style-type: none"> <li>• <b>Instance (&lt;element type&gt;)</b> or</li> <li>• <b>Property (Object)</b></li> </ul> <p>The text is determined by the value (<b>Instance</b> or <b>Property</b>) of the attribute's <b>Initial Value</b> field.</p> <p>If the attribute is not applied, the option defaults to <b>Instance</b>.</p>	
<b>_instanceOwner</b>	<p><b>DEPRECATED</b></p> <p>Modifies the second option of the <b>Paste as</b> field on the dialog to:</p> <ul style="list-style-type: none"> <li>• <b>as Instance of &lt;element type&gt;</b></li> </ul> <p>The text is determined by the value of the attribute's <b>Initial Value</b> field, such as <b>Block</b>.</p> <p>If the attribute is not applied, the option defaults to <b>Element</b>.</p>	
<b>_instanceType</b>	Modifies the second option of the <b>Paste as</b> field on the dialog to:	<a href="#">Define a</a>

Attribute	Meaning	See also
<b>e</b>	<ul style="list-style-type: none"> <li>as Instance of Element (ProfileName::&lt;&lt;stereotype&gt;&gt;)</li> </ul> <p>The &lt;&lt;stereotype&gt;&gt; value is defined in the <b>Initial Value</b> field of the attribute, and corresponds to the metatype given to the stereotyped element using the <b>_metatype</b> attribute.</p>	<a href="#">Stereotype as a Metatype</a> <sup>[1508]</sup>

The following example from SysML shows the definition of any instances of a SysML Block element that might be created.



When a user drags a SysML Block element from the Project Browser onto a diagram, the system checks the **\_instanceType** attribute value and searches the SysML Profile for an element template with a matching **\_metatype** attribute value, and generates the instance from that - with the above definition you would get a Block element with the «property» stereotype.

#### Learn more

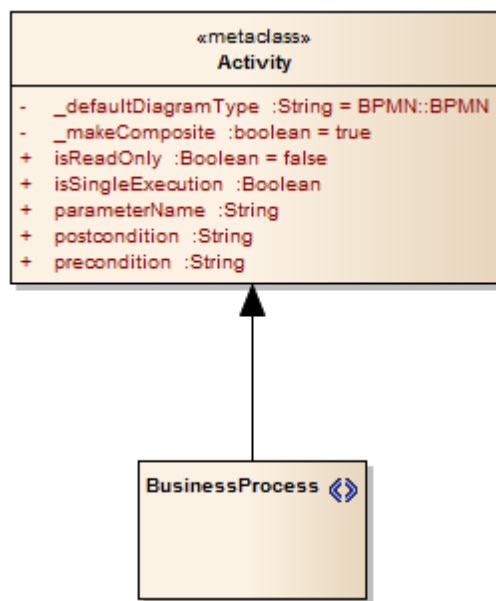
- [Paste Element from the Project Browser](#)<sup>[833]</sup>
- [Stereotype and Metaclass Attributes](#)<sup>[1508]</sup>

#### 7.3.2.1.8.4 Define Composite Elements

A stereotyped element can be created automatically as a composite element. You can define this, and whether the child diagrams of the composite are of a specific type, using special attributes.

To define whether an element is always made composite on creation, you apply the **\_makeComposite** special attribute to the appropriate **metaclass** element (not to a stereotype element). A stereotyped **class**, when created, does not default to having a child diagram, so you use the **\_makeComposite** attribute to trigger creation of the child diagram. For a stereotyped composite, the child diagram is of the usual default diagram type for the metaclass; you can change the child diagram type using the **\_defaultDiagramType** special attribute to identify the preferred diagram type,

The following example from BPMN shows that a *BusinessProcess* element is always created as a Composite element with a BPMN custom child diagram.



#### Learn more

- [Define Child Diagram Type](#) <sup>[1512]</sup>
- [Stereotype and Metaclass Attributes](#) <sup>[1503]</sup>

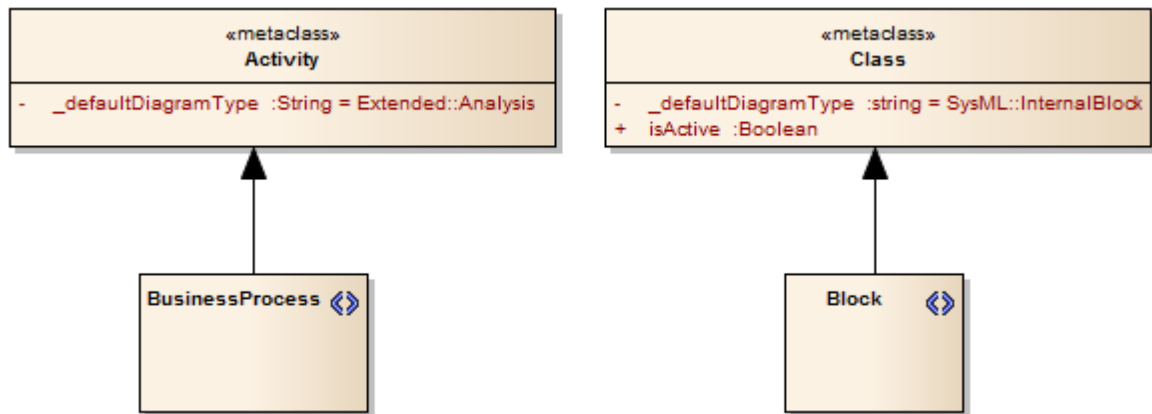
#### 7.3.2.1.8.5 Define Child Diagram Type

If you define a stereotyped element type as being a **composite**, its **child diagram type** is initially the same as the default for the Metaclass element you extend. You can change the diagram type to any other of the inbuilt UML or Extended types, or to any of your own custom diagram types, using the **\_defaultDiagramType** special attribute. As the diagram type defaults from the Metaclass element, you set the attribute on that Metaclass element (and not the Stereotype element) to change the default.

You identify the child diagram type in the **Initial Value** field for the attribute. The actual values for the inbuilt UML and Extended diagram types are listed below, in the *Initial Values* section. If you want to set a custom diagram type, you prefix the diagram type name with the **diagram profile name** and '::.'. The diagram profile name is the name given to the profile when you save it, which by default is the name of the Profile package or Profile diagram. We recommend that the diagram profile name is based on the technology name. You can also use the **\_defaultDiagramType** attribute for packages, extending the Package Metaclass element.

The following examples show a *«BusinessProcess»Activity* that, when made a composite element, automatically creates an Analysis diagram, and a *«block»* stereotype that creates a SysML *InternalBlock* custom diagram.





### Initial Values

The following strings can be used in the **Initial Value** field for `_defaultDiagramType`, to identify the inbuilt UML and Extended diagram types:

- UML Behavioral::Use Case
- UML Behavioral::Activity
- UML Behavioral::State Machine
- UML Behavioral::Communication
- UML Behavioral::Sequence
- UML Behavioral::Timing
- UML Behavioral::Interaction Overview
- UML Structural::Package
- UML Structural::Class
- UML Structural::Object
- UML Structural::Composite Structure
- UML Structural::Component
- UML Structural::Deployment
- Extended::Custom
- Extended::Requirements
- Extended::Maintenance
- Extended::Analysis
- Extended::User Interface
- Extended::Data Modeling
- Extended::ModelDocument.

### Notes

- Although we recommend that the **diagram profile name** for custom diagram types is based on the technology name, the attribute prefix is **not** a **direct reference** to the technology name

### Learn more

- [Define Composite Elements](#) <sup>[1511]</sup>
- [Custom Diagram Types](#) <sup>[1570]</sup>

#### 7.3.2.1.8.6 Define Tag Groupings

In developing a stereotyped element in a Profile, you might define a large number of Tagged Values. For example, a BPMN Activity element in the BPMN 2.0 Profile has 30 Tagged Values. By default, in the Tagged Values window for the element, these Tagged Values would initially **all** be displayed in alphabetical order, which might split related tags if they happen to have alphabetically distant names. To keep related tags together and control which tags are initially shown, in the BPMN 2.0 Profile the Tagged Values have been **grouped**. You can apply the same solution, using three tag grouping special attributes in the **Metaclass element** extended by the Stereotype element in which the tags are defined as attributes.

You apply the grouping using:

- **\_tagGroups** to define the group names
- **\_tagGroupings** to define which tags go into each group
- **\_tagGroupStates** to define which tag groups are initially expanded in the Tagged Values window, and which are collapsed

The Tagged Values window for the BPMN 2.0 Activity element initially displays as shown:

<input type="checkbox"/> BPMN2.0::Activity (Activity4)	
<input type="checkbox"/> Base Element	
<input type="checkbox"/> Activity	
activityType	Task
calledActivityRef	
instantiate	false
isACalledActivity	false
isATransaction	false
isForCompensation	false
resources	
<input type="checkbox"/> Task	
messageRef	
operationRef	
rendering	
script	
scriptFormat	
taskType	Service
<input type="checkbox"/> AdHoc	
<input type="checkbox"/> Loop	
<input type="checkbox"/> Sub-Process	
<input type="checkbox"/> Callable Element	
<input type="checkbox"/> Execution	
<input type="checkbox"/> Other	

To achieve this display of the BPMN 2.0 Activity Tagged Values, the Technology Developer defined the special attributes in the Activity Metaclass element as shown:

Attribute	Values	See also
<b>_tagGroups</b>	Base Element,Activity,Task,AdHoc,Loop,Sub-Process,Callable Element,Execution,Other	<a href="#">Special Attributes</a> <sup>[1507]</sup>
<b>_tagGroupings</b>	auditing=Base Element;categoryValue=Base Element;documentation=Base Element;monitoring=Base Element;activityType=Activity;calledActivityRef=Activity;instantiate=Activity;isACalledActivity=Activity;isATransaction=Activity;isForCompensation=Activity;resources=Activity;messageRef=Task;operationRef=Task;rendering=Task;script=Task;scriptFormat=Task;taskType=Task;adHoc=AdHoc;adHocOrdering=AdHoc; ... (and so on)	
<b>_tagGroupStates</b>	Base Element=closed;Activity=open;Task=open;AdHoc=closed;Loop=closed;Sub-Process=closed;Callable Element=closed;Execution=closed;Other=closed	

#### Notes

- This facility currently is available for object types only, not for other types such as attributes

#### 7.3.2.1.9 Quick Linker

When a user is creating new elements and connectors on a diagram they can simplify the process by using the Quick Linker arrow, which displays a list of the common connectors that can issue from a selected element and a list of the common elements each connector can connect to. These lists are derived from a Quick Linker definition, which is a Comma Separated Value (CSV) format file. As part of a Profile, you can add to or replace the built-in Quick Linker definitions using your own CSV files. These files are best manipulated in a spreadsheet, which should be set up to save the file as comma-separated text without quotation marks around text fields. You integrate the definition with the Profile by adding the CSV text to a Document Artifact element on the Profile diagram.

#### Notes

- The philosophy behind a Quick Linker definition is not to provide a **complete** list of valid or legal connections, but a short and convenient list of the **commonest** connections for the given context

#### Learn more

- [The Quick Linker](#) <sup>[896]</sup>
- [Quick Linker Definition Format](#) <sup>[1518]</sup>
- [Quick Linker Example](#) <sup>[1519]</sup>
- [Hide Default Quick Linker Settings](#) <sup>[1521]</sup>
- [Quick Linker Object Names](#) <sup>[1521]</sup>
- [Add Quick Linker Definition To Profile](#) <sup>[1523]</sup>

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect** | **Modeling Languages** | **Build a Profile** | **Define Linking Rules**

### 7.3.2.1.9.1 Quick Linker Definition Format

In order to replace or change the Quick Linker menus that are displayed when a user drags the Quick Linker arrow from one of your profile elements on a diagram, you can create or edit the corresponding Quick Linker definition. This is a Comma Separated Value (CSV) text file consisting of records (rows), each record consisting of 23 comma-separated fields as defined in the table below. Some of these fields define the menu command and some act as filters, with the entry being ignored if the filter condition isn't met. A Quick Linker definition can include comments: all lines in which `//` are the first two characters are ignored by Enterprise Architect. Quotes (") in the field values are not required.

Each record of the Quick Linker definition represents a single combination of entries on the Quick Linker menus; that is, for the selected source element, a specific connector type and specific target element type. A menu is populated from all rows that satisfy the filters; that is, the first menu lists all defined connectors that are legal and valid for the source element type, and the second menu lists all target elements that are legal and valid for the combination of source element and connector type.

#### Quick Linker Definition fields

Column	Title (enter as comment for guidance)	Description
<b>A</b>	<b>Source Element Type</b>	Identifies a valid source element in the Profile; if a connector is being dragged away from this type of element, the row is evaluated. Otherwise, the row is ignored.
<b>B</b>	<b>Source Stereotype Filter</b>	Identifies a stereotype of the source element base type (for example, an Event source element can be a normal Event, or a Start Event, Intermediate Event or End Event stereotyped element).  If set, and if a connector is being dragged away from an element of this stereotype, the row is evaluated. Otherwise, the row is ignored.
<b>C</b>	<b>Target Element Type</b>	Identifies a valid target element in the Profile.  If <b>set</b> , and if a connector is being dragged <b>onto</b> this type of element, the row is evaluated.  If <b>blank</b> , and if a connector is being dragged onto an empty space on the diagram, the row is evaluated.  Otherwise the row is ignored.
<b>D</b>	<b>Target Stereotype Filter</b>	Identifies a stereotype of the target element base type.  If set, if <b>Target Element Type</b> is also set, and if a connector is being dragged onto an element of this stereotype, the row is evaluated. Otherwise, the row is ignored.
<b>E</b>	<b>Diagram Filter</b>	Contains either an inclusive list or an exclusive list of diagram types, which limits the diagrams the specified connector can be created on. <ul style="list-style-type: none"> <li>• Each diagram name is terminated by a semi-colon; for example:</li> </ul>

Column	Title (enter as comment for guidance)	Description
		<p><i>Collaboration;Object;Custom;</i></p> <ul style="list-style-type: none"> <li><b>Custom</b> diagram types from MDG Technologies can be referenced using the fully qualified diagram type (DiagramProfile::DiagramType); for example:  <i>BPMN2.0::Business Process;BPMN2.0::Choreography;BPMN2.0::Collaboration;</i></li> <li>As a shorthand for <b>all</b> diagram types in a diagram profile you can use the '*' wildcard, which must be preceded by the diagram profile ID; for example:  <i>BPMN2.0::*;</i></li> <li>Each <b>excluded</b> diagram name is preceded by an exclamation mark; for example:  <i>!Sequence;</i></li> </ul>
<b>F</b>	<b>New Element Type</b>	Defines the type of element to be created if the connector is dragged into open space, provided that the <b>Create Element</b> field is set to <b>TRUE</b> .
<b>G</b>	<b>New Element Stereotype</b>	Defines the type of element <b>stereotype</b> to be created if the connector is dragged into open space, provided that the <b>Create Element</b> field is set to <b>TRUE</b> .
<b>H</b>	<b>New Link Type</b>	Defines the type of connector to create, if <b>Create Link</b> is also set to <b>TRUE</b> .
<b>I</b>	<b>New Link Stereotype</b>	Defines the type of connector <b>stereotype</b> to correct, if <b>Create Link</b> is also set to <b>TRUE</b> .
<b>J</b>	<b>New Link Direction</b>	<p>Defines the connector direction, which can be:</p> <ul style="list-style-type: none"> <li><b>directed</b> (always creates an association from source to target)</li> <li><b>from</b> (always creates an association from target to source)</li> <li><b>undirected</b> (always creates an association with unspecified direction)</li> <li><b>bidirectional</b> (always creates a bi-directional association), or</li> <li><b>to</b> (creates either a directed or undirected association, depending on the value of the <b>Association Direction</b> field)</li> </ul>

Column	Title (enter as comment for guidance)	Description
		Not all of these work with all connector types; for example, you cannot create a bi-directional Generalization.
<b>K</b>	<b>New Link Caption</b>	Defines the text to display in the Quick Linker menu if a new connector is being created but not a new element.
<b>L</b>	<b>New Link &amp; Element Caption</b>	Defines the text to display in the Quick Linker menu if a new connector AND a new element are being created.
<b>M</b>	<b>Create Link</b>	If set to <b>TRUE</b> , results in the creation of a new connector; leave blank to stop the creation of a connector.
<b>N</b>	<b>Create Element</b>	If set to <b>TRUE</b> and a connector is being dragged onto an empty space on the diagram, results in the creation of a new element.  Leave blank to stop the element from being created. This overrides the values of <b>Target Element Type</b> and <b>Target Stereotype Filter</b> .
<b>O</b>	<b>Disallow Self connector</b>	Set to <b>TRUE</b> if self connectors are invalid for this kind of connector; otherwise leave this field blank.
<b>P</b>	<b>Exclusive to ST Filter + No inherit from Metatype</b>	Set to <b>TRUE</b> to indicate that elements of type <b>Source Element Type</b> with the stereotype <b>Source Stereotype Filter</b> do not display the Quick Linker definitions of the equivalent unstereotyped element.
<b>Q</b>	<b>Menu Group</b>	Indicates the name of the submenu in which a menu item is created.
<b>R</b>	<b>Complexity Level</b>	Not implemented, always set to <b>0</b> .
<b>S</b>	<b>Target Must Be Parent</b>	Set to <b>TRUE</b> if the menu item should only appear when dragging from a child element to its parent; for example, from a Port to its containing Class. Otherwise leave this field blank.
<b>T</b>	<b>Embed element</b>	Set to <b>TRUE</b> to embed the element being created in the target element; otherwise leave this field blank.
<b>U</b>	<b>Precedes Separator LEAF</b>	Set to <b>TRUE</b> to add a menu item separator to the Quick Linker menu, underneath this entry; otherwise leave this field blank.
<b>V</b>	<b>Precedes Separator GROUP</b>	Set to <b>TRUE</b> to add a menu item <b>group</b> separator to the Quick

Column	Title (enter as comment for guidance)	Description
		Linker sub-menu; otherwise leave this field blank.
<b>W</b>	<b>Dummy Column</b>	Depending on which spreadsheet application you use, this column might require a value in every cell to force a CSV export to work correctly with trailing blank values.

#### Learn more

- [Quick Linker](#)<sup>[1515]</sup>
- [Quick Linker Example](#)<sup>[1519]</sup>
- [Quick Linker Object Names](#)<sup>[1521]</sup>
- [Built-in Diagram Types](#)<sup>[1572]</sup>

#### 7.3.2.1.9.2 Quick Linker Example

If you want to create a Quick Linker definition, it is easiest to set it up in a spreadsheet with each menu item definition constructed across a row, as in the example below:

	A	B	C	E	F	H	J	K
1	//Source Element Type	Source ST filter	Target Element Type	Diagram Filter	New Element Type	New Link Type	New Link Direction	New Link Caption
2	Class	quick			Component	Dependency	to	
3	Class	quick			Component	Dependency	from	
4	Class	quick	Component			Dependency	to	Dependency to
5	Class	quick	Component			Dependency	from	Dependency from
6	Class	quick	Port			Dependency	to	Dependency to
7	Class	quick	Port			Dependency	from	Dependency from
8	Class	quick	Component		Port	Dependency	to	
9	Class	quick	Component		Port	Dependency	from	

	L	M	N	O	P	Q	R	S	T	U	V
1	New Link & Element Caption	Create Link	Create Element	Disallow Self connector	No inherit from metatype	Menu Group	Complexity Level	Target Must Be Parent	Embed element	Preceeds Separator LEAF	Preceeds Separator GROUP
2	Dependency to	TRUE	TRUE	TRUE	TRUE	Component	0				
3	Dependency from	TRUE	TRUE	TRUE	TRUE	Component	0			TRUE	
4		TRUE		TRUE	TRUE		0				
5		TRUE		TRUE	TRUE		0			TRUE	
6		TRUE		TRUE	TRUE		0				
7		TRUE		TRUE	TRUE		0			TRUE	
8	Dependency to	TRUE	TRUE	TRUE	TRUE	Port	0		TRUE		
9	Dependency from	TRUE	TRUE	TRUE	TRUE	Port	0		TRUE	TRUE	

The first row of the example is a comment line identifying the column headings. The subsequent lines define the connector/target element options for a Class element with the stereotype «*quick*». When a connector is dragged **away** from an element of this type, you want the user to create a Dependency either to or from a Component element. When they drag a connector onto an **existing** Port or Component element, you want a Dependency either to or from the Component **or**, in the case of a Component, you want the user to be able to create an embedded Port element.

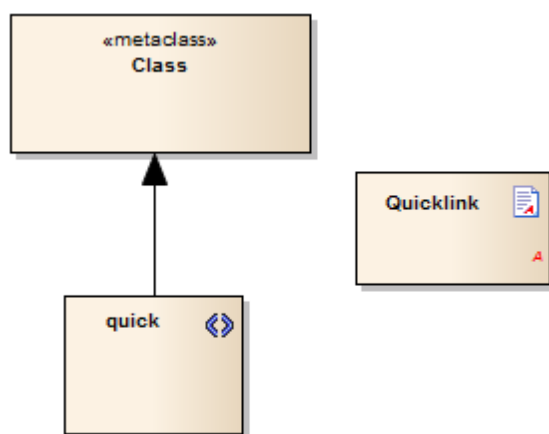
These requirements are defined in eight records in the Quick Linker definition file:

1. Dependency to new Component
2. Dependency from new Component
3. Dependency to existing Component
4. Dependency from existing Component
5. Dependency to existing Port
6. Dependency from existing Port
7. Dependency to existing Component, create new Port
8. Dependency from existing Component, create new Port

The records save to the following CSV file:

```
Cl ass, qui ck, , , , Component , , Dependency , , to , , Dependency to, TRUE, TRUE, TRUE,
TRUE, Component , 0, , , , ,
Cl ass, qui ck, , , , Component , , Dependency , , from , , Dependency from, TRUE, TRUE,
TRUE, TRUE, Component , 0, , , TRUE, ,
Cl ass, qui ck, Component , , , , , Dependency , , to, Dependency to, , TRUE, , TRUE,
TRUE, , 0, , , , ,
Cl ass, qui ck, Component , , , , , Dependency , , from, Dependency from, , TRUE, , TRUE,
TRUE, , 0, , , TRUE, ,
Cl ass, qui ck, Port , , , , , Dependency , , to, Dependency to, , TRUE, , TRUE,
TRUE, , 0, , , , ,
Cl ass, qui ck, Port , , , , , Dependency , , from, Dependency from, , TRUE, , TRUE,
TRUE, , 0, , , TRUE, ,
Cl ass, qui ck, Component , , , Port , , Dependency , , to, , Dependency to, TRUE, TRUE,
TRUE, TRUE, Port , 0, , TRUE, , ,
Cl ass, qui ck, Component , , , Port , , Dependency , , from , , Dependency from, TRUE,
TRUE, TRUE, TRUE, Port , 0, , TRUE, TRUE, ,
```

If you want to test the effect, you can create the following Profile and cut and paste the above CSV lines into the **QuickLink** Document Artifact element.



[Learn more](#)



- [Quick Linker Definition Format](#)<sup>[1516]</sup>
- [Add Quick Linker Definition To Profile](#)<sup>[1523]</sup>

#### 7.3.2.1.9.3 Hide Default Quick Linker Settings

If you create your own Quick Linker definition for an element, you might want to hide the default Quick Linker options between the given source and target elements. You can do this by setting the *Exclusive to stereotype* flag (column **P**) to **TRUE**, in the definition CSV file.

Alternatively, you might want to hide the default Quick Linker options without having a replacement custom option. For example, normally if you don't define any Quick Links for one «quick» Class to another «quick» Class, the Quick Linker arrow displays the default Quick Links for one Class to another Class. To override this behaviour, create a Quick Linker definition in which you set the:

- Source Element Type (column **A**)
- Source Stereotype Filter (column **B**)
- Target Element Type (column **C**)
- Target Stereotype Filter (column **D**)
- New Link Type (column **H**) to **<none>**
- Exclusive to ST Filter + No inherit from Metatype (column **P**) to **TRUE**

Try adding this line to the *Quick Linker Example*:

```
Cl ass, qui ck, I nt er f a c e, , , , , <none>, , , , , , TRUE, , 0, , , , ,
```

With this line in the definition, when a Quick Link is dragged from a «quick» Class to an Interface element, the default Class-to-Interface Quick Links are hidden.

#### Notes

- This technique does not affect the automatic appearance of Dependency, Trace, Information Flow and Help options on the Quick Linker menu

#### Learn more

- [Quick Linker Example](#)<sup>[1519]</sup>
- [Quick Linker Definition Format](#)<sup>[1516]</sup>

#### 7.3.2.1.9.4 Quick Linker Object Names

When you create a Quick Linker definition file, you use a range of base element and connector types to identify the:

- Source element type (column **A**)
- Target element type (column **C**)
- New element type (column **F**) and
- New link type (column **H**)

These are then qualified by the stereotypes you specify in the definition. The base element and connector types you can use are listed below.

**Element Types**

Action	ExecutionEnvironment	ObjectNode
ActionPin	ExitPoint	Package
Activity	ExitState	Part
ActivityParameter	ExpansionNode	Port
ActivityPartition	ExpansionRegion	PrimitiveType
Actor	Feature	ProvidedInterface
Artifact	FinalActivity	Receive
Boundary	GUIElement	RequiredInterface
CentralBufferNode	HistoryState	Requirement
Change	InformationItem	Screen
ChoiceState	InitialActivity	Send
Class	InitialState	Sequence
Collaboration	InteractionOccurrence	Signal
Component	Interface	State
DataType	Issue	StateLifeline
Decision	InterruptableActivityRegion	StateMachine
DeepHistoryState	JunctionState	Synchronization_H
Deployment Specification	MergeNode	Synchronization_V
Device	MessageEndpoint	SynchState
DiagramGate	n-ary Association	UMLDiagram
EntryPoint	Node	UseCase
EntryState	Object	ValueLifeline

**Connector Types**

Abstraction	Deployment	Realization
Aggregation	Extension	Redefinition
Association	Generalization	Sequence
AssociationClass	InformationFlow	StateFlow
CommunicationPath	InterfaceLink	Substitution
Composition	Manifest	TemplateBinding
ConnectorLink	Nesting	UCExtends
ControlFlow	ObjectFlow	UCIncludes
DelegateLink	PackageImport	Usage
Dependency	PackageMerge	UseCase

### 7.3.2.1.9.5 Add Quick Linker Definition To Profile

When you have set up your Profile Quick Linker definitions as a CSV file, you can incorporate them into the Profile. To do this, you copy the file contents into the Linked Document of a **Document Artifact** element that exists in the same diagram as the Stereotype elements of the Profile.

#### Add Definition to Profile

Step	Discussion	See also
1	Open the Profile child diagram containing the Stereotype elements for the Profile.	<a href="#">Create UML Profiles</a> <sup>[1485]</sup>
2	Select the Documentation page of the Diagram Toolbox ( <b>More tools   Documentation</b> ), and drag a Document Artifact element onto the diagram. Give this element the name <b>QuickLink</b> .	<a href="#">Diagram Toolbox</a> <sup>[792]</sup> <a href="#">Document Artifact</a> <sup>[1373]</sup>
3	Double-click on the element to open the Linked Document Editor; cancel the prompt for a template name.	
4	Open your CSV file in a text editor such as Notepad and copy and paste the contents into the Document Artifact element Linked Document.  Save and close the document.	<a href="#">Edit Linked Documents</a> <sup>[1048]</sup>
5	Continue working on the Profile until it is complete, and save it.  The QuickLink definitions are saved with the Profile and are processed and applied when the Profile is imported (within its MDG Technology) into another model.  A technology can contain a number of Profiles and therefore have a number of Quick Link definitions, one for each Profile.	

#### Learn more

- [Quick Linker Definition Format](#) <sup>[1518]</sup>
- [Quick Linker Example](#) <sup>[1519]</sup>

### 7.3.2.2 Export a Profile

Once you have created a Profile, defined the Stereotype elements, and added any Tagged Values, Shape Scripts, Constraints and Quick Linker definitions you need, you can save (export) the Profile to disk. The Profile can then be integrated with an MDG Technology and deployed to other models for use.

#### Save a Profile

Step	Description	See also
1	<p>If your Profile is:</p> <ul style="list-style-type: none"> <li>• A <b>single Profile</b> spread over <b>multiple diagrams</b> within the same Profile Package, find the Profile Package in the Project Browser window, right-click on it and select the <b>Advanced   Save Package as UML Profile</b> context menu option</li> <li>• One of <b>multiple</b> Profiles within the <b>same</b> Profile Package, right-click anywhere in the background of the Profile diagram and select the <b>Save as Profile</b> context menu option</li> <li>• A <b>single diagram</b> within the Profile Package, choose either the <b>Save Package as UML Profile</b> context menu option or the <b>Save as Profile</b> context menu option</li> </ul> <p>The Save UML Profile dialog displays.</p>	<a href="#">Save Profile Options</a> <sup>[1525]</sup>
2	<p>Click on the ( ... ) (Browse) button, and select the destination directory path for the XML Profile file.</p> <p>If necessary, edit the Profile filename, but do not delete the .xml extension.</p>	
3	<p>In the <b>Profile Type</b> field, use the default value <b>EA (UML)2.X</b> (or, if necessary, click on the drop-down arrow and select this value).</p>	
4	<p>Set the required export options for all stereotypes defined in the Profile:</p> <ul style="list-style-type: none"> <li>• <b>Element Size</b> - select the checkbox to export the element size attributes</li> <li>• <b>Color and Appearance</b> - select the checkbox to export the color (background, border and font) and appearance (border thickness) attributes</li> <li>• <b>Alternate Image</b> - select the checkbox to export the metafile images</li> <li>• <b>Code Templates</b> - select the checkbox to export the code templates, if they exist</li> </ul>	<a href="#">Color and Appearance</a> <sup>[1502]</sup>
5	<p>Click on the <b>Save</b> button to save the Profile to disk.</p>	

### Avoiding Profile Name and ID conflicts

Each Profile should have a unique name and ID. The Profile name is specified when saving the Profile, while the ID is derived from the GUID of the diagram or Package that was used to save the Profile. To avoid name and ID conflicts:

- When creating multiple Profiles, use a new diagram or Package for each Profile
- When saving Profiles enter a Profile name that is unique

On starting Enterprise Architect or enabling an MDG Technology, if a duplicate Profile name or duplicate Profile ID is detected, a warning will be displayed in the System Output window.

### Notes

- To quickly test a Profile, you can import the XML file on its own into the Resources window; for final

deployment, incorporate the Profile into an MDG Technology

### Learn more

- [Add a Profile](#)<sup>[1548]</sup> (to an MDG Technology)
- [UML Profiles in the Resources Window](#)<sup>[1526]</sup>

### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Modeling Languages | Build a Profile | Save UML Profile**

#### 7.3.2.2.1 Save Profile Options

When you save a Profile, you can save it either from its parent Package or from the Profile diagram, depending on whether the Profile is:

- A single Profile spread over multiple diagrams within the same Profile Package (find the Profile *Package* in the Project Browser, right-click on it and select the **Advanced | Save Package as UML Profile** context menu option), which is typically the case for a Stereotypes Profile
- One of multiple Profiles within the same Profile Package (right-click anywhere in the background of the Profile *diagram* and select the **Save as Profile** context menu option); for example, when creating multiple Toolbox profiles
- A single diagram within the Profile package (choose *either* the **Save Package as UML Profile** context menu option *or* the **Save as Profile** context menu option)

The two context menu options produce slightly different results, which you might consider when deciding how to create your Profile, especially in the third instance where you could choose either option.

Save From Diagram	Save From Package	Notes
The Profile takes the diagram name.	The Profile takes the Package name.	Package and diagram names are not necessarily the same, although you can save a lot of confusion if you make them the same or very similar.  For example: Package <i>GL</i> with diagrams <i>GL 1</i> , <i>GL2</i> , <i>GL3</i> .
The Profile takes the diagram's notes.	The Profile takes the Package's notes.	Diagram notes can be significant in the Profile definition, such as for Toolbox Profiles.  See <a href="#">Create Toolbox Profiles</a> <sup>[1567]</sup> .
You can take the default size and appearance (including alternate image) from the diagram object.	You cannot take the default size and appearance from the diagram object.  You can use the <code>_sizeX</code> , <code>_sizeY</code> and <code>_image</code> properties, but there is no equivalent for default colors.	

Save From Diagram	Save From Package	Notes
This option can be much faster.	This option can be much slower.	<p>The difference arises because diagram objects are kept in memory and Project Browser elements are not.</p> <p>This is only likely to be an issue if the Profile is a large one and you are using a slow network connection to a remote repository.</p>

### 7.3.2.3 UML Profiles in the Resources Window

The Resources window contains a tree structure with entries for a range of items including UML Profiles. The *UML Profiles* node initially contains no entries; to be able to use Profiles from the Resources window you must import them into the project from external XML files.

Items in a Profile represent stereotypes. These can be applied to UML elements in the following ways:

- Stereotypes that apply to elements such as *Classes* and *interfaces* can be dragged directly from the Resources window to the current diagram, automatically creating a stereotyped element; alternatively, they can be dragged onto existing elements, automatically applying them to the element
- Stereotypes that apply to *attributes* can be drag-and-dropped onto a host element (such as a Class); a stereotyped attribute is automatically added to the element's feature list
- Stereotypes that apply to *operations* are like those that apply to attributes; drag-and-drop onto a host element to add the stereotyped operation
- Stereotypes that apply to *connectors* such as *associations*, *generalizations*, *messages* and *dependencies* are added by selecting them in the Resources window, then clicking on the start element in a diagram and dragging to the end element (in the same manner as adding normal connectors); a stereotyped connector is added
- Stereotypes that apply to *association ends* can be added by dragging the connector end element over the end of an Association in the diagram

#### Learn more


- [Import UML Profiles Into the Resources Window](#) 

#### 7.3.2.3.1 Import UML Profiles Into the Resources Window

Profiles exist as XML files, which can be imported into any project to provide tailored modeling structures for specific domains. A number of Profile XML files are available to you on the Sparx Systems website, for importing into your models. You can also import Profile XML files that you have created yourself. If a Profile includes references to any metafiles, copy these metafiles into the same directory as the Profile XML file.

**Access** **Project | Resources (Alt+6) > UML Profiles | Import Profile**

#### Import a Profile

Field/Button	Action	See also
<b>Filename</b>	Click on the ( ... ) (Browse) button and locate the XML Profile file to import.	<a href="#">Resources</a> 

Field/Button	Action	See also
<b>Element Size</b>	Select the checkbox to import the element size attributes for all stereotypes defined in the Profile.	
<b>Color and Appearance</b>	Select the checkbox to import the color (background, border and font) and appearance (border thickness) attributes for all stereotypes defined in the Profile.	
<b>Alternate Image</b>	Select the checkbox to import the metafile image for all stereotypes defined in the Profile.	
<b>Code Templates</b>	Select the checkbox to import the code templates, if they exist, for all stereotypes defined in the Profile.	
<b>Overwrite Existing Templates</b>	Select the checkbox to overwrite any existing code templates defined in the current project, for all stereotypes defined in the Profile.	
<b>Import</b>	<p>Click on this button to add the Profile to the <i>UML Profiles</i> folder.</p> <p>If the Profile already exists, a prompt displays for you to overwrite the existing version and import the new one.</p> <p>When the import is complete, the Profile is ready to use.</p>	<a href="#">Add Profile Objects to a Diagram</a> <sup>[1472]</sup>

#### Learn more

- [Creating MDG Technologies](#) <sup>[1545]</sup>
- [Deploying MDG Technologies](#) <sup>[1580]</sup>

### 7.3.3 MDG Technologies - Creating

If you want to access and use resources pertaining to a specific technology within Enterprise Architect, you can do so using a Model Driven Generation (MDG) Technology. There are various options for an administrator or individual user to bring existing MDG Technologies into use with Enterprise Architect. Technology Developers can also develop **new** MDG Technologies and deploy them to the project team as necessary, providing a solution tailored to your working domain or environment.

#### Learn more - General

- [MDG Technologies](#) <sup>[1475]</sup>
- [Manage MDG Technologies](#) <sup>[1477]</sup>
- [Defining a Modeling Language](#) <sup>[1483]</sup>

#### Learn more - Developing Technologies

- [Create MDG Technologies](#) <sup>[1545]</sup>

- [Working With MTS Files](#) <sup>[1559]</sup>
- [Developing Profiles](#) <sup>[1485]</sup>
- [Customize Toolbox Profiles](#) <sup>[1560]</sup>
- [Create Custom Diagram Profiles](#) <sup>[1570]</sup>
- [Add Import/Export Scripts](#) <sup>[1578]</sup>
- [Define Validation Configuration](#) <sup>[1578]</sup>
- [Incorporate Model Templates](#) <sup>[1576]</sup>
- [Set Up Technology Element Images](#) <sup>[1574]</sup>
- [Deploy an MDG Technology](#) <sup>[1580]</sup>
- [Enterprise Architecture Framework Design with Sparx Systems Enterprise Architect](#) (an example of creating an MDG Technology for an Enterprise Architecture framework)

### 7.3.3.1 Using the Profile Helpers

MDG Technologies and Profiles are developed using diagrams and elements within Enterprise Architect. These diagrams and elements use specific attributes and properties which determine the content and behavior of the resulting MDG Technology. **Profile Helpers** assist in creating new MDG Technologies, and the following Profile types:

- Stereotype Profiles
- Toolbox Profiles and
- Diagram Profiles

The Profile Helpers consist of two components:






- **MDG Technology Builder** templates in the **Model Wizard** dialog which provide a starting point for creating a new MDG Technology
- **Profile Helper** items in the **Profile Toolbox** which provide dialogs that simplify the creation of Stereotype, Toolbox and Diagram Profiles

**Access**    **Right-click on a Project Browser Package node | Add a Model using Wizard (Ctrl+Shift+M)**

#### Create a new MDG Technology

Step	Description	See also
1	In the Model Wizard dialog, select <b>MDG Technology Builder</b> from the <b>Technology</b> list and tick the <b>Basic Template</b> checkbox in the <b>Name</b> list. Click the <b>OK</b> button. A prompt displays for the Technology name.	<a href="#">Model Wizard</a> <sup>[753]</sup>
2	Enter a name for your new MDG Technology, and click on the <b>OK</b> button.  This will create a basic template of Packages and example elements, which can be used as a starting point for creating an MDG Technology. The template includes three Packages, each having the same name as the technology but a different stereotype corresponding to the type of Profile they define: <ul style="list-style-type: none"> <li>• &lt;&lt;profile&gt;&gt; - Package for defining a Profile containing the Stereotypes</li> </ul>	



Step	Description	See also
	<p>users will apply to elements</p> <ul style="list-style-type: none"> <li>• <code>&lt;&lt;diagram profile&gt;&gt;</code> - Package for a Profile describing the diagram types users will create</li> <li>• <code>&lt;&lt;toolbox profile&gt;&gt;</code> - Package for a Profile describing the elements to show in a toolbox</li> </ul>	
3	<p>Within each Package, open the diagram and, referring to the sample elements provided, add additional items to the Profile.</p> <p>The <b>Profile Toolbox</b> contains a page of Profile Helper icons that, when dragged onto the diagram, help you create and populate the elements of the various Profiles.</p>	<p><a href="#">Create Stereotype Profiles using Profile Helpers</a>  <sup>[1529]</sup></p> <p><a href="#">Create Diagram Profiles using the Profile Helpers</a>  <sup>[1536]</sup></p> <p><a href="#">Create Toolbox Profiles using the Profile Helpers</a>  <sup>[1538]</sup></p>
4	Save each of these Profiles to disk.	<a href="#">Export a Profile</a>  <sup>[1523]</sup>
5	Incorporate the saved Profiles into an MDG Technology.	<a href="#">Create MDG Technology File</a>  <sup>[1545]</sup>

#### Learn more

- [Using MDG Technologies](#)  <sup>[1475]</sup>
- [Profile Toolbox](#)  <sup>[810]</sup>
- [Rapid Technology Development with Profile Helpers & Model Scripts](#) (Online Resource)

#### 7.3.3.1.1 Create Stereotype Profiles using Profile Helpers

When creating a technology to provide a domain-specific toolset, the typical starting point is to define each element, connector, feature and structural component you want to provide. These are defined by a **Profile**.

All **Stereotypes** defined in a Profile are either extensions of **Core UML objects** (Metaclasses) defined by Enterprise Architect, or extensions of **non-UML objects** (Stereotypes) defined by other existing Profiles and technologies.

When development of a Profile is complete, it is saved to an external XML file and then incorporated into an MDG Technology for final deployment.

Each Stereotype defined in a Profile modifies the behavior of the Metaclass or Stereotype that it extends. These modifications might include:

- Tagged Values to provide additional properties
- Constraints to define the conditions and rules that apply to each Stereotype
- A Shape Script to customize the overall appearance of the new object
- A change to the default appearance of the object, such as background, border and font colors

- Quick Linker definitions to provide a list of the most common connection types from each Stereotype
- Special attributes that define the specific appearance and behavior of the new object, including the initial element size and Project Browser icon

### Create a UML Profile

Step	Description	See also
1	<p>In the Project Browser, locate the Package with the &lt;&lt;profile&gt;&gt; stereotype and open its child diagram.</p> <p>If you do not have an existing &lt;&lt;profile&gt;&gt; Package, use the <b>MDG Technology Builder</b> option in the <b>Model Wizard</b> dialog to create a new technology, then open the diagram from the newly created &lt;&lt;profile&gt;&gt; Package.</p>	<a href="#">Using the Profile Helpers</a> [1528]
2	<p>(Optional) If you intend your Stereotype elements to include <b>Tagged Values</b> with a drop-down list of several pre-defined values, each set of values must be defined by an <b>Enumeration</b> element on the Profile diagram.</p> <p>If you intend your Stereotype elements to include a <b>Structured Tagged Value</b> to provide a composite set of information, each structure must be defined by a <b>Class</b> element on the Profile diagram.</p> <p>The Enumeration and Class elements have to exist before you can define these Tagged Value types for your Stereotype; you can either create the elements at this point, or add these Tagged Values to your Stereotype at a later time.</p>	<a href="#">Add an Enumeration to a Stereotype</a> [1493]  <a href="#">Add a Structured Tagged Value to a Stereotype</a> [1495]
3	Add a new Stereotype by dragging the <b>Add Stereotype</b> Profile Helper from the Diagram Toolbox. The dialog opened by the <b>Add Stereotype</b> Profile helper will allow you to specify various general Properties, Tagged Values, and the Shape Script for your Stereotype.	<a href="#">Add Stereotypes and Metaclasses using Profile Helpers</a> [1531]
4	(Optional) Define <b>Constraints</b> for the Stereotype.	<a href="#">Define Stereotype Constraints</a> [1500]
5	(Optional) Set the <b>Default Appearance</b> for the Stereotype.	<a href="#">Set Default Appearance</a> [1502]
6	Repeat steps 3 to 5 for each new Stereotype element you want to create.	
7	(Optional) Add a <b>Quick Linker Definition</b> to the Profile.	<a href="#">Add Quick Linker Definition To Profile</a> [1523]
8	<p>Save the Package as a Profile.</p> <p>When saving the Profile, the name used should match the name of the</p>	<a href="#">Export a Profile</a> [1523]

Step	Description	See also
	Profile Package; this is necessary for the references within a Toolbox profile to function correctly	
9	Incorporate the Profile into an MDG Technology.	<a href="#">Generate MDG Technology</a> <sup>[1545]</sup> <a href="#">Add a UML Profile</a> <sup>[1548]</sup>

### Notes

- A Profile Package cannot contain other Packages; do not add any other Packages to the Profile

### Learn more

- [Create Diagram Profiles using the Profile Helpers](#) <sup>[1536]</sup>
- [Create Toolbox Profiles using the Profile Helpers](#) <sup>[1538]</sup>

#### 7.3.3.1.1.1 Add Stereotypes and Metaclasses using Profile Helpers

You can define Stereotypes in a Profile to either extend:

- **Core UML objects (Metaclasses)** pre-defined in Enterprise Architect), or
- **Objects (Stereotypes)** defined by other Profiles and technologies (for instance objects defined in ArchiMate or SysML)

Stereotypes can extend Metaclasses in several ways:

- **One** Stereotype extending **one** Metaclass, for a specific definition of one object type
- **One** Stereotype extending **more than one** Metaclass, where the definition applies to more than one object type - such as modifying both a Class and an Object in the same way
- **Several** Stereotypes extending **one** Metaclass, where you are creating several variations of the same base object type; for example, to define types of Association connector, representing Parent, Sibling, Grandparent, Uncle/Aunt and Cousin relationships

### Add Metaclasses and Stereotypes to a Profile

Step	Description	See also
1	If you are extending a <b>non-UML</b> type defined by an existing Profile or technology, follow the process described in the <i>Create Stereotypes Extending non-UML Objects</i> Help topic.	<a href="#">Create Stereotypes Extending non-UML Objects</a> <sup>[1491]</sup>
2	In the Project Browser, locate the Package with the <<profile>> Stereotype and open its child diagram.	<a href="#">Create UML Profiles using Profile Helpers</a> <sup>[1529]</sup>

3	<p>Drag the <b>Add Stereotype</b> icon from the Profile Helpers page of the Diagram Toolbox onto the diagram.</p> <p>The Add Stereotype dialog displays.</p>	
4	<p>In the <b>Name</b> field, type the Stereotype name (which will also be the name of the new modeling object).</p>	
5	<p>Select one of the following object groups by clicking on the <b>Type</b> drop-down arrow:</p> <ul style="list-style-type: none"> <li>• <b>Element Extension</b> - to create a Stereotype that extends an element</li> <li>• <b>Connector Extension</b> - to create a Stereotype that extends a connector</li> <li>• <b>Abstract Metaclass</b> - to create a Stereotype that extends a structural or behavioral modifier</li> <li>• <b>Metaclass Extension</b> - to create a Stereotype that extends a Metaclass that already exists within your model (and most likely within the diagram you are currently working in)</li> </ul>	
6	<p>Click on the <b>Add Metaclass</b> button.</p> <p>The Extend Metaclass dialog displays, showing a list of object types associated with the <b>object group</b> selected in step 5.</p> <p>Select the Metaclass to be extended from the list and click on the <b>OK</b> button.</p> <p>If you selected <b>Metaclass Extension</b> in step 5, the Select a Profile Element browser/search dialog displays; search for and select the existing Metaclass element to extend with this Stereotype.</p> <p>The Metaclass name is added to the <b>Extensions</b> field.</p>	
7	<p>If you want to extend more than one Metaclass with the Stereotype, click on the <b>Add Metaclass</b> button again and select the next object type to extend. You can repeat this for as many Metaclasses as you want to extend with this Stereotype.</p> <p>To delete a selected Metaclass from the <b>Extensions</b> list click on the <b>Remove</b> button.</p>	
8	<p>Review the available Properties in the <b>Stereotype</b> panel. These properties modify the behaviour of the Stereotype.</p> <p>To apply a property, click in the <b>Value</b> field and type or select the appropriate value.</p> <p>When you select a property field, a description of the property's effect is displayed at the bottom of the <b>Stereotype</b> panel.</p> <p>Only provide values for properties that you want to apply to this Stereotype.</p>	
9	<p>Click on the name of a Metaclass in the <b>Extensions</b> field and review the</p>	

	<p>available properties in the <b>Metaclass</b> panel. These properties further modify the behaviour of the stereotype based on options specific to the Metaclass being extended.</p> <p>To apply a property, click in the <b>Value</b> field and type or select the appropriate value.</p> <p>When you select a property field, a description of the property's effect is displayed at the bottom of the <b>Metaclass</b> panel.</p> <p>Do not provide values for any properties that you do not want to apply to this Stereotype.</p> <p>If you are extending more than one Metaclass, click on the next Metaclass name in the <b>Extensions</b> field and review the properties for that object type.</p>	
10	Click on the <b>Next</b> button. The Define Tagged Values page displays.	
11	<p>In the <b>Property</b> panel right-click to display a context menu with options for creating and grouping Tagged Values of different types. These options include:</p> <ul style="list-style-type: none"> <li>• <b>Add Tagged Value:</b> Create a simple Tagged Value - a prompt displays for the Tagged Value name. Add a name and click on the <b>OK</b> button. This displays in the name in the <b>Property</b> column; to set a default value, type it in to the <b>Default Value</b> field</li> <li>• <b>Add Specialized Tagged Value :</b> <ul style="list-style-type: none"> <li>• <b>Enumeration:</b> create an enumeration Tagged Value, based on an existing <b>Enumeration</b> element</li> <li>• <b>Predefined:</b> select a Predefined Tagged Value Type from a list and, in the <b>Default Value</b> field, type or select an initial value if necessary</li> <li>• <b>Structured:</b> create a Structured Tagged Value composed of several other Tagged Values, typed by an existing <b>Class</b> element</li> <li>• <b>Reference:</b> create a Tagged Value with which the user can locate and reference an element created with a specified <b>Stereotype</b> (a form of <b>RefGUID</b> Tagged Value); in creating this, you must select the existing Stereotype element that defines the stereotype</li> <li>• <b>Reference List:</b> create a Tagged Value with which the user can locate and reference a <b>list</b> of elements created with a specified <b>Stereotype</b> (a form of <b>RefGUIDList</b> Tagged Value); in creating this, you must select an existing Stereotype element that defines the stereotype</li> </ul> </li> <li>• <b>Edit Tagged Value Name:</b> displays a simple prompt in which you overtype the current name to correct or change it</li> <li>• <b>Create Tag Group:</b> create Tag Groups in the <b>Metaclass</b> element, through which to organize the Tagged Values you have created in the Stereotype element</li> <li>• <b>Move Tag to Group</b> (displayed when you right-click on an existing Tagged Value): displays the Move Tag to Group dialog, on which you can select an existing Tag Group to contain the selected Tagged Value</li> </ul>	<p><a href="#">Add an Enumeration to a Stereotype</a> <sup>[1493]</sup></p> <p><a href="#">Create Tagged Value Type from Predefined Types</a> <sup>[1622]</sup></p> <p><a href="#">Add a Structured Tagged Value to a Stereotype</a> <sup>[1495]</sup></p> <p><a href="#">RefGUID Tagged Value</a> <sup>[1624]</sup></p> <p><a href="#">RefGUIDList Tagged Value</a> <sup>[1625]</sup></p> <p><a href="#">Define Tag Groupings</a> <sup>[1514]</sup></p>

	<ul style="list-style-type: none"> <li>• <b>Remove Grouping</b>: remove the selected Tag Group, leaving its member Tagged Values listed at the end of the <b>Property</b> column</li> <li>• <b>Delete</b>: Remove the selected Tagged Value from the list and from the Stereotype</li> </ul>	
12	<p>Click on the <b>Next</b> button. The Define a Shape Script page displays.</p> <p>A Shape Script can be used to define the <b>appearance</b> of the Stereotype. To include a Shape Script, click on the <b>Edit</b> button.</p> <p>The Shape Editor window displays. Create your Shape Script using this editor.</p> <p>When you have finished creating the Script, click on the <b>OK</b> button. The image defined by the Shape Script is shown in the <b>Preview</b> panel.</p> <p>Note: For the Shape Script to take effect, you must select the <b>Alternate Image</b> option when you save the Profile.</p> <p>Alternatively, you can define a simple default appearance (background color, line color) for the model object, after you have created the Stereotype element.</p>	<a href="#">Shape Scripts</a> <sup>[1582]</sup>  <a href="#">Shape Editor</a> <sup>[1584]</sup>  <a href="#">Export a Profile</a> <sup>[1523]</sup>
13	Click on the <b>Finish</b> button. The Stereotype element and Metaclass element(s) are now displayed on the Profile Diagram.	
14	<p>You can now:</p> <ul style="list-style-type: none"> <li>• Repeat steps 2 to 13 for each of the other Stereotype elements you want to create</li> <li>• Edit the Stereotype (and through it, the Metaclass) element properties you defined above, using the Profile Helper</li> <li>• Add Constraints to your Stereotype element</li> <li>• If a shape has not been set then you can now define the object's default appearance (background color, line color)</li> <li>• Set up the Quick Linker definitions for the stereotyped elements and connectors in the Profile</li> </ul>	<a href="#">Edit a Stereotype Element</a> <sup>[1535]</sup> <a href="#">Define Stereotype Constraints</a> <sup>[1500]</sup> <a href="#">Set Default Appearance</a> <sup>[1502]</sup> <a href="#">Quick Linker</a> <sup>[1515]</sup>

### Notes

- If you intend to extend a large number of model elements, rather than putting all of them on one diagram you can create additional child Class diagrams under the <<profile>> Package and add different types of Metaclass elements to different diagrams; in this case you save the **Package** as the Profile, not the individual **diagrams**
- Stereotype elements must have unique names, but Metaclass elements can have the same name (for example, there can be several *Action* Metaclasses, each with a different *ActionKind* attribute)
- If you have a number of Tagged Values in the Stereotype element, and you have assigned them to groups, you can define which of those groups default to expanded (open) in the Tagged Values window, and which default to closed; open the Attributes dialog for the metaclass and add the attribute **\_tagGroupStates** with the initial value `<groupname>=closed;<groupname>=closed;<groupname>=open;...`

### Learn more

- [Using the Profile Helpers](#) <sup>[1528]</sup>
- [Export a Profile](#) <sup>[1529]</sup>

#### 7.3.3.1.1.2 Edit a Stereotype Element

If you want to add to or correct the properties of a Stereotype or Metaclass element in a Profile, you can edit it using the standard facilities such as the element Properties dialogs and Tagged Values window. However, you can also update the Stereotype element through the **Profile Helper Stereotype Properties dialog** and, through the Stereotype, also update the Metaclass elements that the Stereotype extends.

Any changes you have made to the elements by other means, such as through the element Properties dialog, are reflected in the contents of the Profile Helper.

**Access**    **Right-click on Stereotype element | Edit with Profile Helper**

##### Edit the Stereotype element

Step	Description	See also
1	<p>The Stereotype Properties dialog defaults to the General tab. On this tab you can:</p> <ul style="list-style-type: none"> <li>• Change the Stereotype element name</li> <li>• Add further Metaclass elements to be extended by this Stereotype element</li> <li>• Add or change values for the attributes of the Stereotype element</li> <li>• Add or change values for the attributes of each Metaclass element</li> </ul>	<a href="#">Add Stereotypes and Metaclasses</a> <sup>[1534]</sup>
2	<p>Click on the Tagged Values tab. On this tab you can:</p> <ul style="list-style-type: none"> <li>• Edit the default value of a tag</li> <li>• Add a new tag of one of a range of types</li> <li>• Create a tag group</li> <li>• Assign or reassign a tag to a group</li> <li>• Remove a tag group</li> <li>• Delete a Tagged Value from the Stereotype</li> </ul>	
3	<p>Click on the Shape Script tab. On this tab you can:</p> <ul style="list-style-type: none"> <li>• Add a Shape Script (if one does not exist)</li> <li>• Edit the existing Shape Script using the Shape Editor</li> </ul>	
4	<p>When you have finished editing the Stereotype element, click on the <b>OK</b> button.</p> <p>The Profile Class diagram redisplay, with the edited elements showing the changes you have made.</p>	

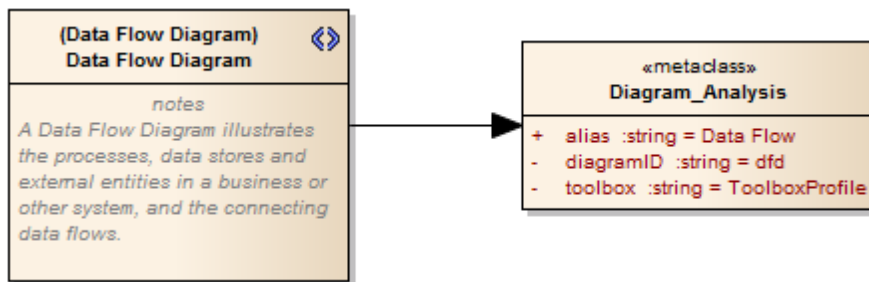
##### Learn more

- [Using the Profile Helpers](#) <sup>[1528]</sup>

- [Create UML Profiles using Profile Helpers](#) <sup>[1529]</sup>
- [Add Stereotypes and Metaclasses using Profile Helpers](#) <sup>[1531]</sup>

### 7.3.3.1.2 Create Diagram Profiles using the Profile Helpers

When you develop an MDG Technology, it is possible to create **extended diagram types** and include them in your MDG Technology as custom **Diagram Profiles**. For example, you might create a DFD Diagram Profile that defines a DFD diagram as an extension of the built-in Analysis diagram, as shown:



The **Add Diagram Extension Profile Helper** can assist you in defining your Diagram Profile, adding the necessary elements and giving them the appropriate attributes to define the functionality of the resulting custom diagram types.

#### Create extended diagram types

Step	Action	See also
1	If you have not done so already, use the Model Wizard's MDG Technology Builder option to create a set of Packages for defining Profiles.  In the Project Browser, locate the Package with the <<diagram profile>> stereotype and open its child diagram.	<a href="#">Using the Profile Helpers</a> <sup>[1528]</sup>
2	Drag the <b>Add Diagram Extension</b> item from the Profile Helpers toolbox page onto the diagram.  The Add Diagram Extension Dialog displays.	
3	In the <b>Name</b> field, type the name for the custom diagram type.	
4	In the <b>Extension Type</b> field click on the drop-down arrow and select the built-in diagram type that the custom diagram type will extend.	
5	In the <b>Description</b> field type a brief description of what the diagram is used for.  When a user selects this diagram type in the New Diagram dialog, this description will be displayed in the bottom right of the dialog.	
6	Within the <b>Properties</b> pane enter values for the following fields: <ul style="list-style-type: none"> <li>• <b>Alias:</b> Defines the diagram type displayed before the word 'Diagram' on</li> </ul>	



Step	Action	See also
	<p>the diagram title bar; for example: Block Diagram.</p> <ul style="list-style-type: none"> <li>• <b>Frame ID:</b> Defines the diagram type that will appear in the diagram frame label</li> <li>• <b>Frame Format String:</b> Enter a string containing substitution macros for defining the frame title, with or without additional delimiters such as ( ); macros that can be used are: <ul style="list-style-type: none"> <li>• #DGMALIAS#</li> <li>• #DG MID#</li> <li>• #DGMNAME#</li> <li>• #DGMNAMEFULL#</li> <li>• #DGMOWNERNAME#</li> <li>• #DGMOWNERNAMEFULL#</li> <li>• #DGMOWNERTYPE#</li> <li>• #DGMSTEREO#</li> <li>• #DG MTYPE#</li> </ul> </li> <li>• <b>Toolbox Profile:</b> Enter the name of the Toolbox Profile (the name entered when saving the profile); this Toolbox will be opened automatically each time a diagram of this type is opened</li> <li>• <b>Swimlanes:</b> Defines swimlanes that will be displayed on the diagram; for example: <pre>Lanes=2; Orientation=Horizontal; Lane1=Title1; Lane2=Title2;</pre> <p>(where <b>Lanes</b> can be any value, but the number of <b>Lane&lt;n&gt;</b> values must equal the value of <b>Lanes</b>; <b>Orientation</b> can be omitted, in which case the swimlanes default to <b>vertical</b>)</p> </li> </ul>	
7	<p>The remaining fields in the <b>Properties</b> pane can be used to customize the diagram's default options. Any Attributes left blank will not be applied.</p> <p>When a user selects a field, a description of the property's effect is displayed at the bottom of the <b>Properties</b> pane.</p>	
8	Click on the <b>OK</b> button. The appropriate Stereotype and Metaclass elements are added to the diagram.	
9	Repeat steps 2 to 8 for each diagram extension to include in the diagram Profile.	
10	Save the diagram as a Profile.	<a href="#">Export a Profile</a> <sup>[1523]</sup>
11	Incorporate the Profile into an MDG Technology.	<a href="#">Create MDG Technology File</a>

Step	Action	See also
		<a href="#">1545</a>

### Notes

- After a diagram extension has been added you can modify its properties again by right clicking the appropriate Stereotype element on the diagram and selecting **Edit with Profile Helper**

### Learn more



- [Using the Profile Helpers](#) [1528](#)
- [Create UML Profiles using Profile Helpers](#) [1529](#)
- [Create Toolbox Profiles using the Profile Helpers](#) [1538](#)

#### 7.3.3.1.3 Create Toolbox Profiles using the Profile Helpers

Within an MDG Technology you can create multiple Toolbox Profiles. Each Toolbox Profile defines a single Toolbox. A Toolbox consists of one or more expandable/collapsible regions, referred to as Toolbox Pages.

#### Create a Toolbox Profile

Step	Action	See also
1	If a group of Packages for defining Profiles has not been created, use the Model Wizard's <b>MDG Technology Builder</b> option to create this group.  In the Project Browser, locate the Package with the <code>&lt;&lt;toolbox profile&gt;&gt;</code> stereotype and open its child diagram. This diagram shows an example Toolbox Profile with a single Toolbox Page.	<a href="#">Using the Profile Helpers</a> <a href="#">1528</a>
2	Drag the <b>Create Custom Toolbox</b> item from the Profile Helpers Toolbox Page onto the diagram.  The Select a Toolbox Profile Package Dialog displays.	
3	Select the Package with the <code>&lt;&lt;toolbox profile&gt;&gt;</code> stereotype referred to in step 1.  Click on the <b>OK</b> button. The Create Toolbox Page dialog displays.	
4	In the <b>Toolbox Name</b> field type the name for your Toolbox.  This is the name that will be displayed for the Toolbox when using the <b>More tools...</b> option in the Diagram Toolbox.	
5	In the <b>Description</b> field type a description for the Toolbox.  This description acts as a default <b>tool tip</b> for your Toolbox, unless you define a specific tool tip for a Toolbox Page as mentioned in step 10.	

6	<p>Click on the <b>OK</b> button.</p> <p>The diagram that you will use to define your Toolbox is created and displayed.</p>	
7	<p>(Optional) When dragging an item from a Toolbox onto a diagram, the item will typically create an element or a connector.</p> <p>It is also possible to have a single Toolbox item that, when dragged onto a diagram, will provide a selection of items to choose from. This is referred to as a <b>hidden sub-menu</b>.</p> <p>If you want your Toolbox to contain one or more hidden sub-menus you should define these before proceeding with the steps on this page.</p>	<a href="#">Create Hidden Sub-Menus using the Profile Helpers</a> 
8	<p>You can now define one or more Toolbox Pages that will appear on the Toolbox.</p> <p>Drag the <b>Add Toolbox Page</b> item from the Profile Helpers Toolbox Page onto the diagram.</p> <p>The Add Toolbox Page dialog displays.</p>	
9	<p>In the <b>Name</b> field, type a name for the Toolbox Page.</p> <p>This is text that will display in the title bar of the corresponding Toolbox Page.</p>	
10	<p>In the <b>Tool Tip</b> field, type the Tool Tip for the corresponding Toolbox Page.</p>	
11	<p>The <b>Icon</b> field in this case will be disabled. This field is only used when defining hidden sub-menu toolboxes.</p>	<a href="#">Create Hidden Sub-Menus using the Profile Helpers</a> 
12	<p>The following options can be used to determine the appearance and functionality of the Toolbox Page. When enabled:</p> <ul style="list-style-type: none"> <li>• <b>Images Only</b>: displays the Toolbox Page without the text labels next to the icons</li> <li>• <b>Is Hidden</b>: defines the Toolbox Page as a hidden sub-menu</li> <li>• <b>Is Common</b>: the Toolbox Page is common to all defined Toolboxes while your technology is active; the page is initially displayed as collapsed</li> <li>• <b>Is Collapsed</b>: the Toolbox Page is initially minimized</li> </ul>	
13	<p>You can now define items to be added to the Toolbox.</p> <p>Click on the down arrow on the right of the <b>Add</b> button. Select one of the following options:</p> <ul style="list-style-type: none"> <li>• <b>Add Stereotype</b>: adds a Toolbox item for a Stereotype that is defined in a UML Profile in the current model; this Profile must be included with the Toolbox Profile in the MDG Technology</li> </ul>	

	<p>After you select this option, the Select a Profile Element dialog displays; use this to select the Stereotype you want to add</p> <ul style="list-style-type: none"> <li>• <b>Add Built in Type:</b> <ul style="list-style-type: none"> <li>• <b>Element:</b> adds a Toolbox item for a UML <b>element type</b> <p>After you select this option the Create new Toolbox Item dialog displays; in the <b>Alias</b> field, type the label to appear on the Toolbox item, and click on the <b>OK</b> button</p> <p>The Select Metaclass dialog then displays; select the UML element type to add to your Toolbox, and click on the <b>OK</b> button</p> </li> <li>• <b>Connector:</b> adds a Toolbox item for a UML <b>connector type</b> <p>After you select this option the Create new Toolbox Item dialog displays; in the <b>Alias</b> field, type the label to appear on the Toolbox item, and click on the <b>OK</b> button</p> <p>The Select Metaclass dialog then displays; select the UML connector type to add to your Toolbox, and click on the <b>OK</b> button</p> </li> <li>• <b>Add Hidden Toolbox:</b> adds a hidden Toolbox sub-menu item; the hidden Toolbox must be defined <b>before</b> you use this option <p>After you select this option, the Create new Toolbox Item dialog displays; in the <b>Alias</b> field, type the label to appear on the Toolbox item and click on the <b>OK</b> button</p> <p>The Select a Hidden Toolbox Stereotype dialog then displays; select the hidden Toolbox to add to your Toolbox, and click on the <b>OK</b> button</p> </li> <li>• <b>Add New Item:</b> adds a Toolbox item with an Alias only <p>This option alone will not create a functional Toolbox item; a Toolbox item added in this way must be later modified via the <b>Toolbox Items</b> list</p> </li> </ul> <p>Clicking on the <b>Add</b> button, and not on the drop-down arrow, is the same as selecting the <b>Add Stereotype</b> option.</p> </li> </ul>	
14	<p>(Optional) Define a Toolbox item that will create an item from an external MDG Technology. For example, adding a Toolbox item that creates a SysML 1.3 Block element.</p> <ol style="list-style-type: none"> <li>1. Click on the down-arrow on the right of the <b>Add</b> button.</li> <li>2. Select the <b>Add New Item</b> option. The Create new Toolbox Item dialog displays.</li> <li>3. In the <b>Alias</b> field, type the label to appear on the Toolbox item, and click on the <b>OK</b> button. The Toolbox item will be added to the <b>Toolbox Items</b> list.</li> <li>4. In the <b>Stereotype</b> field for this Toolbox item, type:</li> </ol>	

	<p><code>Profile : Stereotype( UML : BaseUMLType )</code></p> <p><b>Profile</b> is the name of the Profile that the Stereotype is defined in  <b>Stereotype</b> is the name of the Stereotype/Metatype that this toolbox item will create  <b>BaseUMLType</b> is the base UML type of the non-UML object</p> <p>For example, to include a <b>SysML Block</b> in a Toolbox you would type:</p> <p><code>SysML1.3 : Block( UML : Class )</code></p> <p>5. To identify the <code>Profile : Stereotype</code> string, create an element of the type to include in your Toolbox (for example; a SysML 1.3 Block), then select the element and display the Tagged Values window.</p> <p>Any predefined tags for this element will be grouped under the <code>Profile : Stereotype</code> heading. For example, a SysML 1.3 Block's tags are grouped under <code>SysML1.3 : Block</code>.</p> <p>6. All non-UML objects in Enterprise Architect are an extension of a UML Type. You can reveal an element's base UML type by deleting its Stereotypes. For example, create a SysML 1.3 Block and then, using the Properties Window, delete the Block element's Stereotype. The element type will change from <b>Block</b> to <b>Class</b>.</p>	<p><a href="#">Modify Tagged Values</a> <sup>[1138]</sup></p> <p><a href="#">Elements Used in Toolboxes</a> <sup>[1567]</sup></p> <p><a href="#">Connectors Used in Toolboxes</a> <sup>[1569]</sup></p>
15	<p>(Optional) Create a Toolbox item that will drop a Pattern onto a diagram.</p> <ol style="list-style-type: none"> <li>Click on the down arrow on the right of the <b>Add</b> button.</li> <li>Select the <b>Add New Item</b> option. The Create new Toolbox Item dialog displays.</li> <li>In the <b>Alias</b> field, type the label to appear on the Toolbox item, then click on the <b>OK</b> button.</li> <li>The Toolbox item will be added to the <b>Toolbox Items</b> list.</li> <li>In the <b>Stereotype</b> field for this Toolbox item, type: <p><code>TechnologyID : PatternName( UMLPattern )</code></p> <p><b>TechnologyID</b> is the ID of the Technology, as entered in the MDG Technology Creation Wizard  <b>PatternName</b> is the name that was entered when saving the Pattern; for example:</p> <p><code>BusFramework : Builder( UMLPattern )</code></p> <p>If you want to avoid displaying the Add Pattern dialog, replace <code>( UMLPattern )</code> with <code>( UMLPatternSilent )</code>.</p> <p>6. To define a model-based Pattern in a custom Toolbox (such as the GoF Patterns), create an attribute with a name of the format:</p> <p><code>PatternCategory : PatternName( UMLPattern )</code></p> <p>For example:</p> <p><code>GoF Behavioral Patterns : Mediator( UMLPattern )</code></p> </li></ol>	<p><a href="#">Design Patterns</a> <sup>[1464]</sup></p> <p><a href="#">Create MDG Technology File</a> <sup>[1545]</sup></p> <p><a href="#">Create a Pattern</a> <sup>[1464]</sup></p>
16	After you add the Toolbox item it will appear in the <b>Toolbox Items</b> list. You can	

	<p>optionally add a custom icon image for a Toolbox item.</p> <p>The icon image must be a <b>16x16</b> pixel <b>bitmap</b> file; for a transparent background use light grey - <b>RGB(192,192,192)</b>.</p> <p>To set the icon for a Toolbox item:</p> <ol style="list-style-type: none"> <li>1. Locate the item in the <b>Toolbox Items</b> list and click within the <b>Toolbox Icon</b> column.</li> <li>2. Click on the ellipsis (...) within this column. The Select a Toolbox Icon dialog displays.</li> <li>3. Locate the image file and click the <b>Open</b> button.</li> </ol>	
17	<p>Repeat steps 13 to 16 for each item you want to add to the Toolbox Page.</p> <p>To remove a Toolbox item, select it in the <b>Toolbox Items</b> list and click on the <b>Delete</b> button.</p> <p>Once all the appropriate Toolbox items have been added, click on the <b>OK</b> button. A Stereotype element will be added to your Toolbox Profile diagram.</p>	
18	Repeat steps 8 to 17 for each Toolbox Page you want to include in the Toolbox.	
19	Save the Toolbox Profile by right-clicking on the diagram and selecting the <b>Save as Profile</b> context menu option.	<a href="#">Export a Profile</a> <small>[1523]</small>

### Notes

- A Toolbox Page can be modified by right-clicking the appropriate Stereotype element on the Toolbox Profile diagram and selecting the **Edit with Profile Helper** option
- When assigning a name for a Toolbox Page, be aware that "elements" is a reserved word; if the word "elements" is used, it will not appear in the title bar of the corresponding Toolbox Page
- The sequence of Toolbox Pages in the Toolbox is determined by the sequence of their Stereotype elements in the Profile diagram or Profile Package; if you create and save the Profile from a:
  - Diagram, the Toolbox Page sequence is determined by the **Z-order** of the Stereotype elements on the diagram - the higher the Z-order number of the Stereotype element, the further down the Toolbox its Toolbox Page is placed; if you change the Z-order of a Stereotype element in the diagram, it changes the position of the element's page on the Toolbox
  - Package in the Project Browser, the Toolbox Page sequence is determined by the **list order** of the Stereotype elements in the Package - the Toolbox Page for the first listed element is at the top of the Toolbox; if you re-order the elements in the Project Browser, you produce the same re-ordering of pages in the Toolbox


### Learn more

- [Z Order Elements](#) [844]
- [Diagram Context Menu](#) [778] (Modify Z-Order option)
- [Using the Profile Helpers](#) [1528]
- [Create UML Profiles using Profile Helpers](#) [1529]
- [Create Toolbox Profiles using the Profile Helpers](#) [1538]

### 7.3.3.1.3.1 Create Hidden Sub-Menus using the Profile Helpers


When you create Toolbox items, some of them may be very similar in that they are based on the same type of Metaclass. For example, there are many different types of Action element. Rather than populate a Toolbox Page with every variation, you can create a 'base' Toolbox item and offer a choice of variant from a **Sub-Menu**, which is displayed when the base item is dragged onto the diagram.

#### Define a hidden sub-menu

Step	Action	See also
1	If you have not already done so, create and display the diagram you will be using to define your Toolbox, as described in steps 1 to 6 of <i>Create Toolbox Profiles using the Profile Helpers</i> .	<a href="#">Create Toolbox Profiles using the Profile Helpers</a> 
2	Drag the <b>Add a Toolbox Page</b> item from the Profile Helpers Toolbox Page onto the diagram. The Add Toolbox Page dialog displays.	
3	In the <b>Name</b> field, type the name for the sub-menu Toolbox item.	
4	The <b>Tool Tip</b> field can be left blank in this case.	
5	Select the <b>Is Hidden</b> checkbox. The <b>Images Only</b> , <b>Is Common</b> and <b>Is Collapsed</b> checkboxes should be left unselected.	
6	After selecting the <b>Is Hidden</b> checkbox, the <b>Icon</b> field should become active. You can optionally add a custom icon image for the sub-menu Toolbox item. The icon image must be a <b>16x16 pixel bitmap</b> file; for a transparent background use light grey - <b>RGB(192,192,192)</b> . To set the icon for the sub-menu Toolbox item, click on the folder icon to the right of the <b>Icon</b> field. Select the image file and click the <b>Open</b> button.	
7	You can now add items such as elements and connectors to the sub-menu. Click on the down arrow on the right of the <b>Add</b> button, and select one of the following options: <ul style="list-style-type: none"> <li>• <b>Add Stereotype</b>: adds a Toolbox item for a Stereotype that is defined in a UML Profile in the current model; this Profile must be included with the Toolbox Profile in the MDG Technology</li> </ul> <p>After you select this option, the Select a Profile Element dialog displays; use this to select the Stereotype you want to add</p>	

Step	Action	See also
	<ul style="list-style-type: none"> <li>• <b>Add Built in Type:</b> <ul style="list-style-type: none"> <li>• <b>Element:</b> adds a Toolbox item for a UML <b>element type</b></li> </ul> <p>After you select this option the Create new Toolbox Item dialog displays; in the <b>Alias</b> field, type the label to appear on the Toolbox item, and click on the <b>OK</b> button</p> <p>The Select Metaclass dialog then displays; select the UML element type to add to your Toolbox, and click on the <b>OK</b> button</p> </li> <li>• <b>Connector:</b> adds a Toolbox item for a UML <b>connector type</b></li> </ul> <p>After you select this option the Create new Toolbox Item dialog displays; in the <b>Alias</b> field, type the label to appear on the Toolbox item, and click on the <b>OK</b> button</p> <p>The Select Metaclass dialog then displays; select the UML connector type to add to your Toolbox, and click on the <b>OK</b> button</p> <ul style="list-style-type: none"> <li>• <b>Add Hidden Toolbox:</b> adds a hidden Toolbox sub-menu item; do not use this option when creating the Hidden Toolbox sub-menu itself</li> <li>• <b>Add New Item:</b> adds a Toolbox item with an Alias only</li> </ul> <p>This option alone will not create a functional Toolbox item; a Toolbox item added in this way must be later modified via the <b>Toolbox Items</b> list</p> <p>Clicking on the <b>Add</b> button, and not on the drop-down arrow, is the same as selecting the <b>Add Stereotype</b> option.</p>	
8	<p>(Optional) After adding the Toolbox item it will appear in the <b>Toolbox Items</b> list, and you can add a custom icon image for the item.</p> <p>The icon image must be a <b>16x16 pixel bitmap</b> file; for a transparent background use <b>light grey - RGB(192,192,192)</b>.</p> <p>To set the icon for a Toolbox item, locate the item in the <b>Toolbox Items</b> list and click within the <b>Toolbox Icon</b> column. Click on the ellipsis (...) within this column. The Select a Toolbox Icon dialog displays. Locate the image file and click the <b>Open</b> button.</p>	
9	<p>Repeat steps 7 and 8 for each item to add to the sub-menu.</p> <p>To remove a Toolbox item, select it from the <b>Toolbox Items</b> list and click on the <b>Delete</b> button.</p> <p>Once all the appropriate sub-menu items have been added, click on the <b>OK</b> button. A Stereotype element will be added to your Toolbox Profile diagram.</p>	
10	Repeat steps 2 to 9 for each Toolbox sub-menu to create.	



Step	Action	See also
11	The sub-menu(s) created above can now be included as an item in a Toolbox Page.	<a href="#">Create Toolbox Profiles using the Profile Helpers</a> 

### Notes

- A sub-menu can be modified by right-clicking the appropriate Stereotype element on the Toolbox Profile diagram and selecting the **Edit with Profile Helper** option

### Learn more

- [Using the Profile Helpers](#) 
- [Create Stereotype Profiles using Profile Helpers](#) 

## 7.3.3.2 Create MDG Technology File















When you create an MDG Technology file, you can include a wide range of facilities and tools, including UML Profiles, code modules, scripts, Patterns, images, Tagged Value Types, report templates, linked document templates, Toolbox pages and Learning Center pages. Building all of these into the MDG Technology file in a logical sequence is easy, using the **MDG Technology Creation Wizard**.

**Access** **Tools | Generate MDG Technology File**

### Create an MDG Technology file

Step	Description	See also
1	Select the <b>Generate MDG Technology File</b> menu option. The MDG Technology Creation Wizard screen displays.	
2	Click on the <b>Next</b> button. The MDG Technology Wizard prompts you to: <ul style="list-style-type: none"> <li>Create an MDG Technology file based on a new MDG Technology Selection (MTS) file</li> <li>Create an MDG Technology file based on an existing MTS file, or</li> <li>Not use any MTS file</li> </ul> <p>An MTS file stores the selected options that you define during the creation of an MDG Technology; if you use an MTS file, you can later modify it to add or remove specific items in the MDG Technology, which is the</p>	

Step	Description	See also																
	<b>recommended</b> process.																	
3	<p>Select the appropriate MTS file option.</p> <p>Click on the <b>Next</b> button.</p> <p>If you selected an MTS file, the MDG Technology Wizard prompts you to save the changes in the existing MTS file or into a new MTS file; this enables you to create a modification based on the existing MTS file, while preserving the original file.</p>																	
4	<p>If necessary, type in or browse for the required file path and name.</p> <p>Click on the <b>Next</b> button.</p> <p>The MDG Technology Wizard - Create screen displays.</p>																	
5	<p>Complete the fields on this screen as follows:</p> <table><tr><th>Option</th><th>Action</th></tr><tr><td><b>Technology</b></td><td>Type the name of the MDG Technology.</td></tr><tr><td><b>Filename</b></td><td>Type or select the path and filename of the MDG Technology. The file extension for this file is <b>.xml</b>.</td></tr><tr><td><b>ID</b></td><td>Type a unique reference for the MDG Technology File, up to</td></tr><tr><td><b>Version</b></td><td>Type the version number of the MDG Technology File.</td></tr><tr><td><b>Icon</b></td><td>(Optional) Type or select the path and file name of the graph technology icon.  The icon is a 16x16 bitmap image that is shown in the list of the MDG Technologies dialog.</td></tr><tr><td><b>Logo</b></td><td>(Optional) Type or select the path and file name of the graph technology logo.  The logo is a 64x64 bitmap image that is shown in the display corner of the MDG Technologies dialog.</td></tr><tr><td><b>URL</b></td><td>(Optional) If you have any website product information that is part of this Technology, type or paste the URL in this field.</td></tr></table>	Option	Action	<b>Technology</b>	Type the name of the MDG Technology.	<b>Filename</b>	Type or select the path and filename of the MDG Technology. The file extension for this file is <b>.xml</b> .	<b>ID</b>	Type a unique reference for the MDG Technology File, up to	<b>Version</b>	Type the version number of the MDG Technology File.	<b>Icon</b>	(Optional) Type or select the path and file name of the graph technology icon.  The icon is a 16x16 bitmap image that is shown in the list of the MDG Technologies dialog.	<b>Logo</b>	(Optional) Type or select the path and file name of the graph technology logo.  The logo is a 64x64 bitmap image that is shown in the display corner of the MDG Technologies dialog.	<b>URL</b>	(Optional) If you have any website product information that is part of this Technology, type or paste the URL in this field.	
Option	Action																	
<b>Technology</b>	Type the name of the MDG Technology.																	
<b>Filename</b>	Type or select the path and filename of the MDG Technology. The file extension for this file is <b>.xml</b> .																	
<b>ID</b>	Type a unique reference for the MDG Technology File, up to																	
<b>Version</b>	Type the version number of the MDG Technology File.																	
<b>Icon</b>	(Optional) Type or select the path and file name of the graph technology icon.  The icon is a 16x16 bitmap image that is shown in the list of the MDG Technologies dialog.																	
<b>Logo</b>	(Optional) Type or select the path and file name of the graph technology logo.  The logo is a 64x64 bitmap image that is shown in the display corner of the MDG Technologies dialog.																	
<b>URL</b>	(Optional) If you have any website product information that is part of this Technology, type or paste the URL in this field.																	

Step	Description	See also								
	<table><tr><th>Option</th><th>Action</th></tr><tr><td></td><td></td></tr><tr><td><b>Support</b></td><td>(Optional) If you have any web-based or other support facility for users of this Technology, type or paste the contact address</td></tr><tr><td><b>Notes</b></td><td>Type a short explanation of the functionality of the MDG Technology</td></tr></table>	Option	Action			<b>Support</b>	(Optional) If you have any web-based or other support facility for users of this Technology, type or paste the contact address	<b>Notes</b>	Type a short explanation of the functionality of the MDG Technology	
Option	Action									
<b>Support</b>	(Optional) If you have any web-based or other support facility for users of this Technology, type or paste the contact address									
<b>Notes</b>	Type a short explanation of the functionality of the MDG Technology									
6	Click on the <b>Next</b> button.  The MDG Technology Wizard - Contents screen displays.									
7	Select the checkbox for each item to be included in the MDG Technology file.  When you have selected the checkboxes for all the items you want to include, click on the <b>Next</b> button.  Each selection runs specific dialogs to enable definition of the specific items to be included in the MDG Technology.	<a href="#">Add a Profile</a>  <sup>[1548]</sup> <a href="#">Add a Pattern</a>  <sup>[1549]</sup> <a href="#">Add a Diagram Profile</a>  <sup>[1550]</sup> <a href="#">Add a Toolbox Profile</a>  <sup>[1551]</sup> <a href="#">Add Tagged Value Types</a>  <sup>[1551]</sup> <a href="#">Add Code Modules</a>  <sup>[1552]</sup> <a href="#">Add MDA Transforms</a>  <sup>[1554]</sup> <a href="#">Add Document Report Templates</a>  <sup>[1555]</sup> <a href="#">Add Linked Document Templates</a>  <sup>[1555]</sup> <a href="#">Add Images</a>  <sup>[1556]</sup> <a href="#">Add Scripts</a>  <sup>[1556]</sup> (Corporate and 'Suite' editions) <a href="#">Add Workspace Layouts</a>  <sup>[1557]</sup> <a href="#">Add Model Views</a>  <sup>[1558]</sup> <a href="#">Add Model Searches</a>  <sup>[1559]</sup>								
8	Work through the dialogs displayed in response to your choices, and when all are complete, click on the <b>Next</b> button.  The MDG Technology Wizard - Finish screen displays, providing information on the items included in the MDG Technology File.									

Step	Description	See also
9	If you have used an MTS file and want to update it, select the <b>Save to MTS</b> checkbox.	
10	<p>If you are satisfied with the selection of items, click on the <b>Finish</b> button.</p> <p>You can now edit the MTS file, if required, to add further items such as:</p> <ul style="list-style-type: none"> <li>Model Validation configurations</li> <li>Model Templates</li> </ul> <p>When you have edited the MTS file and regenerated the Technology (.xml) file, you can add another Scripts section to include package XML Export and/or Import scripts. Save the edited Technology file.</p> <p>To make the MDG Technology .xml file accessible to an Enterprise Architect model, you must add the technology file path to the MDG Technologies - Advanced dialog.</p>	<p><a href="#">Working with MTS Files</a> <small>[1559]</small></p> <p><a href="#">Add Import/Export Scripts</a> <small>[1578]</small></p> <p><a href="#">Access Remote MDG Technologies</a> <small>[1479]</small></p>

### Notes

- The facility to create MDG Technologies is not available in the Enterprise Architect Desktop edition

### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Modeling Languages | Build a Technology | Build a Technology**

#### 7.3.3.2.1 Add a Profile

When creating an MDG Technology file, you can include one or more UML 2.4.1-compliant **Profiles** that you have defined to create new types of model element.

**Access** **Tools | Generate MDG Technology File**

#### Add Profiles to the MDG Technology File

Step	Description	See also
1	<p>Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the <b>Profiles</b> checkbox.</p> <p>The MDG Technology Wizard - Profile files selection page displays.</p>	<a href="#">Create MDG Technologies, Step 6</a> <small>[1547]</small>
2	<p>In the <b>Directory</b> field, navigate to the directory containing the required Profile or Profiles.</p> <p>The Profile files are automatically listed in the Available Files panel.</p>	

Step	Description	See also
3	<p>To select each required Profile individually, highlight the Profile in the Available Files list and click on the --&gt; button.</p> <p>The file name displays in the Selected Files list.</p> <p><i>Alternatively:</i></p> <p>To select all available Profiles click on the --&gt;&gt; button, and return each one you do <b>not</b> want by selecting it and clicking on the &lt;-- button.</p> <ul style="list-style-type: none"> <li>• DO NOT select Diagram Profiles or Toolbox Profiles on this dialog; this would generate conflicting commands in the .MTS file</li> <li>• Make sure you do include your UML Profiles</li> </ul>	
4	Click on the <b>Next</b> button to proceed.	

#### Learn more


- [Developing Profiles](#) 
- [Create UML Profiles](#) 

#### 7.3.3.2.2 Add a Pattern

When creating an MDG Technology file, you can include special **Design Patterns** that you want to make available as element templates.

Access **Tools | Generate MDG Technology File**

#### Add Design Patterns to the MDG Technology File

Step	Description	See also
1	<p>Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the <b>Patterns</b> checkbox.</p> <p>The MDG Technology Wizard - Pattern files selection page displays.</p>	<a href="#">Create MDG Technologies, Step 6</a> 
2	<p>In the <b>Directory</b> field, navigate to the directory containing the required Pattern or Patterns.</p> <p>The Pattern files are automatically listed in the Available Files panel.</p>	
3	<p>To select each required Pattern individually, highlight the Pattern in the Available Files list and click on the --&gt; button.</p> <p>The file name displays in the Selected Files list.</p>	

Step	Description	See also
	Alternatively, to select all available Patterns click on the -->> button, and return each one you do <b>not</b> want by selecting it and clicking on the <-- button.	
4	Click on the <b>Next</b> button to proceed.	

#### Learn More:

- [Design Patterns](#)<sup>[1464]</sup>
- [Create a Pattern](#)<sup>[1464]</sup>

#### 7.3.3.2.3 Add a Diagram Profile

When creating an MDG Technology file, you can include **Diagram Profiles** that you have defined to generate new types of diagram.

**Access** [Tools](#) | **Generate MDG Technology File**

#### Add Diagram Profiles to the MDG Technology File

Step	Description	See also
1	Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the <b>Diagram Types</b> checkbox. The MDG Technology Wizard - Diagram Types page displays.	<a href="#">Create MDG Technologies, Step 6</a> <sup>[1547]</sup>
2	In the <b>Directory</b> field, navigate to the directory containing the required Diagram Profiles. The Profiles in the directory are automatically listed in the Available Files panel.	
3	To select each required Diagram Profile individually, highlight the file name in the Available Files list and click on the --> button. The file name displays in the Selected Files list. Alternatively, to select all available Profiles (if they are all Diagram Profiles) click on the -->> button, and return each one you do <b>not</b> want by selecting it and clicking on the <-- button.	
4	Click on the <b>Next</b> button to proceed.	

Learn more


- [Create Custom Diagram Profiles](#) 

#### 7.3.3.2.4 Add a Toolbox Profile

When creating an MDG Technology file, you can include Diagram Toolbox page definitions that you have created to provide Toolbox pages to support customized diagrams.

Access   **Tools | Generate MDG Technology File**

##### Add Toolbox Profiles to the MDG Technology File

Step	Description	See also
1	Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the <b>Toolboxes</b> checkbox.  The MDG Technology Wizard - Toolboxes page displays.	<a href="#">Create MDG Technologies, Step 6</a> 
2	In the <b>Directory</b> field, navigate to the directory containing the required Toolbox Profiles.  The Profile files are automatically listed in the Available Files panel.	
3	To select each required Toolbox Profile individually, highlight the file name in the Available Files list and click on the --> button.  The file name displays in the Selected Files list.  Alternatively, to select all available Profiles (if they are all Toolbox Profiles) click on the -->> button, and return each one you do <b>not</b> want by selecting it and clicking on the <-- button.	
4	Click on the <b>Next</b> button to proceed.	

Learn more

- [Customize Toolbox Profiles](#) 

#### 7.3.3.2.5 Add Tagged Value Types

When creating an MDG Technology file, you can include Tagged Value Types, from which the technology users can create domain-specific Tagged Values.

Access   **Tools | Generate MDG Technology File**

##### Add Tagged Value Types to the MDG Technology File

Step	Description	See also
1	Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the <b>Tagged Value Types</b> checkbox.  The MDG Technology Wizard - Tagged Value Types page displays.	<a href="#">Create MDG Technologies, Step 6</a> <sup>[1547]</sup>
2	To select each required Tagged Value Type individually, highlight the name in the Available Tagged Values list and click on the --> button.  The name displays in the Selected Tagged Values list, and the name, description and notes on the Tagged Value Type are displayed in the panel at the bottom of the page.  Alternatively, to select all available Tagged Value Types, click on the -->> button, and return each one you do <b>not</b> want by selecting it and clicking on the <-- button.	
3	Click on the <b>Next</b> button to proceed.	

#### Learn more

- [Tagged Value Types](#) <sup>[1627]</sup>

#### 7.3.3.2.6 Add Code Modules

When creating an MDG Technology file, you can include **code modules** for which you have set up **code templates** and **data types**. The modules can be for modifications to the system default languages, or for languages you have defined yourself using the code templates and the Code Template Editor. Before you can set up a code template for a new language in the editor, you must define at least one data type for the language. You can also specify code options for the language, which are additional settings that are not covered by the data types or code templates; they are held in an XML document that you include in the MDG Technology file with the module.

Access   **Tools | Generate MDG Technology File**

#### Add Code Modules to the MDG Technology File

Step	Description	See also
1	Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the <b>Code Modules</b> checkbox.  The MDG Technology Wizard - Code Modules page displays, listing the code modules defined in your current project.	<a href="#">Create MDG Technologies, Step 6</a> <sup>[1547]</sup>
2	Click on the checkboxes ( <b>Product</b> , <b>Data Types</b> , <b>Code Grammar</b> , and <b>Code Templates</b> ) for each of the code modules you want to include in the technology.	<a href="#">Developing Programming Languages</a> <sup>[1581]</sup>



Step	Description	See also
3	If you have created a code options XML document for a selected module, click on the ( ... ) button in the <b>Code Options</b> column for that module. A browser displays, through which you locate and select the XML document.	<a href="#">Define Code Options</a> [1553]
4	Click on the <b>Next</b> button to proceed.	

#### 7.3.3.2.6.1 Define Code Options

When modifying code generation templates for an existing programming language, or defining a new programming language, there are additional options which are only available when building an MDG Technology. These additional options can affect how Enterprise Architect handles code generation and reverse-engineering for this language. These options are specified using an XML file, created using your preferred text editor.

The root node in the XML document is named `CodeOptions`. The child nodes are named `CodeOption`. Each `CodeOption` contains a `name` attribute corresponding to the name of one of the available code options. The body of each node contains the option value. For example:

```
<CodeOptions>
  <CodeOption name="DefaultExtension">.h</CodeOption>
  <CodeOption name="HasImplementation">true</CodeOption>
  <CodeOption name="ImplementationExtension">.cpp</CodeOption>
  <CodeOption name="Editor">C:\Windows\notepad.exe</CodeOption>
</CodeOptions>
```

#### Supported code options

Code Option	Description
<b>ConstructorName</b>	The name of a function used as a constructor. Used by the <i>classHasConstructor</i> code template macro.
<b>CopyConstructorName</b>	The name of a function used as a copy constructor. Used by the <i>classHasCopyConstructor</i> code template macro.
<b>DefaultExtension</b>	The default extension when generating code.
<b>DefaultSourceDirectory</b>	The default path to which Enterprise Architect generates new files.
<b>DestructorName</b>	The name of a function used as a destructor. Used by the <i>classHasDestructor</i> code template macro.
<b>Editor</b>	The external editor used for editing source of this language.

Code Option	Description
<b>HasImplementation</b>	Specifies if code generation for this language generates both a source file and implementation file.
<b>ImplementationExtension</b>	The extension used by Enterprise Architect to generate an implementation file.
<b>ImplementationPath</b>	The relative path from the source file to generate the implementation file.
<b>PackagePathSeparator</b>	The delimiter used to separate package names when using the <i>packagePath</i> macro from the code templates.

#### Notes

- Once a language is available for use in a model (by importing and activating the MDG Technology), you can display and edit the code options on the Options dialog (select the **Tools | Options** menu option)

#### Learn more

- [Local Options](#) <sup>[604]</sup>
- [Source Code Options](#) <sup>[2247]</sup>
- [Language Options](#) <sup>[2262]</sup>
- [Field Substitution Macros](#) <sup>[1648]</sup>

#### 7.3.3.2.7 Add MDA Transforms

When creating an MDG Technology file, you can include any MDA Transformation templates that you have created or modified in the model and that you want to deploy as part of the technology.

**Access** **Tools | Generate MDG Technology File**

#### Add MDA Transformation Templates to the MDG Technology File

Step	Description	See also
1	Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the <b>MDA Transforms</b> checkbox.  The MDG Technology Wizard - Transform Modules page displays, listing the MDA transform templates available on your system.	<a href="#">Create MDG Technologies, Step 6</a> <sup>[1547]</sup>
2	Click the checkbox against the name of each transformation template you want to add to your MDG Technology.	

Step	Description	See also
3	Click on the <b>Next</b> button to proceed.	

#### Learn more

- [Model Transformation](#)<sup>[2013]</sup>
- [Write Transformations](#)<sup>[2051]</sup>

### 7.3.3.2.8 Add Document Report Templates

When creating an MDG Technology file, you can include user-defined Document Report templates.

Access **Tools | Generate MDG Technology File**

#### Add Report Templates to the MDG Technology File

Step	Description	See also
1	Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the <b>RTF Templates</b> checkbox. The MDG Technology Wizard - RTF Report Templates dialog displays.	<a href="#">Create MDG Technologies, Step 6</a> <sup>[1547]</sup>
2	For each required user-defined report template available in the current model, select the checkbox next to the template name.	
3	Click on the <b>Next</b> button to proceed.	


### 7.3.3.2.9 Add Linked Document Templates

When creating an MDG Technology file, you can include Linked Document templates.

Access **Tools | Generate MDG Technology File**

#### Add Linked Document Templates to the MDG Technology File

Step	Description	See also
1	Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the <b>Linked Document Templates</b> checkbox.	<a href="#">Create MDG Technologies, Step 6</a>


Step	Description	See also
	The MDG Technology Wizard - Linked Document Templates dialog displays.	
2	For each required document template available in the current model, select the checkbox next to the template name.	
3	Click on the <b>Next</b> button to proceed.	

### 7.3.3.2.10 Add Images

When creating an MDG Technology file, you can incorporate images to be used in all models in which the technology is *deployed*. These images must already be available in the model in which the technology is being *developed*; you can import the images into this model using the **Add New** button on the Image Manager.

Access **Tools | Generate MDG Technology File**

#### Add Images to the MDG Technology File

Step	Description	See also
1	Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the <b>Images</b> checkbox.  The MDG Technology Wizard - Image Selection dialog displays.	<a href="#">Create MDG Technologies, Step 6</a> 
2	For each required model image available in the current model, select the checkbox next to the image name.  A preview of each image displays on the right of the dialog as you select the checkbox.	
3	Click on the <b>Next</b> button to proceed.	

#### Learn more

- [Image Manager](#)

### 7.3.3.2.11 Add Scripts

When creating an MDG Technology file, you can include scripts that you have created in the model.

Access **Tools | Generate MDG Technology File**

**Add Scripts to the MDG Technology File**

Step	Description	See also
1	Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the <b>Scripts</b> checkbox. The MDG Technology Wizard - Scripts dialog displays.	<a href="#">Create MDG Technologies, Step 6</a> <sup>[1547]</sup>
2	For each required script available in the current model, select the checkbox next to the script name.	
3	Click on the <b>Next</b> button to proceed.	

**Notes**

- This facility is available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect

**Learn more**

- [The Scripter Window](#) <sup>[2791]</sup>

**7.3.3.2.12 Add Workspace Layouts**

When developing an MDG Technology file, you can include user-defined **workspace layouts**. Workspace layouts are arrangements of toolbars and windows appropriate to an area of work such as Requirements Management and Code Engineering. The workspace layout automatically opens and organizes all the tools to suit the way in which you use the system.

**Access** **Tools | Generate MDG Technology File**

**Add Workspace Layouts to the MDG Technology File**

Step	Description	See also
1	In your model, create the workspace layouts you want to include in your Technology.	<a href="#">Manage Workspace Layout</a> <sup>[163]</sup>
2	Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the <b>Workspace Layouts</b> checkbox. The MDG Technology Wizard - Workspace Layouts dialog displays, listing the user-defined workspace layouts available to you.	<a href="#">Create MDG Technologies, Step 6</a> <sup>[1547]</sup>

Step	Description	See also
3	For each workspace layout that you want to incorporate in the Technology, select the checkbox next to the layout name.	
4	Click on the <b>Next</b> button to proceed.	

### 7.3.3.2.13 Add Model Views

When developing an MDG Technology file, you can include user-defined Model Views. Model Views are based on searches that extract specific information from a model to provide different perspectives of, and 'entrypoints' into, the model.

**Access** [Tools](#) | **Generate MDG Technology File**

#### Add Model Views to the MDG Technology File

Step	Description	See also
1	In your model, create the Model Views you want to include in your Technology.	<a href="#">Model Views</a> <sup>[686]</sup>
2	Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the <b>Model Views</b> checkbox.  The MDG Technology Wizard - Model Views dialog displays, listing the user-defined views available in the current model.	<a href="#">Create MDG Technologies, Step 6</a> <sup>[1547]</sup>
3	For each Model View that you want to incorporate in the Technology, select the checkbox next to the view name.	
4	Click on the <b>Next</b> button to proceed.	

#### Notes

- Technology views do not store Favorite packages, only Views
- If you incorporate a Model View that runs searches that you have defined, you must also include those searches in your MDG Technology

#### Learn more

- [Model Views Operations](#) <sup>[693]</sup>
- [Add Model Searches](#) <sup>[1559]</sup>

### 7.3.3.2.14 Add Model Searches

When developing an MDG Technology file, you can include user-defined Model Searches. You can set these searches up using the Model Search facility, in SQL, in the Query Builder or as an Add-In, and then link them into your MDG Technology.

**Access** [Tools](#) | **Generate MDG Technology File**

#### Add Model Searches to the MDG Technology File

Step	Description	See also
1	In your model, create the Model Searches you want to include in your Technology.	<a href="#">Create &amp; Modify Searches</a> <sup>[709]</sup>
2	Follow the steps in the <i>Create MDG Technologies</i> topic up to and including Step 6, where you select the <b>Model Searches</b> checkbox.  The MDG Technology Wizard - Model Searches dialog displays, listing the user-defined searches available in the current model.	<a href="#">Create MDG Technologies, Step 6</a> <sup>[1547]</sup>
3	For each Model Search that you want to incorporate in the Technology, select the checkbox next to the search name.	
4	Click on the <b>Next</b> button to proceed.	

#### Notes

- If you use a custom SQL search, the SQL must include *ea\_guid AS CLASSGUID* and the *object type*
- If you incorporate a Model View that runs searches that you have defined, you must also include those searches in your MDG Technology

#### Learn more

- [Add Model Views](#) <sup>[1558]</sup>

### 7.3.3.3 Working with MTS Files

When you are creating an MDG Technology File using the MDG Technology Wizard, you have the choice of storing all of the options and structures that you have defined in an **MDG Technology Selection (.MTS) file**. This captures all the information you enter into the Technology Wizard, so that you do not have to type it in again. If you use a .MTS file, you can subsequently edit it to change the features you selected when you generated the Technology file, and to add or remove additional, advanced features.

**Access** [Tools](#) | **Generate MDG Technology File**

**Manage the .MTS file**

Action	Description	See also
<b>Create a .MTS File</b>	To create a .MTS file, launch and work through the MDG Technology Wizard; on the second page, select the <b>Create a new MTS file</b> option.	<a href="#">Create MDG Technologies</a> <sup>[1545]</sup>
<b>Advanced Options For Your .MTS File</b>	<p>Once you have worked through the MDG Technology Wizard and set up the .MTS file, you can add, separately:</p> <ul style="list-style-type: none"> <li>• Model Validation configurations</li> <li>• Model Templates</li> </ul> <p>Firstly define the code for the model validation configurations and model templates, then open the .MTS file in a text editor and copy in the validation and/or template code just before the <code>&lt;/ MDG. Sel ect i ons&gt;</code> line.</p> <p>Save the .MTS file.</p>	<a href="#">Define Validation Configuration</a> <sup>[1576]</sup> <a href="#">Incorporate Model Templates</a> <sup>[1576]</sup>
<b>Update the MDG Technology</b>	<p>Again launch the MDG Technology Wizard, but this time on the second page select the <b>Open an Existing MTS file</b> option and specify the file path of the .MTS file you have been working on.</p> <p>Click on <b>&lt;Next&gt;</b> until the Wizard is finished; your MDG Technology .xml file is updated.</p>	

**Notes**

- Having created your MDG Technology with the Wizard and the .MTS file, you can add Import and Export scripts via the Technology .XML file

**Learn more**

- [Add Import/Export Scripts](#) <sup>[1578]</sup>






**7.3.3.4 Create Toolbox Profiles**

As a facility of your MDG Technology, you might want to provide Diagram Toolbox pages that give access to any elements and connectors that you have created within the technology. You define these Toolbox pages within specific Profiles, each Profile defining the element and connector Toolbox pages that open or can be selected for a diagram type.

**Create custom Toolboxes**

Step	Action	See also
1	Create a set of Toolbox Profiles that contain the definitions required to generate the Toolbox pages.	<a href="#">Create Toolbox Profiles</a> <sup>[1561]</sup>





Step	Action	See also
2	Edit the definitions, where appropriate, to: <ul style="list-style-type: none"> <li>• Include hidden sub-menus</li> <li>• Override the default Toolboxes</li> <li>• Change the default icons for Toolbox items</li> </ul>	<a href="#">Create Hidden Sub-menus</a>  <sup>[1564]</sup> <a href="#">Override Default Toolboxes</a>  <sup>[1567]</sup> <a href="#">Assign Icons to Toolbox Items</a>  <sup>[1565]</sup>
3	Create a .MTS file containing instructions on how to build your MDG Technology, and include the Toolbox Profiles in the technology.	<a href="#">Create MDG Technologies</a>  <sup>[1545]</sup> <a href="#">Add a Toolbox Profile</a>  <sup>[1551]</sup>

#### 7.3.3.4.1 Create Toolbox Profiles

Within an MDG Technology you can create multiple Toolbox Profiles. Each Toolbox Profile contains definitions that determine what pages appear in the Toolbox when it is opened, either by selection from the **More tools...** option in the Diagram Toolbox, or by opening or creating a diagram of the type that is linked to the Toolbox Profile.

##### Create a Toolbox Profile

Step	Action	See also
1	In a Profile package, create a Class diagram with an appropriate name by which you can refer to it later, such as <b>MyClassDiagram</b> .	<a href="#">Create a Profile Package</a>  <sup>[1487]</sup>
2	Double-click on the diagram background to display the diagram Properties dialog and, in the <b>Notes</b> field, give the diagram an alias and a description in the following format: <pre>Alias=MyClass; Notes=Structural elements for Class diagrams;</pre>	
3	On the diagram, create a Metaclass element with the name <b>ToolboxPage</b> .	
4	Create a Stereotype element for each of the Toolbox pages to create within your Toolbox, such as <i>MyClassElements</i> and <i>MyClassRelationships</i> . Double-click on each element to display the Properties dialog and, in the <b>Alias</b> field, type the text to display in the title bar of the corresponding Toolbox page, such as <b>My Classes</b> or <b>My Class Relationships</b> . In the <b>Notes</b> field of each element, type the tool-tip for the corresponding Toolbox	<a href="#">Add Stereotypes and Metaclasses</a>  <sup>[1488]</sup>

Step	Action	See also
	<p>page; for example, <b>Elements for Class Diagrams</b> or <b>Relationships for Class Diagrams</b>.</p> <p>Create an Extension connector between each Stereotype element and the <b>ToolboxPage</b> Metaclass element.</p>	
5	<p>In each of the Stereotype elements, press <b>F9</b> and create an attribute for each Toolbox item in the page defined by that element.</p> <p>The name of each attribute is the name of the element or connector to be dropped, including the element's namespace; for example, <i>UML::Package</i>, <i>UML::Class</i> and <i>UML::Interface</i>. You might not want to display names including text such as <i>UML::Package</i> or <i>UML::Class</i> in your Toolbox, so give the attributes an <b>Initial Value</b> of, for example, <i>Package</i> or <i>Class</i>.</p> <p>The Toolbox items display in the same sequence as their attributes in the element, so use the attribute ordering buttons (green up/down arrows) in the Attribute Properties dialog to define the order of icons in your Toolbox page.</p> <p>In the name of an attribute for an element or connector from <b>your own</b> technology, use your Profile name as the namespace, and then follow the item name with the element or connector type that you are extending, in brackets (to identify to Enterprise Architect what type of object to create); for example, a SysML Block element would appear as:</p> <p style="text-align: center;"><i>SysML::Block(UML::Class)</i></p> <p>Many elements and connectors can be extended for use in Toolboxes.</p>	<p><a href="#">General Properties of Attributes</a> <small>[1007]</small></p> <p><a href="#">Elements Used in Toolboxes</a> <small>[1567]</small></p> <p><a href="#">Connectors Used in Toolboxes</a> <small>[1569]</small></p>
6	<p>To define a Toolbox item to drop a <b>Pattern</b> onto a diagram, name the attribute:</p> <p style="text-align: center;"><i>My Technology::MyPattern(UMLPattern)</i></p> <p><i>MyTechnology</i> is the ID of the technology and <i>MyPattern</i> is the name of the Pattern to drop; for example:</p> <p style="text-align: center;"><i>BusFramework::Builder(UMLPattern)</i></p> <p>If you want to avoid displaying the Add Pattern dialog, replace <i>(UMLPattern)</i> with <i>(UMLPatternSilent)</i>.</p> <p>To define a <b>model-based Pattern</b> in a custom Toolbox (such as the GoF Patterns), create an attribute with a name of the format:</p> <p style="text-align: center;"><i>PatternCategory::PatternName(UMLPattern)</i></p> <p>For example:</p> <p style="text-align: center;"><i>GoF Behavioral Patterns::Mediator(UMLPattern)</i></p>	

Step	Action	See also
7	Define any attributes you need to modify the display of the Toolbox pages, such as whether the Toolbox pages are minimized or displayed without item names (labels).	<a href="#">Toolbox Page Attributes</a> <sup>[1563]</sup>
8	To save the Toolbox profile, right-click on the diagram and select the <b>Save as Profile</b> context menu option.	<a href="#">Export a Profile</a> <sup>[1523]</sup>

### Notes

- When assigning an **Alias** for a Toolbox page, 'elements' is a reserved word. If the word 'elements' is used, it will not appear in the title bar of the corresponding Toolbox page
- Each Profile element incorporated into an MDG Toolbox page enables a context menu option to synchronize the Tagged Values and Constraints of all objects created from it
- The sequence of Toolbox **pages** in the Toolbox is determined by the sequence of their Stereotype elements in the Profile diagram or Profile package; if you create and save the Profile from a:
  - Diagram, the Toolbox page sequence is determined by the **Z-order** of the Stereotype elements on the diagram - the higher the Z-order number of the Stereotype element, the further down the Toolbox its Toolbox page is placed; if you change the Z-order of a Stereotype element in the diagram, it changes the position of the element's page on the Toolbox
  - Package in the Project Browser, the Toolbox page sequence is determined by the **list order** of the Stereotype elements in the package - the Toolbox page for the first listed element is at the top of the Toolbox; if you re-order the elements in the Project Browser, you produce the same re-ordering of pages in the Toolbox

### Learn more

- [Synchronize Tags and Constraints](#) <sup>[1473]</sup>
- [Z Order Elements](#) <sup>[844]</sup>
- [Diagram Context Menu](#) <sup>[778]</sup> (**Modify Z-Order** option)

### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Modeling Languages | Build a Technology | Create a Custom Toolbox**

#### 7.3.3.4.1.1 Toolbox Page Attributes

When you create a Stereotype element to define a Toolbox page in an MDG Technology, you can add a number of attributes to control how the page itself behaves in the Diagram Toolbox. The Stereotype element can be one of several that extend the *ToolboxPage* Metaclass.

The attributes you can add are:

- **Icon** - see [Assign Icons for Toolbox Items](#) <sup>[1565]</sup>
- **ImagesOnly** - if you set **Initial Value** to **true**, the Toolbox page displays without the text labels next to the icons
- **isCollapsed** - if you set **Initial Value** to **true**, the Toolbox page is initially minimized

- **isCommon** - if you set **Initial Value** to **true**, the Toolbox page is common to all defined Toolboxes while your technology is active; the page is initially displayed as collapsed
- **isHidden** - see [Create Hidden Sub-Menus](#) <sup>[1564]</sup>

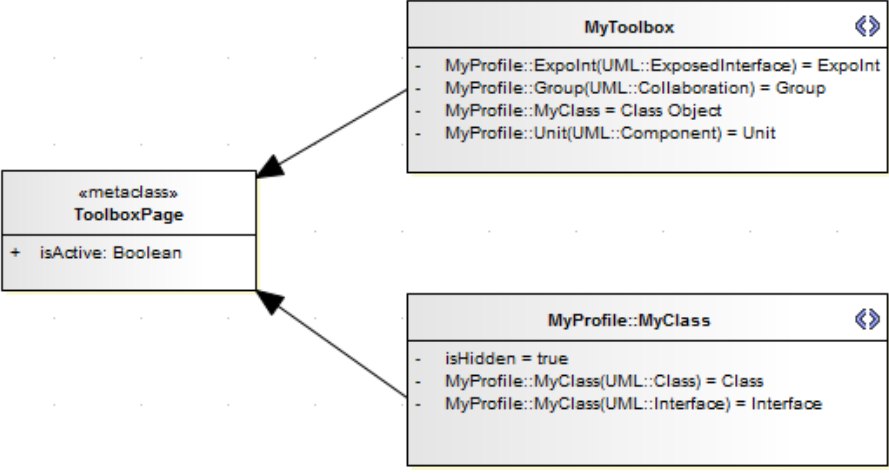
#### 7.3.3.4.2 Create Hidden Sub-Menus

When you create items on a Toolbox page, some of them might be very similar and be based on the same type of Metaclass. For example, there are many different types of Action element and, in BPMN 2.0, you can create each type of Event element either stand-alone or edge-mounted on another element. Rather than populate a Toolbox page with every variation, you can create a 'base' Toolbox item and offer a choice of variant from a **sub-menu**, which is displayed when the base item is dragged onto the diagram but is otherwise **hidden**. This technique is very useful for 'disambiguating' Stereotypes that can be applied to multiple Metaclasses.

In the submenu, you define just the variant types (as for the Action element list). However, if the variant also has a *ToolboxItemImage* defined for it, that icon is displayed against the variant name in the sub-menu (as for the BPMN 2.0 Events). You can also use this method to **specifically** define icons that will be applied to the submenu options.

##### Define a hidden sub-menu

Step	Action	See also
1	<p>Create a Stereotype element on the same diagram as the ToolboxPage Metaclass, with a name prefixed by the Profile name (this is mandatory). For example:</p> <pre>My Profile : My Submenu</pre> <p>The sub-menu element can have an alias.</p>	<a href="#">Add Stereotypes and Metaclasses</a> <sup>[1488]</sup>
2	<p>In this sub-menu Stereotype element, create the attribute <i>isHidden</i> with an initial value of <b>true</b>.</p> <p>For each sub-menu item, add an attribute to identify that item. Set the <b>Initial Value</b> to the name to display in the menu. For example, if the «<i>MySubmenu</i>» stereotype could be applied to a UML Class or UML Interface, the attributes for these two options would be:</p> <pre>My Profile : My Submenu( UML : Class )      I n i t i a l   V a l u e = C l a s s My Profile : My Submenu( UML : Interface )    I n i t i a l   V a l u e = I n t e r f a c e</pre>	<a href="#">Create Toolbox Profiles</a> <sup>[1561]</sup>
3	<p>Create a second Stereotype element and define an attribute with the same name as the sub-menu Stereotype element, and with the initial value of the text to display in the Toolbox item. For example:</p> <pre>My Profile : My Submenu = C l a s s   O b j e c t</pre> <p>Define additional attributes for the rest of the items in the Toolbox, as normal.</p>	<a href="#">Create Toolbox Profiles</a> <sup>[1561]</sup>
4	<p>Create &lt;&lt;Extension&gt;&gt; relationships between each Stereotype element and the ToolboxPage Metaclass element, as illustrated.</p>	

Step	Action	See also
	 <p>When this Profile is in use, and when the <i>Class Object</i> item is dragged onto a diagram from the Toolbox, the hidden menu displays giving the choice of Class or Interface; on selection, the element is dropped onto the diagram.</p>	
5	If no icon has been assigned to the Toolbox item from existing definitions, and you want to display one, define the image as a <code>ToolboxItemImage</code> icon.	<a href="#">Assign Icons To Toolbox Items</a> <small>1565</small>

#### 7.3.3.4.3 Assign Icons To Toolbox Items

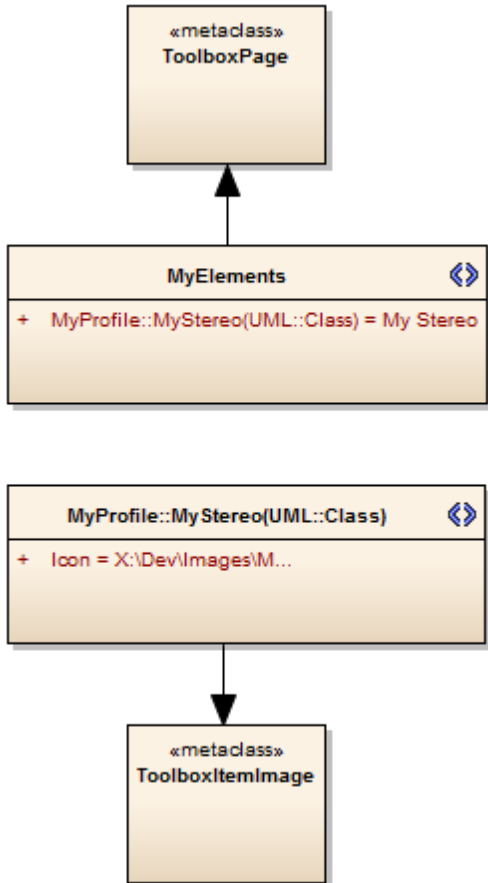
When you create a stereotyped model element to define an element or connector that is represented in a Diagram Toolbox page, you can define the image that is displayed against both the element name in the Project Browser **and** the element or connector type in the Toolbox page, by assigning the special attribute **icon** to the Stereotype element.

This image definition **for the Toolbox item** can be overridden or replaced by extending the *ToolboxItemImage* Metaclass (see below), a process that is generally optional. However, if you want to show an icon **against an item on a hidden sub-menu**, you need to use this method; the system picks up the *ToolboxItemImage* definition as the icon for the hidden menu item.

If you do not use either the **icon** attribute or the *ToolboxItemImage* Metaclass to define the Toolbox icon, the image defaults to the one used for the **standard UML** model element that has been extended. If there is no such image, the icon uses the system default generic 'Toolbox Item' image.

#### Extend the *ToolboxItemImage* Metaclass

Step	Action	See also
1	Create a new Stereotype element in the same Toolbox profile as the Toolbox item.	

Step	Action	See also
2	<p>Give the Stereotype element the same name as the element that it is assigning an image to; for example:</p> <pre>MyProfile::MyStereo(UML::Class)</pre>	
3	<p>Give the Stereotype element the special attribute <i>Icon</i> with <b>Initial Value</b> set to the full path and file name of the image to be used.</p> <p>The icon image is a 16x16 pixel bitmap file; for a transparent background use light grey - RGB(192,192,192).</p>	<a href="#">Special Attributes</a> <sup>[1503]</sup>
4	<p>Create a Metaclass element named <i>ToolboxItemImage</i> and create an Extension association from the Stereotype element to this Metaclass.</p>  <pre> classDiagram     class ToolboxPage["«metaclass»\nToolboxPage"]     class MyElements["MyElements"]     class MyProfile_Stereo["MyProfile::MyStereo(UML::Class)"]     class ToolboxItemImage["«metaclass»\nToolboxItemImage"]      ToolboxPage &lt; -- MyElements     MyElements --&gt; MyProfile_Stereo : + MyProfile::MyStereo(UML::Class) = MyStereo     MyProfile_Stereo --&gt; ToolboxItemImage : + Icon = X:\Dev\Images\I...   </pre>	

#### Learn more

- [Create Hidden Sub-Menus](#) <sup>[1564]</sup>

#### 7.3.3.4.4 Override Default Toolboxes

When you are creating a diagram of one of the inbuilt diagram types, the system displays a Diagram Toolbox based on the corresponding default Toolbox Profile. If you have customized a diagram type, it will still apply the system default Toolbox for the base diagram type that you have extended, unless you override that default with an alternative Toolbox that you might have created yourself. For example, you might have your own version of the UML::Class Toolbox that you want to be displayed every time a Class diagram is opened, when your technology is active.

##### Access    **Right-click Toolbox Profile diagram | Properties > General**

To replace a system default Toolbox with one of your own, in the **Notes** field type a *RedefinedToolbox* clause. For example:

```
RedefinedToolbox=UML::Class; Alias=Class; Notes=Structural elements for
Class diagrams;
```

This states that the Toolbox defined by this Profile replaces the system Toolbox *UML::Class* as the default Toolbox for all UML Class diagrams.

##### Names of system default toolboxes that can be overridden

- |                      |                           |
|----------------------|---------------------------|
| • UML::Activity      | • Extended::Analysis      |
| • UML::Class         | • Extended::Custom        |
| • UML::Communication | • Extended::DataModeling  |
| • UML::Component     | • Extended::Maintenance   |
| • UML::Composite     | • Extended::Requirements  |
| • UML::Deployment    | • Extended::UserInterface |
| • UML::Interaction   | • Extended::WSDL          |
| • UML::Metamodel     | • Extended::XMLSchema     |
| • UML::Object        |                           |
| • UML::Profile       |                           |
| • UML::State         |                           |
| • UML::Timing        |                           |
| • UML::UseCase       |                           |

#### 7.3.3.4.5 Elements Used in Toolboxes

When you are creating Toolbox pages for your MDG Technology, you can incorporate both standard UML elements **and** new elements that you have created by **extending** the UML elements. You define the elements you want to use in the Toolbox Profile. The table below lists the names you use to identify either:

- The standard elements to include in the Toolbox page or
- The standard elements you are extending to define new elements to include in the Toolbox page

Each name you list in the Toolbox Page Stereotype elements is preceded by the namespace **UML::**. The text in red indicates the **label name** displayed in the default Toolbox pages, where this differs in any way from the **UML::** statement text.

Element names for Toolbox Page definitions

- Action
- ActionPin
- Activity
- ActivityFinal (**Final**)
- ActivityInitial (**Initial**)
- ActivityParameter
- ActivityPartition (**Partition**)
- ActivityRegion (**Region**)
- Actor
- Artifact
- AssociationElement (**Association**)
- Boundary (for Use Cases)
- CentralBufferNode (**Central Buffer Node**)
- Change
- Choice
- Class
- Collaboration
- CollaborationOccurrence (**Collaboration Use**)
- Comment (**Note**)
- Component
- Constraint
- Datastore
- Decision
- DeploymentSpecification (**Deployment Specification**)
- Device
- DiagramLegend (**Diagram Legend**)
- DiagramNotes (**Diagram Notes**)
- DocumentArtifact (**Document Artifact** or **Document**)
- Entity (**Information**)
- EntityObject (**Entity**)
- EntryPoint (**Entry**)
- Enumeration
- ExceptionHandler (**Exception**)
- ExecutionEnvironment (**Execution Environment**)
- ExpansionRegion
- ExitPoint (**Exit**)
- InteractionState (**State/Continuation**)
- Interface
- InterruptibleActivityRegion
- Issue
- Junction
- Lifeline
- MergeNode (**Merge**)
- MessageEndPoint (**Endpoint** or **Message Endpoint**)
- MessageLabel (**Message Label**)
- Metaclass
- Node
- Object
- ObjectBoundary (**Boundary**)
- ObjectControl (**Control**)
- ObjectEntity (**Entity**)
- Package
- PackagingComponent
- Part
- Port
- Primitive
- PrimitiveType
- Process
- Profile
- ProvidedInterface (**Expose Interface**)
- ReceiveEvent (**Receive**)
- Requirement
- RobustBoundary (**Boundary**)
- RobustControl (**Control**)
- RobustEntity (**Entity**)
- Screen
- SendEvent (**Send**)
- SequenceBoundary (**Boundary**)
- SequenceControl (**Control**)
- SequenceEntity (**Entity**)
- Signal
- State



- Feature
- FinalState (**Final**)
- FlowFinalNode (**Flow Final**)
- ForkJoinH (**Fork/Join** - Horizontal)
- ForkJoinV (**Fork/Join** - Vertical)
- Gate (**Diagram Gate**)
- GUIElement (**UI Control**)
- HistoryState (**History**)
- Hyperlink
- InformationItem (**Information Item**)
- InitialState (**Initial**)
- InteractionFragment (**Fragment**)
- StateMachine (**State Machine**)
- StateTimeLine (**State Lifeline**)
- Stereotype
- StructuredActivity (**Structured Activity**)
- SynchState (**Synch**)
- Table
- Terminate
- TestCase (**Test Case**)
- Text
- UseCase (**Use Case**)
- UMLBoundary (**Boundary**)
- ValueTimeLine (**Value Lifeline**)

### Notes

- You can also identify standard or extended UML connectors to add to the Toolbox Page definition
- When the element items are deployed in an MDG Toolbox page, you can also synchronize the Tagged Values and Constraints of all elements created from them

### Learn more

- [Add Stereotypes and Metaclasses](#) <sup>[1488]</sup>
- [Connectors Used In Toolboxes](#) <sup>[1569]</sup>
- [Synchronize Tagged Values and Constraints](#) <sup>[1473]</sup>

#### 7.3.3.4.6 Connectors Used In Toolboxes

When you are creating Toolbox pages for your MDG Technology, you can incorporate both standard UML connectors **and** new connectors that you have created by **extending** the UML connectors. You define the connectors you want to use in the Toolbox Profile. The table below lists the names you use to identify either:

- The standard connectors to include in the Toolbox page or
- The standard connectors you are extending to define new connectors to include in the Toolbox page

Each name you list in the Toolbox Page Stereotype elements is preceded by the namespace **UML::**. The text in red indicates the **label name** displayed in the default Toolbox pages, where this differs in any way from the **UML::** statement text.

#### Connector names for Toolbox Page definitions

- Abstraction
- Aggregation (**Aggregate**)
- Assembly
- Association (**Associate**)
- AssociationClass (**Association Class**)
- ObjectFlow (**Object Flow**)
- Occurrence
- PackageImport (**Package Import**)
- PackageMerge (**Package Merge**)
- Precedes

- CallFromRecursion (**Call**)
- CommunicationPath (**Communication Path**)
- Composition (**Compose**)
- Connector
- ControlFlow (**Control Flow**)
- Delegate
- Dependency
- Deployment
- Extension
- Generalization (**Generalize** or **Inheritance**)
- InformationFlow (**Information Flow**)
- InterruptFlow (**Interrupt Flow**)
- Invokes
- Manifest
- Message
- Nesting
- NoteLink (**Note Link**)
- ProfileApplication (**Application**)
- Realization (**Realize** or **Implements**)
- Recursion
- Redefinition
- Representation
- Represents
- RoleBinding (**Role Binding**)
- SelfMessage (**Self-Message**)
- Substitution
- TagValAssociation (**Tagged Value**)
- TemplateBinding (**Template Binding**)
- TraceLink (**Trace**)
- Transition
- UCExtend (**Extend**)
- UCIinclude (**Include**)
- Usage
- UseCaseLink (**Use**)

### Notes

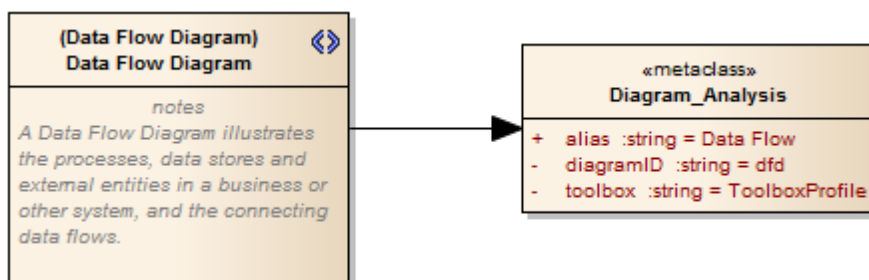
- You can also identify standard or extended UML elements to add to the Toolbox Page definition

### Learn more





- [Elements Used in Toolboxes](#) <sup>[1567]</sup>
- [Add Stereotypes and Metaclasses](#) <sup>[1488]</sup>

### 7.3.3.5 Create Custom Diagram Profiles

When you develop an MDG Technology, it is possible to create extended diagram types and include them in your MDG Technology as custom Diagram Profiles. For example, you might create a DFD Diagram Profile that defines a DFD diagram as an extension of the built-in Analysis diagram, as below:



### Create extended diagram types

Step	Action	See also
1	<p>Create a Profile, with the <b>same name</b> as the MDG Technology in which it is to be included; for example, <i>SysML</i>.</p> <p>This Profile automatically contains one child Class diagram. Depending on how many new diagram types you intend to create, you can define:</p> <ul style="list-style-type: none"> <li>• One diagram type on one child diagram</li> <li>• Several diagram types on one diagram, or</li> <li>• Several diagram types grouped on several diagrams</li> </ul> <p>In the third case, create any further child Class diagrams you need. The diagram names do not have to reflect the technology name.</p>	<a href="#">Create a Profile Package</a> 
2	<p>Open the child Class diagram and create a Stereotype element, giving it the name of the <b>custom</b> diagram type; for example, <i>BlockDefinition</i>.</p> <p>Also on the Stereotype element Properties dialog, in the <b>Notes</b> field, type a brief description of what the diagram is used for.</p> <p>When the Technology is deployed and a diagram of this custom type is being created, this description will display in the bottom right-hand corner of the New Diagram dialog.</p>	<a href="#">Add Stereotypes and Metaclasses</a> 
3	<p>Create a Metaclass element and give it the name of the selected <b>built-in</b> diagram type, with the prefix <i>Diagram_</i>.</p> <p>For example <i>Diagram_Logical</i> to customize the Class diagram type, or <i>Diagram_Use Case</i> to customize the Use Case diagram type.</p>	<a href="#">Built-In Diagram Types</a> 
4	<p>Drag an Extension connector from the Stereotype element to the Metaclass element.</p>	
5	<p>Click on the <i>Diagram_xxxx</i> <b>Metaclass</b> element, press <b>F9</b> and create any or all of the following attributes, to set properties of the custom diagram type:</p> <ul style="list-style-type: none"> <li>• <i>alias: string = Type</i> (where <i>Type</i> will appear before the word 'Diagram' on the diagram title bar; for example, <i>Block Diagram</i>)</li> <li>• <i>diagramID: string = abc</i> (where <i>abc</i> is the diagram type that will appear in the diagram <b>frame</b> label)</li> <li>• <i>toolbox: string = ToolboxName</i> (where <i>ToolboxName</i> is the name of the Toolbox Profile for the Toolbox that opens automatically each time a diagram of this type is opened)</li> <li>• <i>frameString: string = FrameFormatString</i> (where <i>FrameFormatString</i> is a string containing substitution macros for defining the frame title, with or without additional delimiters such as ( ); macros that can be used are: <ul style="list-style-type: none"> <li>• #DGMALIAS#</li> <li>• #DGMID#</li> <li>• #DGMNAME#</li> <li>• #DGMNAMEFULL#</li> </ul> </li> </ul>	<a href="#">Diagram Frame</a> 

Step	Action	See also
	<ul style="list-style-type: none"> <li>• #DGMOWNERNAME#</li> <li>• #DGMOWNERNAMEFULL#</li> <li>• #DGMOWNERTYPE#</li> <li>• #DGMSTEREO#</li> <li>• #DGMTYPE#</li> <li>• <i>swimlanes</i>: <i>string</i> = <i>Lanes=2;Orientation=Horizontal;Lane1=Title1;Lane2=Title2</i>; (where <i>Lanes</i> can be any value, but the number of <i>Lane</i>&lt;<i>n</i>&gt; values must equal the value of <i>Lanes</i>; <i>Orientation</i> can be omitted, in which case the swimlanes default to vertical)</li> <li>• <i>styleex</i>: <i>string</i> = one or more of a range of values</li> <li>• <i>pdata</i>: <i>string</i> = one or more of a range of values</li> </ul>	<a href="#">Attribute Values - styleex &amp; pdata</a> <sup>[1573]</sup>
6	Depending on what Profile package organization you adopted at step 1, and whether you need any further Stereotype-Metaclass element pairs, repeat steps 2 - 5 on this diagram or on another child diagram.	
7	Save the diagram(s) as a Diagram Profile, using the method most appropriate to the Profile package organization you have set up.	<a href="#">Export a Profile</a> <sup>[1523]</sup>
8	Add the Diagram Profile(s) to the .mts file used in the MDG Technology.	<a href="#">Add a Diagram Profile</a> <sup>[1550]</sup>

### Learning Center topics

- (Alt+F1) | [Enterprise Architect](#) | [Modeling Languages](#) | [Build a Technology](#) | [Create a Diagram Profile](#)

#### 7.3.3.5.1 Built-In Diagram Types

In customizing Enterprise Architect to better suit your needs, you might create a Profile that:

- Redefines the type of built-in child diagram created under a new composite element
- Defines the types of built-in diagram on which a Quick Linker menu offers a type of connector, or
- Extends a built-in diagram type to create a custom diagram type

In each case, you provide the **precise** name of each built-in diagram type you are working with; these names are:

- Activity
- Analysis
- Collaboration
- Component
- CompositeStructure
- Custom
- Deployment

- InteractionOverview
- Logical (for Class diagrams)
- Object
- Package
- Sequence
- Statechart
- Timing
- Use Case (note the space between the two words)

#### Learn more

- [Define Child Diagram Type](#)<sup>[1512]</sup>
- [Quick Linker Definition Format](#)<sup>[1516]</sup>
- [Create Custom Diagram Profiles](#)<sup>[1570]</sup>

#### 7.3.3.5.2 Attribute Values - *styleex* & *pdata*

When creating a diagram profile you can define a range of characteristics of the diagrams created with the profile, using the *pdata* and *styleex* attributes. If one of these attributes is defining several characteristics at once, you put the values in a single string separated by semicolons; for example:

*styleex: string = HideQuals=0;AdvanceElementProps=1;ShowNotes=1;*

**Access** Click on Metaclass element: **F9 > General**

#### **styleex: string =**

- *AdvancedConnectorProps=1*; (to show connector property strings)
- *AdvancedElementProps=1*; (to show the element property string)
- *AdvancedFeatureProps=1*; (to show the feature property string)
- *AttPkg=1*; (to show package visible Class members)
- *DefaultLang=Language*; (to set the default language for the diagram; *Language* can be one of the built-in languages such as C++ or Java, or it can be a custom language)
- *HandDraw=1*; (to apply hand drawn mode)
- *HideConnStereotype=1*; (to hide the connector stereotype labels)
- *HideQuals=0*; (to show qualifiers and visibility indicators)
- *SeqTopMargin=50*; (to set the height of the top margin on sequence diagrams)
- *ShowAsList=1*; (to make the diagram open directly into the *Diagram List*)
- *ShowMaint=1*; (to show the element Maintenance compartment)
- *ShowNotes=1*; (to show the element Notes compartment)
- *ShowOpRetType=1*; (to show the operation return type)
- *ShowTests=1*; (to show the element Testing compartment)
- *SuppConnectorLabels=1*; (to suppress all connector labels)
- *SuppressBrackets=1*; (to suppress brackets on operations without parameters)
- *TConnectorNotation=Option*; (where *Option* is one of **UML 2.1**, **IDEF1X**, or **Information Engineering**)
- *TExplicitNavigability=1*; (to show non-navigable connector ends)

- *VisibleAttributeDetail=1*; (to show attribute details on the diagram)
- *Whiteboard=1*; (to apply whiteboard mode)

**pdata: string =**

- *HideAtts=0*; (to show the element Attributes compartment)
- *HideEStereo=0*; (to show element stereotypes in the diagram)
- *HideOps=0*; (to show the element Operations compartment)
- *HideParents=0*; (to show additional parents of elements in the diagram)
- *HideProps=0*; (to show property methods)
- *HideRel=0*; (to show relationships)
- *HideStereo=0*; (to show attribute and operation stereotypes)
- *OpParams=3*; (to show operation parameters)
- *ShowCons=1*; (to show the element Constraints compartment)
- *ShowIcons=1*; (to use stereotype icons)
- *ShowReqs=1*; (to show the element Requirements compartment)
- *ShowSN=1*; (to show sequence notes)
- *ShowTags=1*; (to show the element Tagged Values compartment)
- *SuppCN=0*; (to show collaboration numbers)
- *UseAlias=1*; (to use the aliases or elements in the diagram, if available)

**Learn more**

- [Diagram List](#)<sup>[684]</sup>

### 7.3.3.6 Set Up Technology Element Images

As you define the elements available for use in your technology, you might want to represent those elements with graphical images that will be displayed on the diagrams the users create through the technology, when it is deployed in the users' model.

**Capture images to represent MDG Technology elements**

Step	Action	See also
1	Display the Image Manager and, using the <b>Add New</b> button, import suitable images into the MDG Technology development model from their source locations.	<a href="#">Image Manager</a> <sup>[860]</sup>
2	Design and create a Stereotype (UML) Profile containing (if appropriate) a stereotype definition for each element or connector to be owned by the technology.  These stereotype definitions can contain Shape Scripts that in turn incorporate the imported images.	<a href="#">Stereotype (UML) Profile</a> <sup>[1485]</sup> <a href="#">Shape Scripts</a> <sup>[1501]</sup> <a href="#">Drawing Methods</a> <sup>[1591]</sup>

Step	Action	See also
3	<p>Design and create a Toolbox Profile with stereotype elements that contain an attribute for each element or connector that can be dropped onto a diagram from the toolbox.</p> <p>These attributes identify the name of the technology element or connector, any modifying stereotype (which might incorporate the required image) and the UML or Extended element or connector on which the technology object is based.</p> <p>For example:</p> <pre>SysML: : Block ( UML: : Class )</pre> <ul style="list-style-type: none"> <li>• <i>SysML</i> is the Technology Profile</li> <li>• <i>UML::Class</i> is the UML element used as the base, and</li> <li>• <i>Block</i> is the stereotype that modifies the Class to turn it into a SysML Block element</li> </ul>	<a href="#">Toolbox Profiles</a> <sup>[1561]</sup>
4	<p>Design and create a Diagram Profile that identifies the Toolbox Profile.</p> <p>When a diagram of the type defined in the Diagram Profile is opened, it in turn opens a set of toolbox pages as defined by the Toolbox Profile.</p>	<a href="#">Diagram Profiles</a> <sup>[1570]</sup>
5	<p>Create or update the technology as required, adding the UML Profile, Diagram Profile, Toolbox Profile and Image files to the technology from the development model.</p>	<a href="#">Creating MDG Technologies</a> <sup>[1545]</sup> <a href="#">Adding UML Profile</a> <sup>[1548]</sup> <a href="#">Adding Diagram Profile</a> <sup>[1550]</sup> <a href="#">Adding Toolbox Profile</a> <sup>[1551]</sup> <a href="#">Adding Image Files</a> <sup>[1558]</sup>
6	<p>Deploy the technology as appropriate.</p> <p>When a user applies the technology to their own model, and creates a diagram under that technology, the elements they create on the diagram should be represented by the images you assigned to those elements when you created the technology.</p>	<a href="#">Deploy an MDG Technology</a> <sup>[1580]</sup>

### Notes

- It is recommended that if you create a Shape Script incorporating an MDG Technology image (step 2), you should use the fully qualified image name to avoid conflicts with images used in other technologies
- You would probably work backwards and forwards through the steps many times, adding objects as you identify the requirement for them

### 7.3.3.7 Define Validation Configuration

Using the Model Validation Configuration dialog, you can choose which sets of validation rules are and are not executed when a user performs a validation. Rather than perform this configuration manually and potentially have to change the settings every time Enterprise Architect is started and a different technology is set active, you can define the configuration settings within the MTS file.

**Access** [Project | Model Validation | Configure...](#)

#### White List

To specify a set of rules as a white-list (that is, anything added to this list is turned ON), open your MTS file in a text editor and copy and paste the following `<Model Validation>` block at the top level inside the `<MDG. Selections>` block:

```
<Model Validation>
  <RuleSet name=" BPMNRules" /> <!-- ruleset ID defined in the Project .
  DefineRuleCategory call -->
  <RuleSet name=" MVR7F0001" /> <!-- notice you can turn on/off system
  rules as well! -->
</Model Validation>
```

Ensure that the ruleset IDs do not contain any spaces.

#### Black List

To specify a set of rules as a black-list (that is, anything added to this list is turned OFF), open your MTS file in a text editor and copy and paste the following `<Model Validation>` block at the top level inside the `<MDG. Selections>` block:

```
<Model Validation isBlackList="true">
  <RuleSet name=" BPMNRules" />
  <RuleSet name=" MVR7F0001" />
</Model Validation>
```

In the examples above, "BPMNRules" is the rule-set ID defined in the Project . DefineRuleCategory call - see [Project Interface](#)<sup>[2962]</sup> for details. "MVR7F0001" is a built-in rule-set. These validation options are applied when you activate the appropriate technology. The global (default) technology has all rules turned on.

### 7.3.3.8 Incorporate Model Templates

When a user creates a model within their project, they can choose the type of model to develop from a list of standard model templates. You can also develop **custom** model templates to add to this list via your MDG Technology.

**Access** [Project Browser package context menu | Add a Model using Wizard...](#) (Ctrl+Shift+M)

[Add custom model templates to MDG Technology](#)



Step	Action	See also
1	<p>Create a package that contains all sub-packages, diagrams, elements, notes and information links that you want to provide in your model template.</p> <p>See the <i>EAExample.eap</i> model for illustrations of what you might include, or create a model from a standard template and see what is generated.</p> <p>As a model template, the package would be self contained and not contain any dependencies or other links to elements outside the package.</p>	<a href="#">Model Wizard</a> <sup>[753]</sup>
2	Export your package to XML.	<a href="#">Export to XML</a> <sup>[475]</sup>
3	<p>Create a reference to the XML file in the MTS file; open your MTS file in a text editor and copy and paste the following <code>&lt;Model Templates&gt;</code> block at the top level inside the <code>&lt;MDG. Selections&gt;</code> block:</p> <pre> &lt;Model Templates&gt;   &lt;Model name=" Template Name"     description=" This is the description."     location=" MyTemplatePackage.xml "     default=" yes"     icon = " 34"     filter= " Filter Name"     isFramework=" false" /&gt; &lt;/ Model Templates&gt; </pre> <p>You can include as many <code>&lt;Model Templates&gt;</code> blocks in your MTS file as you have model templates.</p> <p>The attributes within a <code>&lt;Model Templates&gt;</code> block have the following meanings:</p> <ul style="list-style-type: none"> <li>• <b>Model name:</b> The name of the model template to show in the Model Wizard, which displays when you create a new model or when you execute the <b>Add a New Model using Wizard</b> menu option</li> <li>• <b>description:</b> The text to display in the Model Wizard when the name is selected</li> <li>• <b>location:</b> The path of the XML file that contains the XML export of the model template package, relative to the location of the MDG Technology file; if the XML file is in the same folder as the technology file then this just contains the file name</li> <li>• <b>default:</b> Contains either <b>yes</b> to indicate that the model template is checked by default, or <b>no</b> to indicate that the model template is unchecked by default</li> <li>• <b>icon:</b> Contains an index to Enterprise Architect's base icons list; to show the appropriate view icon, use one of the following values: <b>29</b> = Use Case, <b>30</b> = Dynamic; <b>31</b> = Class; <b>32</b> = Component; <b>33</b> = Deployment; <b>34</b> = Simple</li> <li>• <b>filter:</b> If you have a large number of model templates, you can group them on the Model Wizard by giving all the model templates in the same group the same filter name; the filter name given appears in the <b>Select from:</b> list box in the Model Wizard</li> <li>• <b>isFramework:</b> Defines the possible uses of a model pattern; there are three possible values:</li> </ul>	<a href="#">Working with MTS Files</a> <sup>[1559]</sup>

Step	Action	See also
	<ul style="list-style-type: none"> <li>• <b>isFramework="true"</b> - never strip GUIDs; the Pattern is intended as a re-usable package for <b>any</b> model</li> <li>• <b>isFramework="optional"</b> - prompt to strip GUIDs; the Pattern is intended as a re-usable package, but the user can choose</li> <li>• <b>isFramework="false"</b> - always strip GUIDs (the default, if not stated); the Pattern could be applied multiple times within <b>the one</b> model</li> </ul>	

#### Learn more

- [EA\\_OnRetrieveModelTemplate](#)<sup>[3046]</sup>

### 7.3.3.9 Add Import/Export Scripts

In Enterprise Architect, it is possible to import packages from and export (or Publish) packages to external files in a range of XML and XML formats. You can also incorporate this facility in your MDG Technology, adding a script that contains your own Extensible Stylesheet Language Transformation (XSLT) to convert between the file formats.

#### Incorporate an Export (Publish) script

Step	Description	See also
1	In your preferred editor, create an XSLT to convert from the source format (as listed on the Publish Model Package dialog) into the target format you are generating.	
2	In Enterprise Architect, open the Scripter window and create a script under your preferred script engine as a Normal script.  Cut and paste the XSLT into the script editor.	<a href="#">Scripting</a> <sup>[2791]</sup>
3	Add the script to your MDG Technology, in the MDG Technology Creation Wizard.	<a href="#">Add Scripts</a> <sup>[1556]</sup>
4	Make any additions to the technology .MTS file you require, then use the MDG Technology Creation Wizard again to fully generate the technology .XML file.  Open the technology .XML file ( <b>not</b> the .MTS file) in a text editor and locate the <Script> section.	<a href="#">Working with MTS Files</a> <sup>[1559]</sup> <a href="#">Create MDG Technologies</a> <sup>[1545]</sup>
5	Edit the <Script> line to set the appropriate name, type and language: <ul style="list-style-type: none"> <li>• <i>name</i> is the technology option text to display in the <b>Extensions   Publish</b> menu in the Enterprise Architect menu bar</li> <li>• <i>type</i> is the word <b>Publish</b>- followed by the name of the file format to export, as listed on the Publish Model Package dialog</li> </ul>	

Step	Description	See also
	<ul style="list-style-type: none"> <li><i>language</i> is <b>XSLT</b></li> </ul> <p>For example:</p> <pre>&lt;Script   name="Your Technology"   type="Publish-UML 2.1(XMI 2.1)"   language="XSLT"&gt; &lt;Content   xmlns:dt="urn:schemas-microsoft-com:datatypes" dt: dt="bin.base64"&gt; &lt;/Content&gt; &lt;/Script&gt;</pre>	
6	Save the MDG technology .XML file, and deploy it on your system.	<a href="#">Deploy An MDG Technology</a> <small>[1580]</small>

#### Incorporate an Import script

Step	Description	See also
1	In your preferred editor, create an XSLT to convert from the source format into the target XML format.	
2	<p>In Enterprise Architect, open the Scripter window and create a script under your preferred script engine as a Normal script.</p> <p>Cut and paste the XSLT into the script editor.</p>	<a href="#">Scripting</a> <small>[279]</small>
3	Add the script to your MDG Technology, in the MDG Technology Creation Wizard.	<a href="#">Add Scripts</a> <small>[1556]</small>
4	<p>Make any additions to the technology .MTS file you require, then use the MDG Technology Creation Wizard again to fully generate the technology .XML file.</p> <p>Open the technology .XML file (<b>not</b> the .MTS file) in a text editor and locate the &lt;Script&gt; section.</p>	<a href="#">Working with MTS Files</a> <small>[1555]</small> <a href="#">Create MDG Technologies</a> <small>[1545]</small>
5	<p>Edit the &lt;Script&gt; line to set the appropriate name, type and language:</p> <ul style="list-style-type: none"> <li><i>name</i> is the technology option text to display in the <b>Extensions   Import</b> menu in the Enterprise Architect menu bar</li> <li><i>type</i> is the word <b>Import</b>- followed by the name of the XML file format to generate, as listed on the Publish Model Package dialog</li> </ul>	

Step	Description	See also
	<ul style="list-style-type: none"> <li><i>language</i> is <b>XSLT</b></li> </ul> <p>For example:</p> <pre>&lt;Script   name="Your Technology"   type="Import - UML 2.1 (XML 2.1)"   language="XSLT"&gt; &lt;Content   xmlns:dt="urn:schemas-microsoft-com:datatypes" dt: dt="bin.base64"&gt; &lt;/Content&gt; &lt;/Script&gt;</pre>	
6	Save the MDG technology .XML file, and deploy it on your system.	<a href="#">Deploy An MDG Technology</a> <sup>[1580]</sup>

#### Notes

- Create the content of your scripts in XSLT 1.0

#### Learn more

- [XML Import and Export](#)<sup>[473]</sup>
- [Export to XML](#)<sup>[475]</sup>
- [Publish Model Package](#)<sup>[476]</sup>
- [Import from XML](#)<sup>[478]</sup>

### 7.3.3.10 Deploy An MDG Technology

An MDG Technology can be **deployed** in one of two ways: as a .xml file or from an **Add-In**.

#### Deploy From a .xml File

To deploy your technology as a **file**, you have a number of choices:

- Import the technology .xml file into the %APPDATA%\ Sparx Systems\ EA\ MDGTechnologies folder (for your personal use)
- Import the technology .xml file into the Resources window (for all project users to access)
- Copy the file to the MDGTechnologies folder under your Enterprise Architect installation directory (by default this is C:\ Program Files\ Sparx Systems\ EA). When you restart Enterprise Architect, your MDG Technology is deployed
- Copy the file to any folder in your file system, including network drives - use the Enterprise Architect **Settings | MDG Technologies...** menu option, press the **Advanced** button and add the folder to the Technologies path; this deployment method enables you to quickly and easily deploy a technology to all Enterprise Architect users on a LAN
- Upload the file to an internet or intranet location: use the Enterprise Architect **Settings | MDG**

**Technologies...** menu option, press the **Advanced** button and add the URL to the **Technologies** path; this deployment method enables you to quickly and easily deploy a technology to an even wider group of Enterprise Architect users

### Deploy From an Add-in

To deploy your technology from an **Add-In**, you must write an *EA\_OnInitializeTechnologies* function. The following example is written in VB.Net:

```
Public Function EA_OnInitializeTechnologies(ByVal Repository As EA.
Repository) As Object
    EA_OnInitializeTechnologies = My.Resources.MyTechnology
End Function
```

### Learn more

- [Import MDG Technologies to Model](#) <sup>[1480]</sup>
- [EA\\_OnInitializeTechnologies](#) <sup>[3089]</sup>

### Learning Center topics

- (Alt+F1) | **Enterprise Architect** | **Modeling Languages** | **Build a Technology** | **Deploy a Technology**










## 7.3.4 Developing Programming Languages

You can make use of a range of established programming languages in Enterprise Architect, but if these are not suitable to your needs you can develop your own. You would then apply it to your models through an MDG Technology that you might develop just for this purpose, or for broader purposes. After developing the language, you could also write MDA Transformation templates to convert a Product Independent Model or a model in another language into a model for your new language, or vice-versa.

**Access**   **Tools** | **Source Code Generation Templates** (Ctrl+Shift+P)

### Develop a Programming Language

Step	Description	See also
1	In the Code Template Editor, click on the <b>New Language</b> button and, on the Programming Languages Datatypes dialog, click on the <b>Add Product</b> button.  Enter your new programming language name and define the datatypes for it. You cannot access the new language in the Code Template Editor until at least one datatype has been added to the language.	<a href="#">Data Types</a> <sup>[1171]</sup>
2	After you have defined all the datatypes you need, click on the <b>Close</b> button, select the language in the <b>Language</b> field of the Code Template Editor, and start to edit or create the code templates for the new language.  The code templates define how the system should perform: <ul style="list-style-type: none"> <li>• Forward code engineering of your models in the new language</li> </ul>	<a href="#">Code Template Framework</a> <sup>[1631]</sup> <a href="#">Code and Transform Templates</a> <sup>[1632]</sup> <a href="#">Create Custom Templates</a> <sup>[1701]</sup>



Step	Description	See also
	<ul style="list-style-type: none"> <li>Behavioral Code generation (if this is appropriate)</li> </ul>	<a href="#">Customize Base Templates</a> 
3	<p>If you wish, you can also define source code options for your new language. These are additional settings for the language that are not covered by the data types or code templates, that help define how the system handles that language when generating and reverse-engineering code.</p> <p>The code options are made available to your models only through an MDG Technology.</p>	<a href="#">Define Code Options</a> 
4	<p>Defining a grammar for your language is an optional step that provides two primary benefits:</p> <ul style="list-style-type: none"> <li>Reverse engineering of existing code into your model</li> <li>Synchronization during code generation so that changes made to the file since it was last generated are not lost.</li> </ul> <p>To access the grammar editor select <b>Analyzer</b>   <b>Grammar Editor</b>.</p>	<a href="#">Grammar Framework</a> 
5	<p>If you intend MDA transformations to be made to (or from) your new programming language, you can also edit and create transformation templates for it. The process of creating transformation templates is very similar to that for creating code templates.</p>	<a href="#">Edit Transformation Templates</a>  <a href="#">Write Transformations</a> 
6	<p>Having created the datatypes, code templates, code options, grammar and transformation templates for your new language, you can incorporate and distribute them in an MDG Technology.</p>	<a href="#">Create MDG Technologies</a>  <a href="#">Add Code Modules</a>  <a href="#">Add MDA Transforms</a>  <a href="#">Deploy An MDG Technology</a> 



### 7.3.5 Shape Scripts

The elements and connectors you initially use in modeling conform to the **standard UML notation** in terms of shape, color and labeling. You can, however, extend the standard objects to create new ones, and **customize the appearance** of these new objects using **Shape Scripts** to define the exact feature you want to impose on the default - or main - shape. You create a Shape Script in a dedicated scripting language, to define the new shape, orientation, color and labeling of the element or connector. Each script is associated with a **stereotype**, and every element or connector that has that stereotype will adopt the appearance defined by the Shape Script.

If you want to standardize the appearance, to apply to many elements, you can attach the Shape Script to an attribute of a Stereotype **element** in an MDG Technology Stereotype **Profile**.

#### Learn more

- [Getting Started with Shape Scripts](#) 
- [Shape Editor](#) 

- [Write Scripts](#)  <sup>1585</sup>
- [Example Scripts](#)  <sup>1614</sup>
- [Creating an MDG Technology](#)  <sup>1527</sup>

#### Learning Center topics





- (Alt+F1) | **Enterprise Architect | Modeling Languages | Defining Shapes | Introduction to Shape Scripts**
- (Alt+F1) | **Enterprise Architect | Modeling Languages | Build a Profile | Defining an Image**


### 7.3.5.1 Getting Started With Shape Scripts

As Shape Scripts are associated with stereotypes, you define them through the Stereotypes tab of the UML Types dialog; each stereotype can have one Shape Script. The process of setting up a Shape Script is quite simple yet very flexible.

**Access**   **Settings | UML Types > Stereotypes**

#### Shape Script Process

Step	Action	See also
1	Select the stereotype to which to attach the Shape Script, from the list on the right of the dialog.  You select an existing stereotype, but if a suitable one is not available you can create a new stereotype that, once saved, displays in the list and can be selected.	<a href="#">Custom Stereotypes</a>  <sup>1461</sup>
2	In the Override Appearance panel, select the <b>Shape Script</b> radio button and then click on the <b>Assign</b> button.  The Shape Script Editor displays.	<a href="#">Shape Script Editor</a>  <sup>1584</sup>
3	Type or copy the script into the Edit window.  To review the shape in the Preview panel, click on the <b>Refresh</b> button.	<a href="#">Writing Scripts</a>  <sup>1585</sup>
4	If you define a composite Shape Script (a main shape with decorations and labels, or separate parts such as a connector with source-end and target-end shapes), click on the <b>Next Shape</b> button to page through the components of the shape, in the Preview panel.	<a href="#">Example Scripts</a>  <sup>1618</sup>
5	Once you have finished writing your Shape Script, click on the <b>OK</b> button to return to the Stereotypes tab.  Then click on the <b>Save</b> button to save the Shape Script and its assignment to the stereotype.	

Step	Action	See also
6	<p>Drag and drop the appropriate standard UML element or connector into your diagram. The object will be of the type you selected as the <b>Base Class</b> of the stereotype.</p> <p>Right-click on the object and select the <b>Properties</b> context menu option.</p> <p>On the Properties dialog, click on the <b>Stereotype</b> drop-down arrow, select the stereotype you created and click on the <b>OK</b> button.</p> <p>The object's shape now reflects the Shape Script assigned to the stereotype.</p>	<a href="#">Custom Stereotypes</a> 

### Notes

- Using a Shape Script to modify an element's appearance makes some of the normal **Appearance** context menu options redundant for that element, so they will be disabled
- Font selection is not supported in Shape Scripts because the best user experience is achieved by allowing the user to set fonts themselves
- UML defines that the standard mechanism for extending the syntax of UML is through Profiles; for this reason Shape Scripts can not be applied to any element independently of a stereotype
- Shape Scripts do not currently support:
  - Looping constructs
  - String Manipulation
  - Arithmetical Operations
  - Variable declaration

### 7.3.5.2 Shape Editor


When you create a Shape Script through the Stereotypes tab of the UML Types dialog, you write the script using the **Shape Editor**. This provides the facilities of the **Common Code Editor**, including **Intelli-sense** for Shape Script attributes and functions.

**Access** **Settings | UML Types > Stereotypes (specify stereotype): Shape Script + Assign** or  
**Settings | UML Types > Stereotypes (specify stereotype): Shape Script + Edit**

### Editor Options

Field	Usage	See also
<b>Format</b>	Click on the drop-down arrow and select the Shape Script version (currently only <b>EAShapeScript 1.0</b> is available).	
<b>Import</b>	<p>Click on this button to import a Shape Script from a text file (.txt). A file browser displays through which you can locate the file to import.</p> <p>When you have located and selected the file, click on the <b>Open</b> button to import the script into the editing panel.</p>	



Field	Usage	See also
<b>Export</b>	Click on this button to export a Shape Script to a text file. A file browser displays through which you can specify the file to export to.  When you have identified the file, click on the <b>Save</b> button to complete the export and return to the Shape Editor.	
<b>&lt;editing panel&gt;</b>	Type the script commands in this panel.	<a href="#">Write Scripts</a> 
<b>OK</b>	Click on this button to exit from the Shape Editor.  To <b>SAVE</b> your Shape Script, click on the <b>Save</b> button on the Stereotypes tab.	<a href="#">Getting Started With Shape Scripts</a> 
<b>Next Shape</b>	If you have a shape made up of different components, click on this button to rotate through the multiple shape definitions in the Preview panel.	
<b>Refresh</b>	Click on this button to parse your script and display the result in the Preview window.	

#### Learn more

- [Editing Source Code](#) , for more information on Intelli-sense and the Common Code Editor

### 7.3.5.3 Write Scripts

To create an alternative representation for an element or connector, you write a Shape Script that defines the size, shape, orientation and color of the representation. A Shape Script contains a number of sections for defining different aspects of the shape; for an element these include:

- Main object
- Labels
- Decoration (for example, a Document element might contain an icon depicting a document)

For a connector the sections include:

- Main object
- Shape Source
- Shape Target
- Labels

Shape Scripts operate on the basis that the default (UML) representation is used **unless** the script contains an alternative definition. That is:

- If you have a Shape Script containing just a decoration, this decoration is added on top of the **normally-drawn** object
- If you have an **empty** shape routine, it overrides the default; so, a blank 'shape label' prevents the creation of the normal floating text label for elements that have them


You can also comment your scripts using C-style comments; for example:

```
// C Style Single Line comment
/* Multi Line
comment supported */
```

Scripting is not case-sensitive: 'Shape' is the same as 'shape'.

### Script Structure

Layout	Description	See also
Example of Element Script Layout	<pre>shape main {     // draw the object }  shape label {     // draw a floating text label }  decoration &lt;identifier&gt; {     // draw a 16x16 decoration inside the     object }</pre> <p>The &lt;identifier&gt; string is an alphanumeric word.</p>	
Example of Connector Script Layout	<pre>shape main {     // draw the line }  shape target {     // draw the shape at the target end }  shape source {     // draw the shape at the source end }  label &lt;positionLabel&gt; {     // define the text for the label }</pre> <p>The &lt;positionLabel&gt; string can be any of:</p> <ul style="list-style-type: none"> <li>• lefttoplabel</li> <li>• leftbottomlabel</li> <li>• middletoplabel</li> <li>• middlebottomlabel</li> </ul>	

Layout	Description	See also
	<ul style="list-style-type: none"> <li>• righttoplabel</li> <li>• rightbottomlabel</li> </ul>	
<b>Sub-shapes</b>	<p>A shape can have Sub-shapes, which must be declared after the main Shape Script, but called from the Method commands.</p> <p>This is an example of the ordering for declarations:</p> <pre> shape main {     // Initialisation Attributes - these must be     // before drawing commands     noshadow = "true";     h_align = "center";      //drawing commands (Methods)     rectangle(0,0,100,100);     println("foo bar");      // call the sub-shape     addsubshape("red", 20, 70);      // definition of a sub-shape     shape red     {         setfillcolor(200, 50, 100);         rectangle(50,50,100,100);     }      //definition of a label     shape label     {         setOrigin("SW", 0, 0);         println("Object: #NAME#");     }      //definition of a Decoration     decoration triangle     {         // Draw a triangle for the decoration         startpath();         moveto(0, 30);         lineto(50, 100);         lineto(100, 0);          endpath();         setfillcolor(153, 204, 255);         fillandstrokepath();     } </pre> <p>The shape resulting from this script is:</p> 	<a href="#">Drawing Methods</a> <small>1597</small>

Layout	Description	See also
<b>Order of declaration</b>	Shapes can consist of Attribute declarations, Command calls and Sub-shape definitions, which must appear in that order; that is, Attribute declarations must appear before all Command calls and Sub-shape definitions must appear last.	<a href="#">Shape Attributes</a> <sup>[1588]</sup> <a href="#">Sub-Shapes</a> <sup>[1603]</sup>

#### Learn more

- [Getting Started With Shape Scripts](#) <sup>[1583]</sup>
- [Color Queries](#) <sup>[1598]</sup>
- [Conditional Branching](#) <sup>[1598]</sup>
- [Query Methods](#) <sup>[1599]</sup>
- [Display Element/Connector Properties](#) <sup>[1599]</sup>
- [Add Custom Compartments to Element](#) <sup>[1604]</sup>
- [Show Composite Diagram](#) <sup>[1608]</sup>
- [Reserved Names](#) <sup>[1611]</sup>
- [Syntax Grammar](#) <sup>[1613]</sup>
- [Example Scripts](#) <sup>[1614]</sup>

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Modeling Languages | Defining Shapes | Defining an Element Shape**

#### **7.3.5.3.1 Shape Attributes**

When you define a shape using a Shape Script, you define the **properties** of that shape using **attributes**. Properties include:

- The position of the shape relative to the diagram and to other elements
- The positions of components of the shape relative to the shape borders
- Whether the shape has user-editable regions
- Whether the shape can be resized, scaled, rotated or docked

#### Attribute Syntax

*attribute* "=" *value* ";"

#### Example

```

shape main
{
    // Initialisation attributes - must be before drawing commands
    noshadow = "true";
    h_align = "center";

    // drawing commands

```

```

    rectangle(0, 0, 100, 100);
    println("foo bar");
}

```

### Attributes

Attribute Name	Type	Description	See also
<b>bottomAnchorOffset</b>	<i>(int,int)</i>	When creating a Shape Script for an embedded element (such as a Port), use this attribute to offset the shape from the bottom edge of its parent.  For example: <i>bottomAnchorOffset=(0,-10)</i> ; move embedded element up 10 pixels from the bottom edge.	
<b>dockable</b>	<i>string</i>	Makes the shape default to dockable, so that it can be aligned with and joined to other elements (both other Shape Scripts and standard elements) on a diagram. You cannot reverse the dockable status with the <b>Appearance</b> menu option; to change the status, you must edit the Shape Script.  Valid values: <b>standard</b> or <b>off</b>	<a href="#">Operations on Elements</a> <sup>[939]</sup>
<b>editableField</b>	<i>string</i>	Defines a shape as an editable region of the element.  This field impacts element shapes only, line glyphs are not supported.  Valid Values: <b>alias</b> , <b>name</b> , <b>note</b> , <b>stereotype</b>	
<b>endPointY</b> , <b>endPointX</b>	<i>integer</i>	Only used for the reserved target and source shapes for connectors; this point determines where the main connector line connects to the end shapes.  Default: <b>0</b> and <b>0</b>	
<b>fixedAspectRatio</b>	<i>string</i>	Set to <b>true</b> to fix the aspect ratio. Do not use this if you do not want to fix the aspect ratio.	
<b>h_Align</b>	<i>string</i>	Affects horizontal placement of printed text and sub-shapes depending on the <b>layoutType</b> attribute.  Valid values: <b>left</b> , <b>center</b> , or <b>right</b>	
<b>layoutType</b>	<i>string</i>	Determines how sub-shapes are sized and positioned.  Valid values: <b>leftright</b> , <b>topdown</b> , <b>border</b>	<a href="#">Sub-Shapes</a> <sup>[1603]</sup>
<b>leftAnchorOffset</b>	<i>(int,int)</i>	When creating a Shape Script for an embedded element (such as a Port), use this attribute to offset the shape from the left edge of its parent.	

Attribute Name	Type	Description	See also
		For example: <i>leftAnchorOffset</i> =(10,0); move embedded element right 10 pixels from the left edge	
<b>noShadow</b>	<i>string</i>	Set to <b>true</b> to suppress the shapes shadow from being rendered.  Valid values: <b>true</b> or <b>false</b> (default= <b>false</b> )	
<b>orientation</b>	<i>string</i>	Applies to decoration shapes only, to determine where the decoration is positioned within the containing element glyph.  Valid values: <b>NW</b> , <b>N</b> , <b>NE</b> , <b>E</b> , <b>SE</b> , <b>S</b> , <b>SW</b> , <b>W</b>	
<b>preferredHeight</b>		Used by the <b>border</b> layoutType - north and south.  Used in drawing the source and target shapes for connectors to determine how wide the line is.	
<b>preferredWidth</b>		Used by the <b>border</b> layoutType - east and west.  Used by <b>leftright</b> layoutType shapes where <i>scalable</i> is <b>false</b> to determine how much space they occupy for layout purposes.	
<b>rightAnchorOffset</b>	<i>(int,int)</i>	When creating a Shape Script for an embedded element (such as a Port), use this attribute to offset the shape from the right edge of its parent.  For example: <i>rightAnchorOffset</i> =( <b>-10</b> ,0); move embedded element left 10 pixels from the right edge.	
<b>rotatable</b>	<i>string</i>	Set to <b>false</b> to prevent rotation of the shape. This attribute is only applicable to the source and target shapes for line glyphs.  Valid values: <b>true</b> or <b>false</b> (default = <b>true</b> )	
<b>scalable</b>	<i>string</i>	Set to <b>false</b> to stop the shape from being relatively sized to the associated diagram glyph.  Valid values: <b>true</b> or <b>false</b> (default= <b>true</b> )	
<b>topAnchorOffset</b>	<i>(int,int)</i>	When creating a Shape Script for an embedded element (such as a Port), use this attribute to offset the shape from the top edge of its parent.  For example:	

Attribute Name	Type	Description	See also
		<del><code>topAnchorOffset=(0,10)</code></del> ; move embedded element down 10 pixels from the top edge.	
<b>v_Align</b>	<i>string</i>	Affects vertical placement of printed text and sub-shapes depending on the <b>layoutType</b> attribute.  Valid values: <b>top</b> , <b>center</b> , or <b>bottom</b>	

### 7.3.5.3.2 Drawing Methods

When you create a shape using a Shape Script, you define the **values** of the shape using **methods**. The values include things such as:

- What the shape is - a rectangle, a line, a sphere
- The size of the shape
- The colors of the shape and borders
- The compartments and compartment text the shape has
- The text and labels displayed in and around the shape
- Whether the shape consists of or includes a captured image

You can list the valid methods (commands) for any point in a script by pressing **Ctrl+Space**.

#### Method Syntax

```
<MethodName> "(" <ParameterList> ")" ;
```

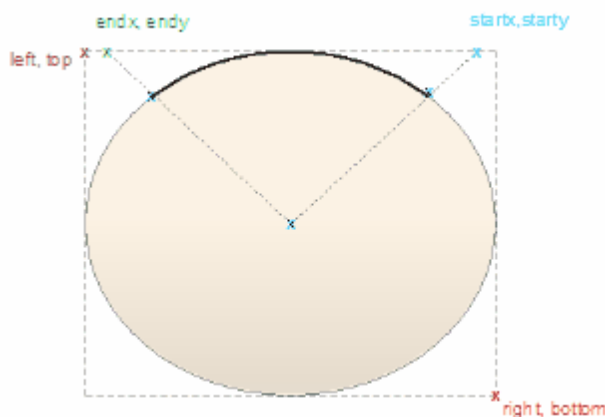
#### Example

```
shape main
{
    // Initialisation Attributes - these must be before drawing commands
    noShadow = "true";
    h_align = "center";

    // drawing commands (Methods)
    rectangle(0, 0, 100, 100);
    println("foo bar");
}
```

#### Methods

Method Name	Description	See also
<b>addsubshape</b> ( <b>string</b> <b>shapename</b> ( <b>int width</b> , <b>int height</b> ) )	Adds a sub-shape with the name <b>shapename</b> that must be defined within the current shape definition.	

Method Name	Description	See also
<b>appendcompartmenttext(</b> <b>string)</b>	<p>Appends additional strings to a compartment's text.</p> <p>The compartment the text is added to depends on the compartment name set using <b>setcompartmentname</b> (see below) prior to using <b>appendcompartmenttext</b>.</p> <p>This method must be called to have the compartment displayed.</p>	<a href="#">Add Custom Compartments to Element</a> <sup>[1604]</sup>
<b>arc(</b> <b>int left,</b> <b>int top,</b> <b>int right,</b> <b>int bottom,</b> <b>int</b> <b>startingpointx,</b> <b>int</b> <b>startingpointy,</b> <b>int</b> <b>endingpointx,</b> <b>int</b> <b>endingpointy)</b>	<p>Draws an elliptical anticlockwise arc with the ellipse having extents at <b>left</b>, <b>top</b>, <b>right</b> and <b>bottom</b>.</p> <p>The start point of the arc is defined by the intersection of the ellipse and the line from the center of the ellipse to the point ( <b>startingpointx</b>, <b>startingpointy</b>).</p> <p>The end of the arc is similarly defined by the intersection of the ellipse and the line from the center of the ellipse to the point ( <b>endingpointx</b>, <b>endingpointy</b>).</p> <p>For example:</p> <pre>Ar c ( 0, 0, 100, 100, 95, 0, 5, 0 );</pre> 	
<b>arco(</b> <b>int left,</b> <b>int top,</b> <b>int right,</b> <b>int bottom,</b> <b>int</b> <b>startingpointx,</b> <b>int</b> <b>startingpointy,</b> <b>int</b> <b>endingpointx,</b> <b>int</b> <b>endingpointy)</b>	<p>As for the arc method, except that a line is drawn from the current position to the starting point of the arc, and then the current position is updated to the end point of the arc.</p>	



Method Name	Description	See also
<b>bezierto</b> ( int controlpoint1x, int controlpoint1y, int controlpoint2x, int controlpoint2y, int endpointx, int endpointy)	Draws a bezier curve and updates the pen position.	
<b>defSize</b> (int width, int height)	<p>Sets the default size of the element.</p> <p>This can appear in IF and ELSE clauses with different values in each, and causes the element to be resized automatically each time the values change</p> <pre> if ( HasTag( " horizontal ", " true" ) ) {     def Si ze( 100, 20 );     rect angl e( 0, 0, 100, 100 ); } el se {     def Si ze( 20, 100 );     rect angl e( 0, 0, 100, 100 ); } </pre> <p>The above example sets the shape to the specified default size each time the Tagged Value <i>horizontal</i> is changed.</p> <p>When this is set, <b>(Alt+Z)</b> also resizes the shape to the defined dimensions.</p> <p>The minimum value for both <b>int width</b> and <b>int height</b> is <b>10</b>.</p>	
<b>drawnativeshape</b> ()	<p>Renders the shape in its usual, non-Shapescript notation; subsequent drawing commands are super-imposed over the native notation.</p> <p>This method is only enabled for <b>element</b> Shape Scripts; <b>line</b> Shape Scripts are not supported.</p>	
<b>drawparentshape</b> ()	<p>Used when extending non-UML Object types.</p> <p>Renders the shape as defined from a parent stereotype. Behaves identically to <b>drawnativeshape()</b> if no inherited shape script is available.</p>	<a href="#">Create Stereotypes Extending non-UML Objects</a> <small>1491</small>
<b>ellipse</b> ( int left, int top,	Draws an ellipse with extents defined by <b>left</b> , <b>top</b> , <b>right</b> and <b>bottom</b> .	

Method Name	Description	See also
<b>int right, int bottom)</b>		
<b>endpath()</b>	Ends the sequence of drawing commands that define a path.	
<b>fillandstrokepath()</b>	Fills the previously defined path with the current fill color, then draws its outline with the current pen.	
<b>fillpath()</b>	Fills the previously defined path with the current fill color.	
<b>hidelabel(     string     labelname)</b>	Hides the label specified by <b>labelname</b> , where <b>labelname</b> is one of the following values: <ul style="list-style-type: none"> <li>• <b>middletoplabel</b></li> <li>• <b>middlebottomlabel</b></li> <li>• <b>lefttoplabel</b></li> <li>• <b>leftbottomlabel</b></li> <li>• <b>righttoplabel</b></li> <li>• <b>rightbottomlabel</b></li> </ul>	
<b>image(     string     imageld,     int left,     int top,     int right,     int bottom)</b>	<p>Draws the image that has the name <b>imageld</b> in the Image Manager.</p> <p>The image must exist within the model in which the stereotype is used; if it does not already exist in the model, you must import it as reference data or select it from within a technology file.</p> <p>If the image is in a technology file, it should have a filename of the format <i>&lt;technology ID&gt;::&lt;imagename&gt;.&lt;extension&gt;</i>.</p>	<a href="#">Reference Data</a> <a href="#">Adding Images in MDG Technology</a>
<b>lineto(     int x,     int y)</b>	Draws a line from the current cursor position to a point specified by <b>x</b> and <b>y</b> , and then updates the pen cursor to that position.	
<b>moveto(     int x,     int y)</b>	Moves the pen cursor to the point specified by <b>x</b> and <b>y</b> .	
<b>polygon(     int centerx,     int centery,     int     numberofsides,     int radius,     float rotation)</b>	Draws a regular polygon with center at the point ( <b>centerx</b> , <b>centery</b> ), and <b>numberofsides</b> number of sides.	

Method Name	Description	See also
<b>print(</b> <b>string text)</b>	Prints the specified text string. You cannot change the font size or type of this text.	
<b>printfdefined(</b> <b>string</b> <b>propertyname,</b> <b>string truepart</b> <b>( ,</b> <b>string</b> <b>falsepart ) )</b>	Prints the <i>truepart</i> if the given property exists and has a non-empty value, otherwise prints the optional <i>falsepart</i> . You cannot change the font size or type of this text.	
<b>println(</b> <b>string text)</b>	Appends a line of text to the shape and a line break. You cannot change the font size or type of this text.	
<b>printwrapped(</b> <b>string text)</b>	Prints the specified text string, wrapped over multiple lines if the text is wider than its containing shape. You cannot change the font size or type of this text.	
<b>rectangle(</b> <b>int left,</b> <b>int top,</b> <b>int right,</b> <b>int bottom)</b>	Draws a rectangle with extents at <b>left</b> , <b>top</b> , <b>right</b> , <b>bottom</b> . Values are percentages.	
<b>roundrect(</b> <b>int left,</b> <b>int top,</b> <b>int right,</b> <b>int bottom,</b> <b>int</b> <b>abs_cornerwidth,</b> <b>int</b> <b>abs_cornerheight)</b>	Draws a rectangle with rounded corners, with extents defined by <b>left</b> , <b>top</b> , <b>right</b> and <b>bottom</b> . The size for the corners is defined by <b>abs_cornerwidth</b> and <b>abs_cornerheight</b> ; these values do not scale with the shape.	
<b>setcompartmentname</b> <b>(</b> <b>string)</b>	Sets a compartment name to the string provided. This method must be used <i>before</i> calling <b>appendcompartmenttext</b> (above); calling this <i>after</i> calling <b>appendcompartmenttext</b> clears any text that has already been added to the compartment.	<a href="#">Add Custom Compartments to Element</a> <sup>[1604]</sup>
<b>setdefaultcolors()</b>	Returns the brush and pen color to the default settings, or to the user-defined colors if available.	<a href="#">Color Queries</a> <sup>[1598]</sup>

Method Name	Description	See also
<b>setfillcolor</b> ( int red, int green, int blue)  <b>setfillcolor</b> ( Color newColor)	<p>Sets the fill color.</p> <p>You can specify the required color by defining RGB values or using a color value returned by <b>any</b> of the Color Queries such as:</p> <p>Get User Fill Color ( ) or Get User Border Color ( )</p> <p>If the shape script also uses <b>drawnativeshape</b>, then this color can be overridden via the Format Toolbar. In all other cases <b>setfillcolor</b> takes precedence over any color definition that applies to the element.</p>	<a href="#">Color Queries</a> <small>[1598]</small>  <a href="#">Format Toolbar</a> <small>[785]</small>
<b>setfixedregion</b> ( int xStart, int yStart, int xEnd, int yEnd)	<p>Fixes a region in a connector into which a sub-shape can be drawn, so that the sub-shape is not rescaled with the length or orientation of the connector line.</p> <p>For an example, see the Rotation Direction script in the <i>Example Scripts</i> topic.</p>	<a href="#">Example Scripts</a> <small>[1614]</small>
<b>setfontcolor</b> ( int red, int green, int blue)  <b>setfontcolor</b> ( Color newColor)	<p>Sets the font color of a text string.</p> <p>You can specify the required color by defining RGB values or using a color value returned by <b>any</b> of the Color Queries such as:</p> <p>Get User Font Color ( ) or Get User Fill Color ( )</p> <p>You can use this command with any of the text print commands.</p>	<a href="#">Color Queries</a> <small>[1598]</small>
<b>setlinestyle</b> ( string linestyle)	<p>Changes the stroke pattern for commands that use the pen.</p> <p><b>string linestyle</b>: has the following valid styles:</p> <ul style="list-style-type: none"> <li>• solid</li> <li>• dash</li> <li>• dot</li> <li>• dashdot</li> <li>• dashdotdot</li> <li>• double</li> </ul>	
<b>setorigin</b> ( string relativeTo, int xOffset, int yOffset)	<p>Positions floating text labels relative to the main shape.</p> <ul style="list-style-type: none"> <li>• <b>relativeTo</b> is one of N, NE, E, SE, S, SW, W, NW, CENTER</li> <li>• <b>xOffset</b> and <b>yOffset</b> are in pixels, not percentage values, and can be negative</li> </ul>	
<b>setpen</b> ( int red, int green, int blue,	<p>Sets the pen to the defined color and optionally sets the pen width.</p> <p>This method is only for line-drawing commands. It does not affect any text print commands.</p>	<a href="#">Color Queries</a> <small>[1598]</small>

Method Name	Description	See also
<pre>         int penwidth ) ) setpen(     Color newcolor( ,     int penwidth ) ) </pre>		
<pre> setpencolor(     int red,     int green,     int blue)  setpencolor(     Color newColor) </pre>	<p>Sets the pen color.</p> <p>You can specify the required color by defining RGB values or using a color value returned by any of the Color Queries</p> <p>Get User Fill Color ( )</p> <p>This method is only for line-drawing commands. It does not affect any text print commands.</p>	<a href="#">Color Queries</a> <small>1598</small>
<pre> setpenwidth(     int penwidth) </pre>	<p>Sets the width of the pen. Pen width should be between <b>1</b> and <b>5</b></p> <p>This method is only for line-drawing commands. It does not affect any text print commands.</p>	
<pre> showlabel(     string labelname) </pre>	<p>Reveals the hidden label specified by <b>labelname</b>, where <b>labelname</b> is one of the following values:</p> <ul style="list-style-type: none"> <li>• <b>middletoplabel</b></li> <li>• <b>middlebottomlabel</b></li> <li>• <b>lefttoplabel</b></li> <li>• <b>leftbottomlabel</b></li> <li>• <b>righttoplabel</b></li> <li>• <b>rightbottomlabel</b></li> </ul>	
<pre> startcloudpath(     puffWidth,     puffHeight,     noise) </pre>	<p>Similar to <b>StartPath</b>, except that it draws the path with cloud-like curved segments (<i>puffs</i>).</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• <i>float</i> <b>puffWidth</b> (default = <b>30</b>), the horizontal distance between puffs</li> <li>• <i>float</i> <b>puffHeight</b> (default = <b>15</b>), the vertical distance between puffs</li> <li>• <i>float</i> <b>noise</b> (default = <b>1.0</b>), the randomization of the puffs' positions</li> </ul>	
<pre> startpath() </pre>	Starts the sequence of drawing commands that define a path.	
<pre> strokepath() </pre>	Draws the outline of the previously defined path with the current pen.	

Method Name	Description	See also

### 7.3.5.3.3 Color Queries

In defining your shape, you might want to retain the fill, border and font colors you have already defined for the base shape. You can set the color definition using a **color query** to retrieve arguments for the **SetPenColor** and **SetFillColor** commands. The following queries can be used in place of arguments.

- `getUserFillColor()`
- `getUserBorderColor()`
- `getUserFontColor()`
- `getUserPenSize()`

For example:

```

shape main
{
    set fill color ( get user border color ( ) );
    set pencolor ( get user fill color ( ) );

    rectangle ( 0, 0, 100, 100 );
}

```

#### Notes

- The user colors are those that would be set on the base object if it were not being modified by the Shape Script; they would have been defined using - in order of decreasing precedence - the Format toolbar options, the Appearance options (**F4**) or the Options dialog (**Tools | Options**)
- Because the user colors are those defined for an element to which the stereotype and Shape Script are **subsequently** applied, they cannot be depicted in the Preview panel of the Shape Editor

#### Learn more

- [Drawing Methods](#) <sup>[1591]</sup>
- [Standard Colors](#) <sup>[616]</sup>
- [Changing Element Appearance](#) <sup>[948]</sup>
- [Diagram Appearance Options](#) <sup>[622]</sup>

### 7.3.5.3.4 Conditional Branching

You can incorporate condition branching in your Shape Scripts, using either the *IfElse* statement or query methods that evaluate to *True* or *False*.

When you use these conditional branching statements, you can use the **return** command to terminate execution of the script when a branch condition has been satisfied. The *Example Scripts* topic provides several examples of this, such as the *Return Statement Shape* script.

#### Learn more

- [Syntax Grammar](#) <sup>[1614]</sup> for *IfElse* statement syntax
- [Query Methods](#) <sup>[1599]</sup> for methods that can be used as the conditional expression for *IfElse* statements

- [Example Scripts](#)<sup>[1614]</sup> for an example of the use of conditional branching ()

### 7.3.5.3.5 Query Methods

When you are using **IfElse** statements in a Shape Script, the condition is usually that the object has a certain tag or property, and possibly if that tag or property has a particular value. You can set up the **conditional statement** to check for the property and value using one of the two **query methods** described below.

#### Queries

Method	Description	See also
<b>boolean HasTag(</b> <b>string tagname,</b> <b>( string</b> <b>tagvalue ) )</b>	Evaluates to <b>true</b> if the associated element has a tag with the name <i>tagname</i> .  If the second parameter <i>tagvalue</i> is provided, the tag <i>tagname</i> must be present, and the value of the tag has to be equal to <i>tagvalue</i> for the method to evaluate to <b>true</b> .	<a href="#">Tagged Values</a> <sup>[1134]</sup>
<b>boolean HasProperty(</b> <b>string</b> <b>propertyname,</b> <b>( string</b> <b>propertyvalue ) )</b>	Evaluates to <b>true</b> if the associated element has a property with the name <i>propertyname</i> .  If the second parameter <i>propertyvalue</i> is provided, the property must be present, and the value of the property has to be equal to <i>propertyvalue</i> for the method to evaluate to <b>true</b> .	<a href="#">Display Element/Connector Properties</a> <sup>[1598]</sup>

#### Learn more

- [Syntax Grammar](#)<sup>[1614]</sup>

### 7.3.5.3.6 Display Element/Connector Properties

A common component of a customized shape is a text string, which can include the name and value of one of the properties of the element or connector. To display the text, you use one of the commands:

- **print**
- **println** and
- **printwrapped**

These all take a string parameter representing the text to be displayed. The element or connector property can be added to the text using the substitution macro `#<propertyname>#`; for example:

```
println("name: #NAME#");
```

You can display several properties by issuing the commands several times, once for each property. The element and connector properties you can display are **listed below**. Additionally, you can display Tagged Values by prefixing the tag name with *TAG*, as shown:

```
print("#TAG:condition#");
```

You can also test for and display an element's **custom** properties in the same way as you do the system-

named properties; for example:

```
if(hasproperty("Name","Value"))
    ...
```

and:

```
print("#Name#");
```

### **Properties for Element Shape Scripts**

- addin - returns a value from an invoked Add-In function; syntax:

```
addin:<addin_name>, <function_name>, <parameter> [, <parameter> ...]
```

Note that in the *hasproperty()* argument, Enterprise Architect requires the hash characters for addin values:

```
if(hasproperty("#ADDIN:MyAddin,MyValue#", "TheValue")) {
```

- alias
- author
- cardinality
- classifier
- classifier.alias
- classifier.metatype
- classifier.name
- classifier.stereotype
- classifier.type
- complexity
- concurrency
- datecreated
- datemodified
- diagram.handdrawn
- diagram.mdgtype
- diagram.name
- diagram.stereotype
- diagram.type
- haslinkeddokument
- incomingedge (returns "none", "left", "right", "top", "bottom", or "multiple")
- isabstract
- isactive
- iscomposite
- isdrawcompositelinkicon
- isembedded
- isinparent



- isleaf
- islocked
- isroot
- isspec
- istagged
- isvisible
- keywords
- language
- metatype
- multiplicity
- name
- notes
- outgoingedge (returns "none", "left", "right", "top", "bottom", or "multiple")
- packagename
- parentedge ("right", "left", "top", "bottom")
- parent.metatype
- partition (returns "vertical" or "horizontal")
- persistence
- phase
- priority
- propertytype
- propertytype.alias
- propertytype.metatype
- propertytype.name
- propertytype.stereotype
- rectanglenotation
- scope
- showcomposeddiagram (returns "true" or "false")
- status
- stereotype
- subtype
- type
- version
- visibility

#### **Properties for Connector Shape Scripts**

- addin - returns a value from an invoked Add-In function; syntax:

*addin:<addin\_name>, <function\_name>, <parameter> [, <parameter> ...]*

Note that in the *hasproperty()* argument, Enterprise Architect requires the hash characters for addin values:

```
if(hasproperty("#ADDIN:MyAddin,MyValue#", "TheValue")) {
```

- alias
- diagram.connectornotation
- diagram.handdrawn
- diagram.mdgtype
- diagram.name
- diagram.stereotype
- diagram.type
- direction
- effect
- guard
- isroot
- isleaf
- name
- rotationdirection ("up", "down", "left", "right")
- source.aggregation
- source.alias
- source.changeable
- source.constraints
- source.element.name
- source.element.stereotype
- source.metatype
- source.multiplicity
- source.multiplicityisordered
- source.name
- source.qualifiers
- source.stereotype
- source.targetscope
- stereotype
- target.aggregation
- target.alias
- target.changeable
- target.constraints
- target.element.name
- target.element.stereotype
- target.metatype
- target.multiplicity
- target.multiplicityisordered
- target.name
- target.qualifiers
- target.stereotype

- `target.targetscope`
- `type`
- `weight`

#### Learn more

- [Example Scripts](#) <sup>[1614]</sup>
- [Drawing Methods](#) <sup>[1591]</sup>
- [Query Methods](#) <sup>[1599]</sup>

### 7.3.5.3.7 Sub-Shapes

When you define an element or connector shape using a Shape Script, you can build the shape from separate components, defined as **sub-shapes**. Using sub-shapes, you can create complex shapes that more closely resemble the objects that they represent.

#### Sub-shape Layout

To set the layout type you use the *layoutType* attribute, which must be set in the *initialization attributes* section of the script; in other words, before any of the methods are called. Valid values for this attribute are:

- **LeftRight** - Shapes with this layout position the sub-shapes side by side, with the first added on the left, and subsequent sub-shapes to the right
- **TopDown** - Places the sub-shapes in a vertical arrangement, with the first sub-shape added to the top and subsequent sub-shapes added below
- **Border** - This requires an additional argument to the **addsubshape** method to specify which region of the containing shape the sub-shape is to occupy: N, E, S, W or CENTER; each region can only be occupied by one sub-shape

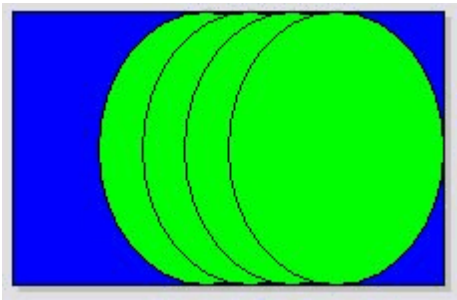
A sub-shape that is assigned to the E or W region must have its *preferredwidth* attribute specified in its declaration and, similarly, sub-shapes added to N or S must have their *preferredheight* attribute set; in this case, the values for these attributes are treated as static lengths and do not scale glyphs

#### Example

```
shape main
{
    layouttype="topdown";
    setfillcolor(0,0,255);
    rectangle(0,0,100,100);
    addsubshape("sub",50,100,20,0);
    addsubshape("sub",50,100,30,-100);
    addsubshape("sub",50,100,40,-200);
    addsubshape("sub",50,100,50,-300);

    shape sub
    {
        setfillcolor(0,255,0);
        ellipse(0,0,100,100);
    }
}
```


This script defines the following shape:



### 7.3.5.3.8 Add Custom Compartments to Element

When you display an element on a diagram in normal, rectangular format, it is possible to show a number of compartments within that frame to reveal added characteristics such as Attributes, Operations and Notes, using the diagram Properties and element Feature and Compartment Visibility dialogs. If you want to reveal other added characteristics, such as **related elements** or **Ports** and **Parts**, you can use a Shape Script to add **custom compartments** to the diagram display of the element. You would usually add this Shape Script to a Stereotype element in a Profile.

Having created a custom compartment, you can add a linked Note to the element to display the content of the compartment, as you can for the other features of the element.

**Access** **Profile Stereotype element: ( F9 ) > General | Initial Value:**  **or**  
**Settings | UML Types > Stereotypes (specify stereotype): Shape Script, Assign**

#### Add custom compartments to elements

Process	Description	See also
<b>Develop script</b>	<p>For the selected stereotype, open the Shape Editor.</p> <p>In the script, replace <code>shape main</code> with:</p> <ul style="list-style-type: none"> <li>• <code>shape ChildElement Or</code></li> <li>• <code>shape RelatedElement</code></li> </ul> <p>You can keep <code>shape main</code> if you prefer, to adjust some properties of the main element (such as color); however, the main shape then requires a call to <b>DrawNativeShape()</b> in order to work correctly.</p> <p>At this point, you can use the <b>HasProperty</b> query method to search child or related elements for specific properties (such as stereotypes) to be displayed in compartments. A Related Element Shape Script determines properties of elements that are <b>linked to</b> the current element via connectors.</p> <p>Examples of Shape Scripts that define custom compartments are provided below.</p>	<p><a href="#">Shape Editor</a> <sup>[1584]</sup></p> <p><a href="#">Drawing Methods</a> <sup>[1591]</sup></p> <p><a href="#">Query Methods</a> <sup>[1599]</sup></p> <p><a href="#">Display Element/Connector Properties</a> <sup>[1599]</sup></p>
<b>Attach Linked Note</b>	<p>You can use one of two methods to create a linked Note to display custom compartment contents:</p> <ul style="list-style-type: none"> <li>• <b>Method 1</b> (the element is currently displaying custom compartments) - highlight the related or child element name in the custom compartment, then right-click on it and select the <b>Create</b></li> </ul>	

Process	Description	See also
	<p><b>Linked Note</b> menu option; the custom compartment is automatically closed, and the linked Note added to the diagram listing all element names in that compartment</p> <ul style="list-style-type: none"> <li>• <b>Method 2</b> (the element is not necessarily showing custom compartments) - drag a Note element from the Common page of the Diagram Toolbox and link it to the element containing the custom compartment with a Notelink connector</li> </ul> <p>Right-click on the connector and select the <b>Link this note to an element feature</b> menu option, to display the Link note to element feature dialog; click on the drop-down arrow in the <b>Feature Type</b> field and click on the name of the custom compartment, such as <i>Properties</i>, then click on the <b>OK</b> button</p> <p>The contents of that compartment are displayed in the Note</p> <p>In Method 2, if the compartment is displayed the method will NOT hide the compartment. It is recommended that you use this method if the compartment is already hidden.</p> <p>Any changes you make to the list of elements in the compartment, or their names, are immediately reflected in the Note to maintain the accuracy of the displayed information.</p>	
<b>Script Example 1: Add compartment without adjusting the parent element</b>	<pre>//Add compartments for Child elements. shape ChildElement {     //Check if a child element has the property stereotype     //the compartment name to Properties.     if(HasProperty("stereotype", "property"))     {         SetCompartmentName("Properties");     }      //Check if the child element has a public scope and     //symbol to the child compartment.     if(HasProperty("scope", "public"))     {         AppendCompartmentText("+");     }      //Add the child elements name to the child compartment     AppendCompartmentText("#NAME#"); }</pre> <p>The Shape Script checks all child elements to see if they have a stereotype of <i>property</i>. If this stereotype is found, the <b>SetCompartmentName</b> function sets a compartment called <i>Properties</i>.</p> <p>The script then checks whether the child element has a public scope and, if it does, appends the + symbol.</p> <p>Finally, the <b>AppendCompartmentText</b> function adds the child element's name to the compartment.</p> <p>If a compartment has already been declared by <b>SetCompartmentName</b>,</p>	

Process	Description	See also
	<p>any additional child elements that fall under the same compartment are automatically added to it without having to declare a new compartment name (that is, all child elements with the stereotype <i>property</i> end up in the <i>Properties</i> compartment).</p>	
<p><b>Script Example 2: Adjust the color of the parent element and add child compartments</b></p>	<pre>//Shape main affects the parent shape main {     //Set the color of the parent element to red     setfillcolor(255,0,0);     //draw the parents native shape     drawnativeshape(); }  //Shape ChildElement adds Child Compartments to the parent shape ChildElement {     if(HasProperty("stereotype", "part"))     {         SetCompartmentName("Parts");     }     else if(HasProperty("stereotype", "mystereotype"))     {         SetCompartmentName("My Stereotype");     }      AppendCompartmentText("#NAME#"); }</pre> <p>The <code>shape main</code> section sets the color of the main element to <b>red</b> and adds child compartments based upon stereotyped child elements.</p> <p>The script checks whether a child element has either the stereotype value <i>"part"</i> or <i>"mystereotype"</i> applied to it. If there are multiple child elements, having a combination of <i>"part"</i> and <i>"mystereotype"</i> stereotypes, two compartments are created called <i>Parts</i> and <i>My Stereotype</i>.</p> <p>In order to display the compartments, <b>AppendCompartmentText</b> must be called to insert content into the compartment.</p> <p>Values passed to <b>SetCompartmentName</b> and <b>AppendCompartmentText</b> can not contain new line characters.</p>	
<p><b>Script Example 3: Only list child element in compartment if it is not already visible on the diagram</b></p>	<pre>shape ChildElement {     //Check if the child element is on the diagram or not     if(hasproperty("IsVisible", "false"))     {         //Create a compartment for parts.         if(hasproperty("type", "part"))         {             SetCompartmentName("Parts");         }         //Create a compartment for ports.         else if(hasproperty("type", "port"))         {             SetCompartmentName("Ports");         }     } }</pre>	

Process	Description	See also
	<pre>         }          //Add child element name to compartment.         AppendCompartmentText("#NAME#");     } } </pre> <p>This script adds custom compartments for Port and Part elements that belong to the current element but that are not visible on the current diagram.</p> <p>The <b>IsVisible</b> property returns <b>true</b> if the child element is already visible on the diagram, <b>false</b> if the child element is not visible.</p> <p>This can be used to prevent the child element from being listed in the custom compartment if it is already visible on the diagram, avoiding display of redundant information.</p>	
<b>Script Example 4:</b> <b>Display elements that are the target of a Dependency connector from the element that owns the shape script</b>	<pre> shape RelatedElement {     //Check if the current connector we are processing has     //dependency type.     if(HasProperty("Connector.Type", "Dependency"))     {         //Check if the element we are currently checking         //the target of the current connector.         if(HasProperty("Element.IsTarget"))         {             //Set the compartment Name             SetCompartmentName("dependsOn");             if(HasProperty("Element.Stereotype", "             {             }             else             {                 AppendCompartmentText("&lt;#Element.Stere             }             AppendCompartmentText("#Element.Name#")         }     } } </pre> <p>With this script, if a Class1 has a stereotype with the 'RelatedElement' Shape Script <b>and</b> Class1 is the source of a Dependency connector to the target Class2, then the name <i>Class2</i> is displayed in a compartment of Class 1, called <b>dependsOn</b>.</p>	
<b>Script Example 5:</b> <b>Display a list of Realized Interfaces within a compartment on an element</b>	<pre> shape RelatedElement {     //Check if the current connector being processed is     if(HasProperty("Connector.Type", "Realization"))     {         //Only display this compartment if the relate         //are checking is the target of the connector         //shape scripts element as the source         if(HasProperty("Element.IsTarget"))     } } </pre>	

Process	Description	See also
	<pre> {     //If the element is an interface, disp     //'realizedInterfaces' compartment     if(HasProperty("Element.Type", "Interf     {         SetCompartmentName("realizedInt         AppendCompartmentText("#Element     } } </pre> <p>If an element <i>Class 1</i> has this Shape Script and is the source of a <b>Realization</b> connector to an element <i>Interface 1</i>, the name <i>Interface 1</i> is displayed in the <b>realizedInterfaces</b> compartment of <i>Class 1</i>.</p>	

### Notes

- If you use punctuation within a compartment name, it is stripped out when the script is saved; for example, *Ports, Parts and Properties* becomes *Ports Parts and Properties*
- Visibility of each individual custom compartment defined by a shape script can be controlled using the **Feature and Compartment Visibility** dialog
- The 'RelatedElement' Shape Scripts have extended capabilities to check both a connector **and** the element on the **other end** of the connector; they are applied only to an element and are solely used to retrieve information to be displayed within a compartment of that element

### Learn more

- [Add Shape Scripts](#) <sup>[150]</sup>
- [Feature Visibility](#) <sup>[845]</sup>
- [Show Composite Diagram](#) <sup>[1608]</sup>
- [Elements Tab](#) <sup>[828]</sup>

#### 7.3.5.3.9 Show Composite Diagram

You can define an element as being **Composite** (using the **New Diagram | Composite Structure Diagram** context menu option), which has a child **composite diagram** depicting the substructure of the element. You can also use context menu options to **display** the composite diagram **on the element**, either recasting the element as a frame or adding a compartment to the element. Ordinarily, a Shape Script that redefines the appearance of the composite element effectively circumvents the effect of these options, but you can edit the script to respond to the **Show Composite Diagram in Compartment** option and show the child composite diagram in the **center compartment** of the element.

To show composite diagrams, the script requires a layout type of **border**, with the composite diagram added to the **center** sub-shape of the **main** shape when drawing. The defining Shape Script statements are, therefore:

```

shape main
{
    layout type="Border";

    if ( HasProperty("ShowComposedDiagram", "true"))
    {
        addsubshape("ComposedDiagram", "CENTER");
    }
}

```



```

    }

    shape ComposedDiagram
    {
        DrawComposedDiagram();
    }
}

```

### Examples

An example of a Shape Script including a composed diagram is shown below:

```

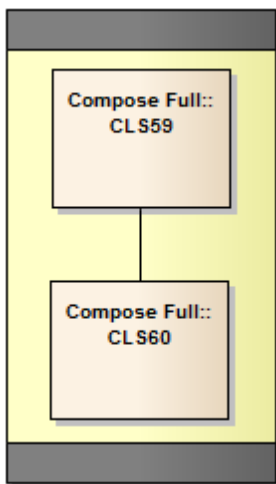
Shape main
{
    // Set the border type
    layouttype="Border";
    // Set a cream fill colour
    setfillcolor(255,255,200);
    // Draw a base rectangle for the object.
    rectangle(0,0,100,100);
    // Add some padding to the top of the shape
    addsubshape("Padding", "N");
    // Check the setting of the context menu option
    if (HasProperty("ShowComposedDiagram", "true"))
    {
        // Add the composed diagram to the center of the object
        addsubshape("ComposedDiagram", "CENTER");
    }
    // Add some padding to the bottom of the shape.
    addsubshape("Padding", "S");

    shape Padding
    {
        // Set the height of this element
        preferredHeight = 20;
        // Set the fill colour to grey
        setfillcolor(128,128,128);
        // Draw a rectangle that will take up the width of the object
        and have a height of 20 pixels.
        rectangle(0,0,100,100);
    }

    shape ComposedDiagram
    {
        // Draw the composed diagram.
        DrawComposedDiagram();
    }
}

```

This script generates the following shape.



Composed diagrams are currently only supported as the **center** sub-shape of the main shape. Adding the diagram to any other location will cause the composed diagram to either not draw correctly or not draw at all. The diagram **can** be a sub-shape of a sub-shape, but only if the parent shape and sub-shape(s) all have a "CENTER" orientation. For example:

//This shapscript is fine, because shape E is the center of shape C, which is the center of shape D, i.e. all shapes leading to DrawComposedDiagram are //"CENTER".

```

shape main
{
    layouttype = "Border";
    rectangle(0,0,100,100);
    addsubshape("D", "CENTER");

    shape D
    {
        layouttype="Border";
        addsubshape("C", "CENTER");
        shape C
        {
            layouttype="Border";
            addsubshape("E", "CENTER");
            addsubshape("Padding", "N");
            addsubshape("Padding", "S");

            shape E
            {
                DrawComposedDiagram();
            }

            shape padding
            {
                preferredHeight = 20;
                setfillcolor(10,30,80);
                rectangle(0,0,100,100);
            }
        }
    }
}

```

//This shapscript is not good - shape E is "CENTER", shape C is "S" and shape D is "CENTER"; because shape C is oriented "S"  
 //the diagram won't draw

```

shape main
{
  layouttype = "Border";
  rectangle(0,0,100,100);
  addsubshape("D", "CENTER");

  shape D
  {
    layouttype="Border";
    addsubshape("C", "S"); //<- this is bad, all parent subshape of a DrawComposedDiagram call
    MUST be                // "CENTER" oriented

    shape C
    {
      layouttype="Border";
      addsubshape("E", "CENTER");
      addsubshape("Padding", "N");
      addsubshape("Padding", "S");

      shape E
      {
        DrawComposedDiagram();
      }


      shape padding
      {
        preferredHeight = 20;
        setfillcolor(10,30,80);
        rectangle(0,0,100,100);
      }
    }
  }
}

```

### Notes

- To display the Composite diagram, the **New Diagram | Show Composite Diagram in Compartment** option should be selected on the element's context menu in the diagram
- The composed diagram is displayed at natural size, so the parent element can not be resized to be smaller than the composed diagram

### Learn more

- [Write Scripts](#) 
- [Composite Elements](#) 

#### 7.3.5.3.10 Reserved Names

When you write a Shape Script, there are certain terms that are reserved because they have special meaning in the script; use them for their specific purposes.

### Elements

Elements (such as Class, State or Event) have the following reserved names for parts of the shape.

Name	Description	See Also
<b>shape main</b>	The <b>main</b> shape is the whole shape.	
<b>shape label</b>	The shape <b>label</b> gives the shape a detached label.	
<b>decoration</b> <identifier>	<b>Decoration</b> gives the shape a decoration as defined by the name in <identifier>.	
<b>shape ChildElement</b>	Allows addition of custom compartments based on child elements belonging to the current element.	<a href="#">Add Custom Compartments to Element</a> <sup>[1604]</sup>
<b>shape RelatedElement</b>	Allows addition of custom compartments based on related elements belonging to the current element.	<a href="#">Add Custom Compartments to Element</a> <sup>[1604]</sup>

### Connectors

Connectors (such as Association, Dependency or Generalization) have the following reserved names for parts of the shape.

Name	Description
<b>shape main</b>	The <b>main</b> shape is the whole shape.
<b>shape source</b>	The <b>source</b> shape is an extra shape at the source end of the connector.
<b>shape target</b>	The <b>target</b> shape is an extra shape at the target end of the connector.
<b>shape LeftTopLabel</b>	Each of these shapes defines a detached label for the connector in the position matching the label name.
<b>shape MiddleTopLabel</b>	
<b>shape RightTopLabel</b>	
<b>shape LeftBottomLabel</b>	
<b>shape</b>	

Name	Description
<b>MiddleBottomLabel</b>	
<b>shape RightBottomLabel</b>	

### 7.3.5.3.11 Syntax Grammar

A section of a Shape Script can be quite complex, containing a number of commands and parameters. The table below provides a breakdown of the Shape Script structure, illustrating how commands and parameters are constructed. The first entry is the top-level declaration, and subsequent entries show the composition of successively more detailed components.

#### Grammar symbols

- \* = zero or more
- + = one or more
- | = or
- ; = terminator

<b>ShapeScript</b>	::=	<Shape>*;
<b>Shape</b>	::=	<ShapeDeclaration> <ShapeBody>;
<b>ShapeDeclaration</b>	::=	<ShapeType> <ShapeName>;
<b>ShapeType</b>	::=	"shape"   "decoration"   "label";
<b>ShapeName</b>	::=	<ReservedShapeName>   <stringliteral>;
<b>ReservedShapeName</b>	::=	See <a href="#">Reserved Names</a> <sup>[1611]</sup> for full reserved shape listing.
<b>ShapeBody</b>	::=	"{" <InitialisationAttributeAssignment>* <DrawingStatement>* <SubShape>* "}";
<b>InitialisationAttributeAssignment</b>	::=	<Attribute> "=" <Value> ",";
<b>Attribute</b>	::=	See <a href="#">Shape Attributes</a> <sup>[1588]</sup> for full listing of attribute names.

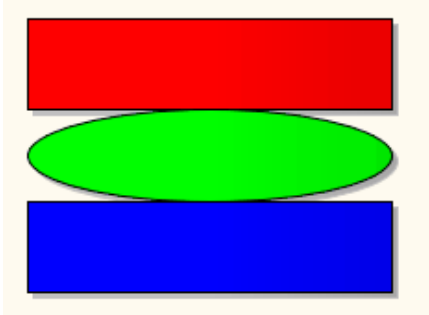
<b>DrawingStatement</b>	::=	<IfElseSection>   <Method>;
<b>IfElseSection</b>	::=	"if" "(" <QueryExpression> ")" <TrueSection> ( <ElseSection> ) ;
<b>QueryExpression</b>	::=	<QueryName> "(" <ParameterList> ")"; See <a href="#">Query Methods</a> <sup>[1599]</sup> for descriptions of the queries and their parameters.
<b>QueryName</b>	::=	See <a href="#">Query Methods</a> <sup>[1599]</sup> for the possible Query names.
<b>TrueSection</b>	::=	"{" <DrawingStatement>* "}"
<b>ElseSection</b>	::=	"else" "{" <DrawingStatement>* "}"
<b>Method</b>	::=	<MethodName> "(" <ParameterList> ")" ";"
<b>MethodName</b>	::=	See <a href="#">Drawing Methods</a> <sup>[1591]</sup> for a full listing of method names.

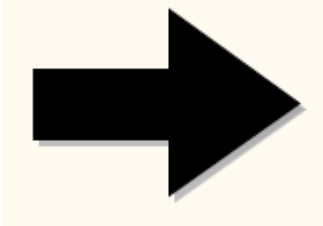
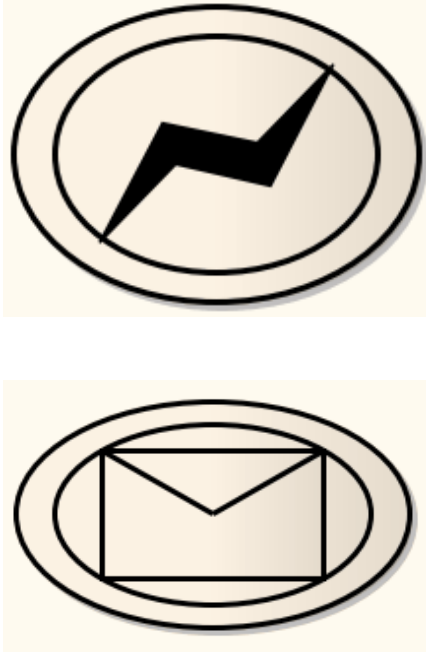
### 7.3.5.4 Example Scripts

You can create a wide range of shapes, effects and text statements using Shape Scripts, to enhance the appearance and information value of the elements and connectors you create. Some examples of such scripts are provided here.

**Access** **Settings | UML Types > Stereotypes (specify stereotype): Shape Script + Assign** or  
**Settings | UML Types > Stereotypes (specify stereotype): Shape Script + Edit**

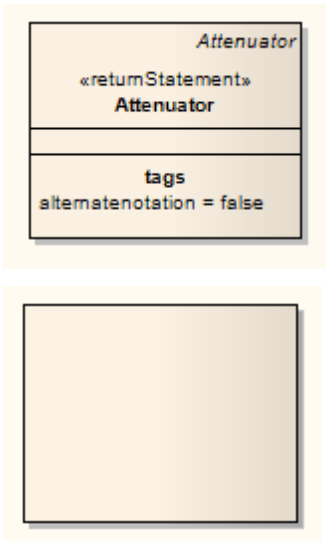
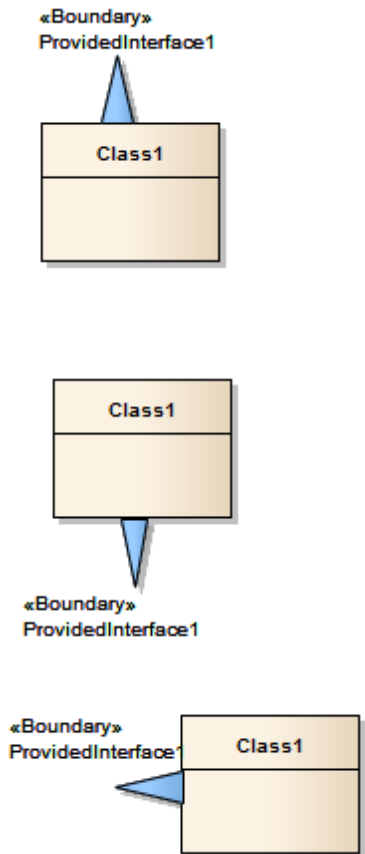
#### Shape Script examples

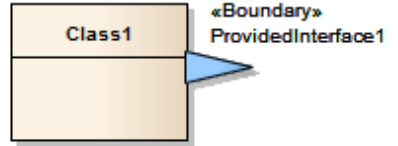
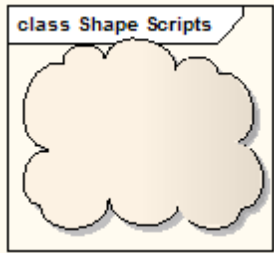
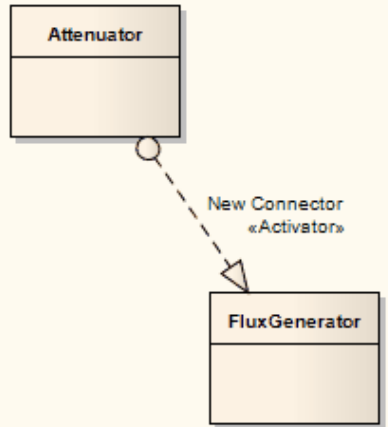
Script	Shape
<pre>// BASIC SHAPES shape main {     setfillcolor(255,0,0); // (R,G,B)     rectangle(0,0,90,30); // (x1,y1,x2,y2)      setfillcolor(0,255,0); // (R,G,B)     ellipse(0,30,90,60); // (x1,y1,x2,y2)      setfillcolor(0,0,255); // (R,G,B)     rectangle(0,60,90,90); // (x1,y1,x2,y2) }</pre>	

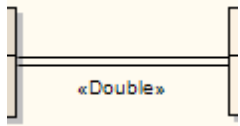
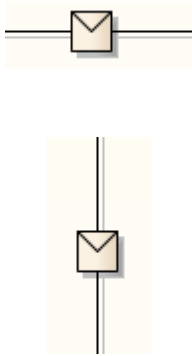
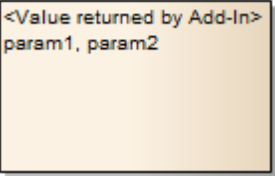
Script	Shape
<pre> // SINGLE CONDITIONAL SHAPE shape main {     if (HasTag("Trigger","Link"))     {<i>// Only draw if the object has a Tagged Value</i>         <i>// Trigger=Link</i>         <i>// Set the fill color for the path</i>         setfillcolor(0,0,0);         startpath(); <i>// Start to trace out a path</i>         moveto(23,40);         lineto(23,60);         lineto(50,60);         lineto(50,76);         lineto(76,50);         lineto(50,23);         lineto(50,40);         endpath(); <i>// End tracing out a path</i>         <i>// Fill the traced path with the fill color</i>         fillandstrokepath();         return;     } } </pre>	
<pre> // MULTI CONDITIONAL SHAPE shape main {     startpath();     ellipse(0,0,100,100);     endpath();     fillandstrokepath();     ellipse(3,3,97,97);      if (HasTag("Trigger","None"))     {         return;     }      if (HasTag("Trigger","Error"))     {         setfillcolor(0,0,0);         startpath();         moveto(23,77);         lineto(37,40);         lineto(60,47);         lineto(77,23);         lineto(63,60);         lineto(40,53);         lineto(23,77);         endpath();         fillandstrokepath();         return;     }      if (HasTag("Trigger","Message"))     {         rectangle(22,22,78,78);         moveto(22,22);         lineto(50,50);         lineto(78,22);         return;     } } </pre>	

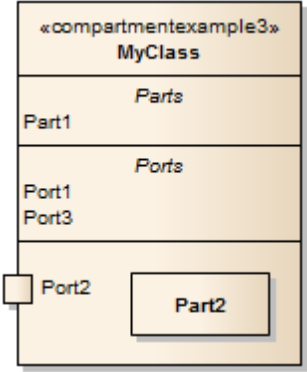
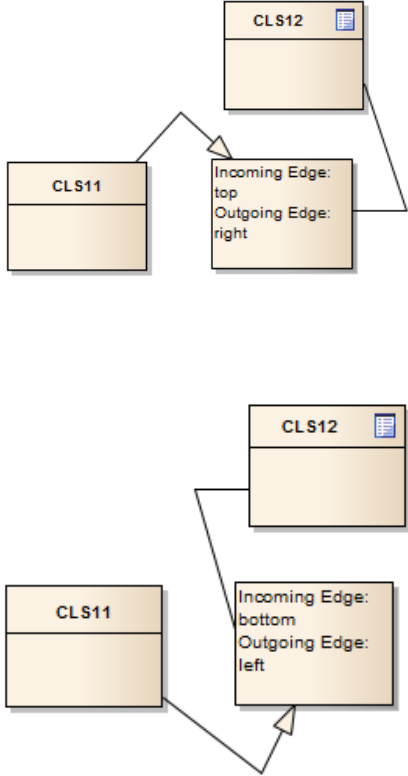
Script	Shape
<pre> }  // SUB SHAPES shape main {     rectangle(0,0,100,100);      addsubshape("red", 10, 20);     addsubshape("blue", 30, 40);     addsubshape("green", 50, 20);     addsubshape("red", 100, 20);      shape red     {         setfillcolor(200, 50, 100);         rectangle(0,0,100,100);     }      shape blue     {         setfillcolor(100, 50, 200);         rectangle(0,0,100,100);     }      shape green     {         setfillcolor(50, 200, 100);         rectangle(0,0,100,100);     } } </pre>	
<pre> // EDITABLE FIELD SHAPE shape main {     rectangle(0,0,100,100);      addsubshape("namecompartment", 100, 20);     addsubshape("stereotypecompartment", 100, 40);      shape namecompartment     {         h_align = "center";         editablefield = "name";          rectangle(0,0,100,100);         println("name: #name#");     }      shape stereotypecompartment     {         h_align = "center";         editablefield = "stereotype";          rectangle(0,0,100,100);         println("stereotype: #stereotype#");     } } </pre>	



Script	Shape
<pre>// RETURN STATEMENT SHAPE shape main {     if(hasTag("alternatenotation", "false"))     {         //draw ea's inbuild glyph         drawnativeshape();         //exit script with the return statement         return;     }     else     {         //alternate notation commands         //...         rectangle(0,0,100,100);     } }</pre>	
<pre>//EMBEDDED ELEMENT SHAPE POSITION ON PARENT EDGE shape main {     defsize(60,60);     startpath();      if(hasproperty("parentedge", "top"))     {         moveto(0,100);         lineto(50,0);         lineto(100,100);     }      if(hasproperty("parentedge", "bottom"))     {         moveto(0,0);         lineto(50,100);         lineto(100,0);     }      if(hasproperty("parentedge", "left"))     {         moveto(100,0);         lineto(0,50);         lineto(100,100);     }      if(hasproperty("parentedge", "right"))     {         moveto(0,0);         lineto(100,50);         lineto(0,100);     }      endpath();     setfillcolor(153,204,255);     filllandstrokepath(); }</pre>	

Script	Shape
	 <p>«Boundary» ProvidedInterface1</p>
<pre>// CLOUD PATH EXAMPLE SHAPE shape main {     StartCloudPath();     Rectangle(0,0,100,100);     EndPath();     FillAndStrokePath(); }</pre>	 <p>class Shape Scripts</p>
<pre>// CONNECTOR SHAPE shape main {     // draw a dashed line     noshadow=true;     setlinestyle("DASH");     moveto(0,0);     lineto(100,0); }  shape source {     // draw a circle at the source end     rotatable = true;     startpath();     ellipse(0,6,12,-6);     endpath();     fillandstrokepath(); }  shape target {     // draw an arrowhead at the target end     rotatable = true;     startpath();     moveto(0,0);     lineto(16,6);     lineto(16,-6);     endpath();     fillandstrokepath(); }</pre>	 <p>Attenuator</p> <p>New Connector «Activator»</p> <p>FluxGenerator</p>
<pre>// DOUBLE LINE shape main {     setlinestyle("DOUBLE"); }</pre>	

Script	Shape
<pre> moveto(0,0); lineto(100,0); } </pre>	
<pre> // ROTATION DIRECTION shape main {   moveto(0,0);   lineto(100,0);   setfixedregion(40,-10,60,10);   rectangle(40,-10,60,10);   if(hasproperty("rotationdirection","up"))   {     moveto(60,-10);     lineto(50,0);     lineto(60,10);   }   if(hasproperty("rotationdirection","down"))   {     moveto(40,-10);     lineto(50,0);     lineto(40,10);   }   if(hasproperty("rotationdirection","left"))   {     moveto(40,-10);     lineto(50,0);     lineto(60,-10);   }   if(hasproperty("rotationdirection","right"))   {     moveto(40,10);     lineto(50,0);     lineto(60,10);   } } </pre>	
<pre> // GET A VALUE RETURNED BY AN ADD-IN shape main {   //Draw a simple rectangle   Rectangle(0,0,100,100);    //Print string value returned from Add-In "My   //Function "MyExample" with two string parame   Print("#ADDIN:MyAddin, MyExample, param1, par  }  // METHOD SIGNATURE FOR ADD-IN FUNCTION: // Public Function MyExample(Repository As EA.Reposi // eaGuid As String, args As Variant) As Variant </pre>	

Script	Shape
<pre>// ADD CUSTOM COMPARTMENTS BASED UPON CHILD ELEMENTS // OR RELATED ELEMENTS  ( See <a href="#">Add Custom Compartments to Element</a><sup>[1604]</sup> )</pre>	 <p>The diagram shows a class named «compartmentexample3» MyClass. It has four compartments: 'Parts' containing 'Part1', 'Ports' containing 'Port1' and 'Port3', and a bottom compartment containing 'Port2' and 'Part2'. 'Port2' is connected to 'Part2' by a line.</p>
<pre>// RETURN THE INCOMING AND OUTGOING EDGE FOR CONNECT // GOING INTO AND OUT OF AN OBJECT  shape main {     //Draw a simple rectangle     Rectangle(0,0,100,100);      //Print incoming edges on the element     Print("Incoming Edge: #incomingedge#\n");      //Print outgoing edges on the element     Print("Outgoing Edge: #outgoingedge#\n"); }</pre>	 <p>The top diagram shows CLS11 connected to CLS12 via an edge. The edge is labeled 'Incoming Edge: top' and 'Outgoing Edge: right'. The bottom diagram shows CLS11 connected to CLS12 via an edge. The edge is labeled 'Incoming Edge: bottom' and 'Outgoing Edge: left'.</p>
<pre>// CHECK WHETHER A COMPOSITE ELEMENT ICON IS REQUIRE // IF SO, DRAW ONE  decoration comp {     orientation="SE";     if(hasproperty("IsDrawCompositeLinkIcon", "true")     {         startpath();         ellipse(-80,29,-10,71);         ellipse(10,29,80,71);         moveto(-10,50);         lineto(10,50);         endpath();     } }</pre>	

Script	Shape
<pre> strokepath( );     } } </pre>	

**Learn more**

- [Display Element/Connector Properties](#) <sup>[1599]</sup>
- [Drawing Methods](#) <sup>[1591]</sup>
- [Show Composite Diagram](#) <sup>[1608]</sup>

**Learning Center topics**

- (Alt+F1) | **Enterprise Architect | Modeling Languages | Defining Shapes | Defining an Element Shape**
- (Alt+F1) | **Enterprise Architect | Modeling Languages | Defining Shapes | Defining a Connector Shape**

**7.3.6 Tagged Value Types**

When you are working with Tagged Values, you can create your own, custom, Tagged Values based on **predefined**, system-provided Tagged Value **Types**. With these, you can create:

- Tagged Values that are **complex** and based on **predefined types**, with or without tag filters
- **Structured Tagged Values** that are composite, containing other Tagged Values
- Tagged Values that return values from the various **reference data** tables
- **Masked** Tagged Values that insert user-provided data into a text string such as line of prompts or field names

By adding Tagged Values of any type to a Stereotype element in a Profile, you can define additional meta-information for the way in which a modeling element appears and behaves in a Technology. The Tagged Values are identified by **attributes** of the Stereotype element.

**Notes**

- You can transport Tagged Value Type definitions **between models**, using the **Export Reference Data** and **Import Reference Data** options; Tagged Value Types are exported as **Property Types**

**Learn more**

- [Create Tagged Value from Predefined Types](#) <sup>[1622]</sup>
- [Predefined Types](#) <sup>[1622]</sup>
- [Create Custom Masked Tagged Value Type](#) <sup>[1626]</sup>
- [Create Reference Data Tagged Values](#) <sup>[1628]</sup>
- [Predefined Reference Data Types](#) <sup>[1629]</sup>
- [Add an Enumeration to a Stereotype](#) <sup>[1493]</sup>
- [Add a Structured Tagged Value to a Stereotype](#) <sup>[1495]</sup>
- [Export Reference Data](#) <sup>[376]</sup>

- [Import Reference Data](#)<sup>[380]</sup>

### 7.3.6.1 Create Tagged Value Type from Predefined Types

When you are working with Tagged Values, you might want to use **structured** Tagged Values; that is, Tagged Values that capture and present more complex information in a specific format. The base types for such Tagged Values (the type you call in when you create a tag in the Tagged Values window) can be easily created specifically for your model, as you can base the customized structured Tagged Value types on a range of **predefined** structured Tagged Value types and **filters**.

**Access** [Settings | UML Types > Tagged Value Types](#)

#### Create a custom Structured Tagged Value type

Field/ Button	Description	See also
<b>Tag Name</b>	Type an appropriate name for your new Tagged Value type.	
<b>Description</b>	Optionally, type a short description or the purpose of the Tagged Value type.	
<b>Detail</b>	Either copy-and-paste or type the syntax of the predefined structured Tagged Value Type on which to base your new Tagged Value type.	<a href="#">Predefined Structured Types</a> <sup>[1622]</sup>
<b>Save</b>	Click on this button to save the new structured Tagged Value type. The Tagged Value type displays in the Defined Tag Types list.	
<b>New</b>	Optionally, click on this button to clear the fields so that you can enter information for another new Tagged Value type.	

### 7.3.6.2 Predefined Structured Types

Tagged Values define a wide range of properties and characteristics of a model element, and some of these properties have complex values. For example, you might want your user to select a value between upper and lower limits (using 'Spin' arrows), set a date, select a color from a palette, or work through a checklist.

You create these complex Tagged Values from any of a number of **predefined** Tagged Value **types** and **filters**, some of which you might have created yourself ([Settings | UML Types > Tagged Value Types](#)).

#### Tagged Value Type Formats

For each Tagged Value Type, the description includes the syntax for creating the initial values for use of the Tagged Value. The name and format are case-sensitive.

Tagged Value Type	Format	Used to	See also
AddinBroadcast	<b>Type=AddinBroadcast;</b> <b>Values=YourAddinName;</b>	Allow an Add-In to respond to an attempt to edit this Tagged Value by showing a dialog in which the value and notes can be edited.	
Boolean	<b>Type=Boolean;</b> <b>Default=Val;</b>	Provide for the input of <b>True</b> or <b>False</b> , either of which can be the default value.	
CheckList	<b>Type=CheckList;</b> <b>Values=Val1,Val2,Val3;</b>	<p>Create a checklist of things to be completed or satisfied before an action is approved or performed.</p> <p><b>Val1, Val2, Val3</b> and so on specify the checklist items, each of which is rendered via the Tagged Values window with a checkbox; the tag has the value <b>Incomplete</b> until each checkbox is selected, at which point the value is <b>Complete</b>.</p> <p>For example:</p> <p><b>Type=CheckList;</b> <b>Values=Does the change solve the task,issue given,Does the code have sufficient error handling,Does the code make sense,Does the code comply with the coding conventions;</b></p> <p>Whilst the element Tagged Value compartment and the Tagged Value window fields display the values <b>Complete</b> or <b>Incomplete</b>, document and web reports will show the list of checklist items and the status of each (<b>True</b> for selected, <b>False</b> for unselected).</p>	
Classifier	<b>Type=Classifier;</b> <b>Values=Type1,Type2;</b> <b>Stereotypes=Stereotype1;</b>	<b>Deprecated - use RefGUID and RefGUIDList</b>	
Color	<b>Type=Color;</b> <b>Default=Val;</b>	<p>Input a color value from a color chooser menu, where the value is the color's Hex RGB value.</p> <p>For example, the Hex RGB for Blue is <b>0000FF</b>, whilst the Hex RGB for Green is <b>00FF00</b>.</p>	
Const	<b>Type=Const;</b> <b>Default=Val;</b>	Create a read-only constant value.	

Tagged Value Type	Format	Used to	See also
Custom	<b>Type=Custom;</b>	Create your own template for predefined types, using a masked value.	<a href="#">Create Custom Masked Tagged Value Type</a> <sup>[1626]</sup>
Date	<b>Type=Date;</b>	Input the date for the Tagged Value, from a calendar menu.	
DateTime	<b>Type=DateTime;</b>	<b>Deprecated - Use Date</b> Input the date for the Tagged Value, from a calendar menu.	
Directory	<b>Type=Directory;</b> <b>Default=Val;</b>	Enter a directory path from a browser. You can set a default directory path as a string value.	
Enum	<b>Type=Enum;</b> <b>Values=Val1,Val2,Val3;</b> <b>Default=Val2;</b>	Define a comma-separated list, where <b>Val1</b> , <b>Val2</b> and <b>Val3</b> represent values in the list and <b>Default</b> represents the default value of the list.	<a href="#">Add an Enumeration to a Stereotype</a> <sup>[1493]</sup>
File	<b>Type=File;</b> <b>Default=Val;</b>	Input a filename from a file browser dialog. The named file can be launched in its default application.  You can set a default file as a string containing the file path and file name.	
Float, Decimal, Double	<b>Type=Float;</b> <b>Type=Decimal;</b> <b>Type=Double;</b> <b>Default=Val;</b>	Enter a Float, Decimal or Double value. These types all map to the same type of data.  You can set a default for any or all of these.	
Integer	<b>Type=Integer;</b> <b>Default=Val;</b>	Enter an Integer value, and a default.	
Memo	<b>Type=Memo;</b>	Input large and complex values for a tag.	
RefGUID	<b>Type=RefGUID;</b> <b>Values=Type1,Type2;</b> <b>Stereotypes=Stereotype1;</b>	Reference an element from the model by specifying the element's <i>GUID</i> , where: <ul style="list-style-type: none"> <li><b>Type1</b> and <b>Type2</b> specify one or more allowed diagram objects</li> </ul>	<a href="#">Classifier</a> <sup>[994]</sup> , <a href="#">Attribute or Operation</a> <sup>[998]</sup>



Tagged Value Type	Format	Used to	See also
		<p>(such as <b>Class</b>, <b>Component</b>, <b>Attribute</b> or <b>Operation</b>)</p> <ul style="list-style-type: none"> <li><b>Stereotype1</b> represents an allowed stereotype</li> </ul> <p>You set the classifier, attribute or operation for a Tagged Value of this type by clicking on the ( ... ) button against the Tagged Value in the Tagged Value window.</p>	
RefGUIDList	<b>Type=RefGUIDList;</b> <b>Values=Type1,Type2;</b> <b>Stereotypes=Stereotype1;</b>	<p>Reference a list of elements from the model by specifying each element's <i>GUID</i>, where:</p> <ul style="list-style-type: none"> <li><b>Type1</b> and <b>Type2</b> specify one or more allowed diagram objects (such as <b>Class</b> or <b>Component</b>)</li> <li><b>Stereotype1</b> represents an allowed stereotype</li> </ul> <p>You set the classifier, attribute or operation for a Tagged Value of this type by clicking on the ( ... ) button against the Tagged Value in the Tagged Value window.</p>	<a href="#">Classifier</a> <sup>[994]</sup> , <a href="#">Attribute or Operation</a> <sup>[998]</sup>
Spin	<b>Type=Spin;</b> <b>LowerBound=x;</b> <b>UpperBound=x;</b> <b>Default=Val;</b>	<p>Create a spin control with the value of <b>LowerBound</b> being the lowest value and <b>UpperBound</b> being the highest value.</p> <p>You can also set a default within that range.</p>	
String	<b>Type=String;</b> <b>Default=Val;</b>	<p>Enter a string value, up to 255 characters in length, and a default text string.</p> <p>For longer texts, use <b>Type=Memo</b>.</p>	<a href="#">Memo</a> <sup>[1624]</sup>
Time	<b>Type=Time;</b>	Input the time for the Tagged Value.	
TimeStamp	<b>Type=TimeStamp;</b>	Input the date and time for the Tagged Value, from a calendar menu.	
URL	<b>Type=URL;</b> <b>Default=Val;</b>	<p>Enter a web URL. The URL should start with:</p> <ul style="list-style-type: none"> <li>'http:/'</li> <li>'https:/' or</li> </ul>	

Tagged Value Type	Format	Used to	See also
		<ul style="list-style-type: none"> <li>'www.'</li> </ul> <p>You can set a default URL as a string value.</p>	

### Tag Filters

You can use filters to restrict where a Tagged Value can be applied.

Filter	Format	Description	See also
AppliesTo	<b>AppliesTo=Type1, Type2;</b>	<p>Restricts the <b>element types</b> this tag can be applied to, where <b>Type1</b> and <b>Type2</b> are the valid types.</p> <p>Possible values are:</p> <ul style="list-style-type: none"> <li>All element types</li> <li>All connector types</li> <li>Attribute</li> <li>Operation, and</li> <li>OperationParameter</li> </ul>	
BaseStereotype	<b>BaseStereotype=S1, S2;</b>	<p>Restricts the <b>stereotypes</b> that this tag belongs to, where <b>S1</b> and <b>S2</b> are the allowed stereotypes.</p>	

### Learn more

- [Create Tagged Value Type from Predefined Types](#) 

### 7.3.6.3 Create Custom Masked Tagged Value Type

If you are creating a custom predefined Tagged Value type, you can achieve great flexibility in designing model components to **accept data entries**, by defining a **mask** that formats the data into a **template**.

**Access** [Settings | UML Types > Tagged Value Types](#)

### Create a masked Tagged Value Type

Field	Action	See also
<b>Tag Name</b>	Type an appropriate name for the masked Tagged Value Type.	

Field	Action	See also
<b>Description</b>	Optionally, type a description or the purpose of the Tagged Value Type.	
<b>Detail</b>	<p>Type or copy-and-paste the Tagged Value structure:</p> <p><b>Type=Custom;</b>  <b>Mask=&lt;mask values&gt; ;</b>  <b>Template=&lt;template text&gt;;</b></p> <p>The mask values are explained in the next table, with an example to demonstrate how to use the template.</p> <p>The template text defines information to be displayed in every use of this custom Tagged Value, such as field names and prompts for data.</p>	
<b>Save</b>	<p>Click on this button to save the new masked Tagged Value type.</p> <p>The Tagged Value type displays in the Defined Tag Types list.</p>	
<b>New</b>	Optionally, click on this button to clear the fields so that you can enter information for another new Tagged Value type.	

### Mask Values

When defining the format of the mask in a masked Tagged Value type, use these characters:

Mask	Action
<b>D</b>	Display a digit only in this character space.
<b>d</b>	Display a digit or space only in this character space.
<b>+</b>	Display <b>+</b> , <b>-</b> or a space in this character space.
<b>C</b>	Display a letter of the alphabet only in this character space.
<b>c</b>	Display a letter of the alphabet or a space only in this character space.
<b>A</b>	Display any alphanumeric character in this character space.
<b>a</b>	Display any alphanumeric character or a space in this character space.
<b>.</b> or	Leave a character space, to be filled by text from the Template parameter. Using dots might

Mask	Action
<b>&lt;space&gt;</b>	make it easier to see how many spaces you have set.

### Example

The screenshot shows a software interface for defining Tagged Value Types. It has three tabs: 'Stereotypes', 'Tagged Value Types' (which is active), and 'Cardinality Values'. In the 'Tagged Value Types' tab, there are two input fields: 'Tag Name' with the value 'MemberZip' and 'Description' with the value 'Zip Code'. Below these is a 'Detail' section containing a text area with the following text: 'Type=Custom; Mask= cc dddd.dddd; Template=State: \_\_ Zip: \_\_\_\_ - \_\_\_\_;'. At the bottom right of the dialog are three buttons: 'New', 'Save', and 'Delete'.

In the diagram, the **Mask** parameter first defines seven blank spaces, which are occupied by characters defined by the **Template** parameter.

The first two visible characters in the **Mask** are each represented by a lower case **c**, indicating that the user can enter information as either an alphabetic character or a space.

The next six blank spaces again indicate characters defined by the **Template**, followed by five characters each represented by a **d**, which indicates that the user can input data in the form of digits or spaces. The dot marks a space to be filled by a hyphen from the **Template**, followed by four more **ds** (digits or spaces).

The **Template** syntax defines the template for the Mask parameter, filling in the blank spaces in the **Mask**. The text is the information to be printed with every use of this Tagged Value; the underscored values indicate the character spaces that are to be occupied by data input by the user, as defined in the **Mask** option.

### 7.3.6.4 Create Reference Data Tagged Values

When working with Tagged Values, you might want to use a Reference Data Tagged Value, which is used to return the values held in an Enterprise Architect reference table. The base types for such Tagged Values (the type you call in when you create a tag in the Tagged Values window) can be easily created specifically for your model, as you can base the customized Reference Data Tagged Value types on a range of **predefined** Tagged Value types and **filters**.

**Access** [Settings | UML Types > Tagged Value Types](#)

Create a custom Reference Data Tagged Value type

Field/ Button	Description	See also
<b>Tag Name</b>	Type an appropriate name for the new Tagged Value type.	
<b>Description</b>	Optionally, type the a description or the purpose of the Tagged Value type.	
<b>Detail</b>	Either copy-and-paste or type the syntax of the predefined Reference Data Tagged Value type on which to base your new Tagged Value type.	<a href="#">Predefined Reference Data Types</a> <sup>[1629]</sup>
<b>Save</b>	Click on this button to save the new Reference Data Tagged Value type. The Tagged Value type displays in the Defined Tag Types list.	
<b>New</b>	Optionally, click on this button to clear the fields so that you can enter information for another new Tagged Value type.	

#### Notes

- If the values in the reference data are changed after the Tagged Value Type is created, you must reload the system in order to reflect the changes in the Tagged Value Type

### 7.3.6.5 Predefined Reference Data Types

If you want to create your own, customized, Reference Data Tagged Values, you can base them on a range of **predefined** Reference Data Tagged Value **types**. Each of the predefined Reference Data Tagged Value types returns the values held in a specific reference data table.

#### Tagged Value Types

Each description includes the syntax for creating the initial values for use of the Tagged Value. The **Tagged Value Type** and **Format** entries are case-sensitive.

Tagged Value Type	Format	Drop-Down List Returned, of Data Defined for the Model	See also
<b>Authors</b>	<b>Type=Enum; List=Authors;</b>	Authors.	
<b>Cardinality</b>	<b>Type=Enum; List=Cardinality;</b>	Cardinality types.	
<b>Clients</b>	<b>Type=Enum; List=Clients;</b>	Clients.	
<b>ComplexityTypes</b>	<b>Type=Enum;</b>	Complexity types.	

Tagged Value Type	Format	Drop-Down List Returned, of Data Defined for the Model	See also
	<b>List=ComplexityTypes;</b>	Whilst complexity types can be exported and imported as project reference data, they cannot be updated and so are effectively standard across all projects.	
<b>ConstraintTypes</b>	<b>Type=Enum; List=ConstraintTypes;</b>	Constraint types.	
<b>EffortTypes</b>	<b>Type=Enum; List=EffortTypes;</b>	Effort types.	
<b>MaintenanceTypes</b>	<b>Type=Enum; List=MaintenanceTypes;</b>	Maintenance types.	
<b>ObjectTypes</b>	<b>Type=Enum; List=ObjectTypes;</b>	Object types.	
<b>Phases</b>	<b>Type=Enum; List=Phases;</b>	Phases.	
<b>ProblemTypes</b>	<b>Type=Enum; List=ProblemTypes;</b>	Problem types.	
<b>RoleTypes</b>	<b>Type=Enum; List=RoleTypes;</b>	Role types.	
<b>RequirementTypes</b>	<b>Type=Enum; List=RequirementTypes;</b>	Requirement types.	
<b>Resources</b>	<b>Type=Enum; List=Resources;</b>	Resources.	
<b>RiskTypes</b>	<b>Type=Enum; List=RiskTypes;</b>	Risk types.	
<b>RTFTemplates</b>	<b>Type=Enum; List=RTFTemplates;</b>	Document Report Templates.	

Tagged Value Type	Format	Drop-Down List Returned, of Data Defined for the Model	See also
<b>ScenarioTypes</b>	<b>Type=Enum; List=ScenarioTypes;</b>	Scenario types.	
<b>TestTypes</b>	<b>Type=Enum; List=TestTypes;</b>	Test types.	

#### Learn more

- [Reference Data](#) <sup>[1146]</sup>
- [Create Reference Data Tagged Values](#) <sup>[1628]</sup>

### 7.3.7 Code Template Framework

When you use Enterprise Architect to generate code from a model, or transform the model, the system refers to the Code Template Framework (CTF) for the parameters that define how it should:

- Forward engineer a UML model
- Generate Behavioral Code
- Perform a Model Driven Architecture (MDA) Transformation

A range of standard templates is available for the direct generation of code and for transformation; if you do not want to use the standard CTF configurations, you can customize them to meet your needs.

#### CTF Templates

Template Type	Detail	See also
<b>Code Templates</b>	<p>When forward engineering a Class model, the code templates define how the skeletal code is to be generated for a given programming language. The templates for a language are automatically associated with the language.</p> <p>The templates are written as plain text with a syntax that shares some aspects of both mark-up languages and scripting languages.</p>	<a href="#">Code Template Framework Tool</a> <sup>[2281]</sup> <a href="#">Base Templates</a> <sup>[1633]</sup> <a href="#">Code Template Syntax</a> <sup>[1643]</sup> <a href="#">The Code Template Editor in MDG Development</a> <sup>[1701]</sup>
<b>Model Transformation Templates</b>	<p>Model Transformation Templates provide a fully configurable method of defining how Model Driven Architecture (MDA) Transformations convert model elements and model fragments from one domain to another.</p> <p>This process is two-tiered. It creates an intermediary language</p>	<a href="#">Edit Transformation Templates</a> <sup>[2048]</sup> <a href="#">Write Transformations</a> <sup>[2051]</sup> <a href="#">Intermediary</a>

	(which can be viewed for debugging) which is then processed to create the objects.	<a href="#">Language</a> <sup>[2053]</sup>
<b>Behavioral Code Generation Templates</b>	Enterprise Architect supports user-definable code generation of the UML Behavioral models.  This applies the standard Code Template Framework but includes specific Enterprise Architect Simulation Library (EASL) code generation macros.	<a href="#">The Code Template Editor in MDG Development</a> <sup>[1701]</sup>  <a href="#">EASL Code Generation Macros</a> <sup>[1688]</sup>

#### Learn more

- [Code and Transform Templates](#) <sup>[1632]</sup>
- [Code Template Editor](#) <sup>[1701]</sup>
- [Macros](#) <sup>[1646]</sup>

### 7.3.7.1 Code and Transform Templates

Code templates and transform (Model Transformation) templates define how the system should generate or transform code in one or other of the programming languages that Enterprise Architect supports. Each language has a wide range of base templates, each of which defines how a particular code structure is generated. You can use these base templates as they are, or you can customize and add to the templates to better support your use of the standard languages, or of other languages that you might define to the system. You review, update and create templates through the Code Template editor or Transformation Template editor.



The order in which the base templates are listed in the two editors relates to the hierarchical order of the objects and their parts that are to be processed. Calls are made from certain base templates to others, and you can add further calls to both base templates and to your own custom templates. By default, the *File* template is the starting point of a code generation process though the templates; a File consists of Classes which can contain Attributes and Operations.

**Access**    **Tools | Source Code Generation Templates**    (Ctrl+Shift+P)  
               **Tools | MDA Transformation Templates**    (Ctrl+Alt+H)

#### Application of Templates

Action	Detail	See also
<b>Calling Templates</b>	<p>Within any template, you can call other templates using %TemplateName%. The enclosing percent (%) signs indicate a macro.</p> <p>You would use this for a single call to the ClassBody template, %ClassBody% as below:</p> <pre>%list="TemplateName" @separator="\n" @indent="  "%</pre> <p>The %list macro performs an iterative pass on all the objects in the scope of the current template and calls the <i>TemplateName</i> for each of these:</p>	<a href="#">List Macro</a> <sup>[1683]</sup>



	<pre>%list="ClassBody" @separator="\n" @indent="  "%</pre> <p>After generation or transformation, each macro is substituted to produce the generated output; for a language such as C++, the result of processing the above template might be:</p> <pre>/**  * This is an example class note generated  * using code templates  * @author Sparx Systems  */ class ClassA: public ClassB {     ... }</pre>	
<b>Execution of Code Templates</b>	<p>Each template might act only on a particular element type; for example, the <i>ClassNotes</i> template only acts on UML Class and Interface elements.</p> <p>The element from which code is currently being generated is said to be <i>in scope</i>; if the element in scope is stereotyped, the system searches for a template that has been defined for that stereotype. If a specialized template is found, it is executed; otherwise the default implementation of the base template is used.</p> <p>Templates are processed sequentially, line by line, replacing each macro with its underlying text value from the model.</p>	
<b>Transfer Templates Between Projects</b>	<p>If you edit a base Code Generation or Transformation template, or create a customized template, you can copy them from one project to another as Reference Data.</p>	<p><a href="#">Export Code Generation and Transformation Templates</a> </p> <p><a href="#">Import Code Generation and Transformation Templates</a> </p>

#### Learn more

- [Generate Source Code](#) 
- [Base Templates](#) 
- [Model Transformation](#) 
- [Built-in Transformations](#) 

#### 7.3.7.1.1 Base Templates

The Code Template Framework consists of a number of base templates. Each base template transforms particular aspects of the UML to corresponding parts of object-oriented languages.

#### Topics

Topic	Detail	See also
<b>Further Information</b>	<p>The base templates form a hierarchy, which varies slightly across different programming languages.</p> <p>In a typical template hierarchy relevant to a language like C# or Java (which do not have header files) the templates can be modeled as Classes, but usually are just plain text.</p> <p>This hierarchy would be slightly more complicated for languages like C++ and Delphi, which have separate implementation templates.</p> <p>Each of the base templates must be specialized to be of use in code engineering; in particular, each template is specialized for the supported languages (or 'products').</p> <p>For example, there is a <i>ClassBody</i> template defined for C++, another for C#, another for Java, and so on; by specializing the templates, you can tailor the code generated for the corresponding UML entity.</p> <p>Once the base templates are specialized for a given language, they can be further specialized based on:</p> <ul style="list-style-type: none"> <li>• A Class's stereotype</li> <li>• A feature's stereotype (where the feature can be an operation or attribute)</li> </ul> <p>This type of specialization enables, for example, a C# operation that is stereotyped as «property» to have a different <i>Operation Body</i> template from an ordinary operation; the <i>Operation Body</i> template can then be specialized further, based on the Class stereotype.</p>	<a href="#">Code Template Framework</a> <a href="#">Code and Transformation Templates</a>

### Reference

The following table lists and briefly describes the base templates used in the CTF.

Template	Description
<b>Attribute</b>	A top-level template to generate member variables from UML attributes.
<b>Attribute Declaration</b>	Used by the <i>Attribute</i> template to generate a member variable declaration.
<b>Attribute Notes</b>	Used by the <i>Attribute</i> template to generate member variable notes.
<b>Class</b>	A top-level template for generating Classes from UML Classes.
<b>Class Base</b>	Used by the <i>Class</i> template to generate a base Class name in the inheritance list of a derived Class, where the base Class doesn't exist in the model.

Template	Description
<b>Class Body</b>	Used by the <i>Class</i> template to generate the body of a Class.
<b>Class Declaration</b>	Used by the <i>Class</i> template to generate the declaration of a Class.
<b>Class Interface</b>	Used by the <i>Class</i> template to generate an interface name in the inheritance list of a derived Class, where the interface doesn't exist in the model.
<b>Class Notes</b>	Used by the <i>Class</i> template to generate the Class notes.
<b>File</b>	A top-level template for generating the source file. For languages such as C++, this corresponds to the header file.
<b>Import Section</b>	Used in the <i>File</i> template to generate external dependencies.
<b>Linked Attribute</b>	A top-level template for generating attributes derived from UML Associations.
<b>Linked Attribute Notes</b>	Used by the <i>Linked Attribute</i> template to generate the attribute notes.
<b>Linked Attribute Declaration</b>	Used by the <i>Linked Attribute</i> template to generate the attribute declaration.
<b>Linked Class Base</b>	Used by the <i>Class</i> template to generate a base Class name in the inheritance list of a derived Class, for a Class element in the model that is a parent of the current Class.
<b>Linked Class Interface</b>	Used by the <i>Class</i> template to generate an Interface name in the inheritance list of a derived Class, for an Interface element in the model that is a parent of the current Class.
<b>Namespace</b>	A top-level template for generating namespaces from UML packages (although not all languages have namespaces, this template can be used to generate an equivalent construct, such as packages in Java).
<b>Namespace Body</b>	Used by the <i>Namespace</i> template to generate the body of a namespace.
<b>Namespace Declaration</b>	Used by the <i>Namespace</i> template to generate the namespace declaration.
<b>Operation</b>	A top-level template for generating operations from a UML Class's operations.

Template	Description
<b>Operation Body</b>	Used by the <i>Operation</i> template to generate the body of a UML operation.
<b>Operation Declaration</b>	Used by the <i>Operation</i> template to generate the operation declaration.
<b>Operation Notes</b>	Used by the <i>Operation</i> template to generate documentation for an operation.
<b>Parameter</b>	Used by the <i>Operation Declaration</i> template to generate parameters.

The second table lists templates used for generating code for languages that have separate interface and implementation sections.

Template	Description
<b>Class Impl</b>	A top-level template for generating the implementation of a Class.
<b>Class Body Impl</b>	Used by the <i>Class Impl</i> template to generate the implementation of Class members.
<b>File Impl</b>	A top-level template for generating the implementation file.
<b>File Notes Impl</b>	Used by the <i>File Impl</i> template to generate notes in the source file.
<b>Import Section Impl</b>	Used by the <i>File Impl</i> template to generate external dependencies.
<b>Operation Impl</b>	A top-level template for generating operations from a UML Class's operations.
<b>Operation Body Impl</b>	Used by the <i>Operation</i> template to generate the body of a UML operation.
<b>Operation Declaration Impl</b>	Used by the <i>Operation</i> template to generate the operation declaration.
<b>Operation Notes Impl</b>	Used by the <i>Operation</i> template to generate documentation for an operation.

### 7.3.7.1.2 Export Code Generation and Transformation Templates

**Access**    **Project | Model Import/Export | Export Reference Data**

#### Use to

- Transfer Code Generation and Transformation templates between models

#### How to

To export a Code Generation template or a Transformation template

Step	Action	See also
1	On the Export Reference Data dialog, in the <b>Name</b> list, select the templates to export.  The list includes any standard Code Generation or Transformation templates that have been <i>changed</i> , and any customized templates that you have created or changed.  You can select one or more templates to be exported to a single XML file, by pressing ( <b>Ctrl</b> ) or ( <b>Shift</b> ) as you click on the template names.	
2	Click on the <b>Export</b> button.	
3	When prompted to do so, enter a valid file name with a .XML extension.	
4	Click on the <b>Save</b> and <b>OK</b> buttons.  This exports the template(s) to the file; you can use any text or XML viewer to examine the file.	

#### Learn more

- [Write Transformations](#) <sup>[205]</sup>
- [Import Code Generation and Transformation Templates](#) <sup>[1637]</sup>
- [Export Reference Data](#) <sup>[376]</sup>

### 7.3.7.1.3 Import Code Generation and Transformation Templates

**Access**    **Project | Model Import/Export | Import Reference Data**

#### Use to

- Transfer Code Generation and Transformation templates between models

#### How to

To import a Code Generation template or a Transformation template

Step	Action	See also
1	On the Import Reference Data dialog, click on the <b>Select File</b> button and browse to the .XML file containing the required Code Generation or Transformation template.	
2	Select the name of one or more template datasets and click on the <b>Import</b> button.	

#### Learn more

- [Write Transformations](#) <sup>[205]</sup>
- [Import Reference Data](#) <sup>[380]</sup>
- [Export Code Generation and Transformation Templates](#) <sup>[1637]</sup>

#### 7.3.7.1.4 Synchronize Code

Enterprise Architect uses code templates during the forward synchronization of the following programming languages:

- ActionScript
- C
- C++
- C#
- Delphi
- Java
- PHP
- Python
- VB
- VB.Net

#### Topics

Topic	Detail	See also
<b>Change Types</b>	<p>Three types of change can occur in the source when it is synchronized with the UML model:</p> <ul style="list-style-type: none"> <li>• <b>Synchronize Existing Sections:</b> for example, changing the return type in an operation declaration</li> <li>• <b>Add New Sections to Existing Features:</b> for example, adding notes to a Class declaration, where there were previously none</li> <li>• <b>Add New Features and Elements:</b> for example, adding a new operation to a Class</li> </ul> <p>Each of these changes must be handled differently by Enterprise Architect; their effect on the CTF is described in the</p>	<p><a href="#">Synchronize Existing Sections</a> <sup>[1640]</sup></p> <p><a href="#">Add New Sections to Existing Features</a> <sup>[1640]</sup></p> <p><a href="#">Add New Features and Elements</a> <sup>[1641]</sup></p>

	topics listed on the right.	
--	-----------------------------	--

Only a *subset* of the CTF base templates is used during synchronization. This subset corresponds to the distinct sections that Enterprise Architect recognizes in the source code. The following table lists the code templates and the corresponding code sections that can be synchronized.

Code Template	Code Section
Class Notes	Comments preceding Class declaration.
Class Declaration	Up to and including Class parents.
Attribute Notes	Comments preceding Attribute declaration.
Attribute Declaration	Up to and including terminating character.
Operation Notes	Comments preceding operation declaration.
Operation Notes Impl	As for <i>Operation Notes</i> .
Operation Declaration	Up to and including terminating character.
Operation Declaration Impl	Up to and including terminating character.
Operation Body	Everything between and including the braces.
Operation Body Impl	As for <i>Operation Body</i> .

#### Learn more

- [Base Templates](#)  1633

#### 7.3.7.1.4.1 Synchronize Existing Sections

##### Topics

Topic	Detail	See also
<b>Usage</b>	<p>When an existing section in the source code differs from the result generated by the corresponding template, that section is replaced.</p> <p>Consider for example, the following C++ Class declaration:</p> <pre>( asm ) class A: public B</pre> <p>Now assume you add an inheritance relationship from Class A to Class C; the entire Class declaration would be replaced with something like:</p> <pre>( asm ) class A: public B, public C</pre>	

#### 7.3.7.1.4.2 Add New Sections

##### Topics

Topic	Detail	See also
<b>Usage</b>	<p>The following can be added as new sections, to existing features in the source code:</p> <ul style="list-style-type: none"> <li>• Class Notes</li> <li>• Attribute Notes</li> <li>• Operation Notes</li> <li>• Operation Notes Impl</li> <li>• Operation Body</li> <li>• Operation Body Impl</li> </ul> <p>Assume Class <b>A</b> from the previous example had no note when you originally generated the code, and that you now specify a note in the model for Class A. Enterprise Architect attempts to add the new note from the model during synchronization, by executing the <i>Class Notes</i> template.</p> <p>To make room for the new section to be inserted, you can specify how much white space to append to the section via synchronization macros.</p>	<p><a href="#">Synchronize Existing Sections</a> <small>1640</small></p> <p><a href="#">Synchronization Macros</a> <small>1686</small></p>



### 7.3.7.1.4.3 Add New Features and Elements

#### Topics

Topic	Detail	See also
<b>Usage</b>	<p>The following features and elements can be added to the source code during synchronization:</p> <ul style="list-style-type: none"> <li>• Attributes</li> <li>• Inner Classes</li> <li>• Operations</li> </ul> <p>These are added by executing the relevant templates for each new element or feature in the model.</p> <p>Enterprise Architect attempts to preserve the appropriate indenting of new features in the code, by finding the indents specified in list macros of the Class; for languages that make use of namespaces, the <i>synchNamespaceBodyIndent</i> macro is available.</p> <p>Classes defined within a (non-global) namespace are indented according to the value set for this macro, during synchronization.</p> <p>The value is ignored:</p> <ul style="list-style-type: none"> <li>• For Classes defined within a package setup as a root namespace, or</li> <li>• If the <b>Generate Namespaces</b> option is set to <b>False</b> in the appropriate language page (C#, C++ or VB.Net) on the Options dialog (<b>Tools   Options   Source Code Engineering   &lt;language&gt;</b>)</li> </ul>	<p><a href="#">Synchronization Macros</a> <sup>[1686]</sup></p> <p><a href="#">Language Options</a> <sup>[2262]</sup></p>

### 7.3.7.2 The Code Template Editor

The Code Template Editor provides the facilities of the *Common Code Editor*, including Intelli-sense for the various macros. For more information on Intelli-sense and the Common Code Editor, see the *Editing Source Code* topic.

**Access** **Tools | Source Code Generation Templates** (Ctrl+Shift+P)

#### Reference

Option	Action	See also
<b>Language</b>	Select the programming language.	
<b>New Language</b>	Display the Programming Languages Datatypes dialog, which enables you to include programming languages other than those supported for Enterprise Architect, for which to create or edit code templates.	<a href="#">Programming Languages Datatypes</a> <sup>[1171]</sup>

<b>Template</b>	Display the contents of the active template, and provides the editor for modifying templates.	
<b>Templates</b>	List the base code templates; the active template is highlighted.  The <b>Modified</b> field indicates whether you have changed the default template for the current language.	
<b>Stereotype Overrides</b>	List the stereotyped templates, for the active base template.  The <b>Modified</b> field indicates whether you have modified a default stereotyped template.	
<b>Add New Custom Template</b>	Invoke a dialog for creating a custom stereotyped template.	
<b>Add New Stereotyped Override</b>	Invoke a dialog for adding a stereotyped template, for the currently selected base template.	
<b>Get Default Template</b>	Update the editor display with the default version of the active template.	
<b>Save</b>	Overwrite the active templates with the contents of the editor.	
<b>Delete</b>	If you have overridden the active template, the override is deleted and replaced by the corresponding default code template.	

### Notes

- User-modified and user-defined Code Templates can be imported and exported as Reference Data (see the *Sharing Reference Data* topic); the templates defined for each language are indicated in the Export Reference Data dialog by the language name with the suffix *\_Code\_Templates* - if no templates exist for a language, there is no entry for the language in the dialog

### Learn more

- [The Code Template Editor in MDG Development](#)<sup>[170]</sup>
- [Sharing Reference Data](#)<sup>[374]</sup>
- [Editing Source Code](#)<sup>[2146]</sup>

### 7.3.7.3 Code Template Syntax

Code Templates are written using Enterprise Architect's Code Template Editor. The Code Template Editor supports syntax highlighting of the Code Template Framework language.

#### Topics

Topic	Detail	See also
<b>Basic Constructs</b>	Templates can contain: <ul style="list-style-type: none"> <li>• Literal Text</li> <li>• Variables</li> <li>• Macros</li> <li>• Calls to other templates</li> </ul>	<a href="#">Literal Text</a> <sup>[1643]</sup> <a href="#">Variables</a> <sup>[1644]</sup> <a href="#">Macros</a> <sup>[1646]</sup> <a href="#">Call Templates From Templates</a> <sup>[1700]</sup>
<b>Comments</b>	If you want to add comments to the templates, use the command: <pre>\$COMMENT=" t e x t "</pre> where "text" is the text of the comment; this must be enclosed in quotes. The command is case-sensitive, so must be typed in upper case.	

#### Learn more

- [Code Template Framework](#)<sup>[1637]</sup>
- [The Code Template Editor](#)<sup>[1701]</sup>

#### 7.3.7.3.1 Literal Text

All text within a given template that is not part of a macro or a variable definition/reference, is considered literal text. With the exception of blank lines, which are ignored, literal text is directly substituted from the template into the generated code.

Consider the following excerpt from the Java Class Declaration template:

```
$bases = "Base"
```

```
class %className%$bases
```

On the final line, the word *class*, including the subsequent space, would be treated as literal text and thus for a Class named *foo* would return the output:

```
class fooBase
```

The blank line following the variable `$bases` would have no effect on the output.

#### Inserting System Characters:

The %, \$, " and \ characters have special meaning in the template syntax and cannot always be used as literal text. If these characters must be generated from within the templates, they can be safely reproduced using the following direct substitution macros:

Macro	Action
<code>%dl%</code>	Produce a literal \$ character.
<code>%pc%</code>	Produce a literal % character.
<code>%qt%</code>	Produce a literal " character.
<code>%sl%</code>	Produce a literal \ character

#### Notes

- String conjunction operators (" + " , " += ") are not required but can be used (see literals)

#### Learn more

- [Intermediary Language](#) 

### 7.3.7.3.2 Variables

Template variables provide a convenient way of storing and retrieving data within a template. This section explains how variables are defined and referenced.

#### Variable Definitions:

Variable definitions take the basic form:

```
$<name> = <value>
```

where *<name>* can be any alpha-numeric sequence and *<value>* is derived from a macro or another variable.

A simple example definition would be:

```
$foo = %className%
```

Variables can be defined, using values from:

- Substitution, function or list macros
- String literals, enclosed within double quotation marks
- Variable references

#### Definition Rules:

The following rules apply to variable definitions:

- Variables have global scope within the template in which they are defined and are not accessible to other templates
- Each variable must be defined at the start of a line, without any intervening whitespace
- Variables are denoted by prefixing the name with \$, as in \$foo
- Variables do not have to be declared, prior to being defined
- Variables must be defined using either the assignment operator (=), or the addition-assignment operator (+=)
- Multiple terms can be combined in a single definition using the addition operator (+)

### **Examples**

Using a substitution macro:

```
$foo = %opTag: " bar " %
```

Using a literal string:

```
$foo = " bar "
```

Using another variable:

```
$foo = $bar
```

Using a list macro:

```
$ops = %list=" Operation" @separator="\n\n" @indent="\t" %
```

Using the addition-assignment operator (+=):

```
$body += %list=" Operation" @separator="\n\n" @indent="\t" %
```

The above definition is equivalent to the following:

```
$body = $body + %list=" Operation" @separator="\n\n" @indent="\t" %
```

Using multiple terms:

```
$templateArgs = %list=" ClassParameter" @separator=", " %  
$template = "template<" + $templateArgs + ">"
```

### **Variable References:**

Variable values can be retrieved by using a reference of the form:

```
$<name>
```

where <name> can be a previously defined variable.

Variable references can be used in one of the following ways:

- As part of a macro, such as the argument to a function macro
- As a term in a variable definition
- As a direct substitution of the variable value into the output

It is legal to reference a variable before it is defined. In this case, the variable is assumed to contain an empty string value: ""

**Example 1:**

Using variables as part of a macro. The following is an excerpt from the default C++ *ClassNotes* template.

```
%wrapLen = %genOpt WrapComment %
%style = %
genOpt CPPComment Style%
%i f %style == "XML.NET" %
%XML_COMMENT( $wrapLen) %
%el se%
%CSTYLE_COMMENT( $wrapLen) %
%endI f %
```

Define variables to store the style and wrap length options.

Reference to *\$style* as part of a condition.

Reference to *\$wrapLen* as an argument to function macro.

**Example 2:**

Using variable references as part of a variable definitions.

```
$foo = "foo"
$bar = "bar"

$foobar = $foo + $bar
```

Define our variables.

*\$foobar* now contains the value *foobar*.

**Example 3:**

Substituting variable values into the output.

```
$bases=%classInherits%
Class %className%$bases
```

Store the result of the *ClassInherits* template in *\$bases*.

Now output the value of *\$bases* after the Class name.

**7.3.7.3.3 Macros**

Macros provide access to element fields within the UML model and are also used to structure the generated output. All macros are enclosed within percent (%) signs in the form of:

```
%<macroName>%
```

In general, macros (including the % delimiters) are substituted with literal text in the output. For example, consider the following item from the *Class Declaration* template:

```
... class %className% ...
```

The field substitution macro, *%className%*, would result in the current Class name being substituted in the output. So if the Class being generated was named *Foo*, the output would be:

```
... class Foo ...
```

The CTF contains a number of types of macros:

- [Template Substitution Macros](#) <sup>[1647]</sup>
- [Field Substitution Macros](#) <sup>[1648]</sup>
  - [Substitution Examples](#) <sup>[1649]</sup>
  - [Attribute Field Substitution Macros](#) <sup>[1651]</sup>
  - [Class Field Substitution Macros](#) <sup>[1652]</sup>

- [Code Generation Option Field Substitution Macros](#) <sup>[1655]</sup>
- [Connector Field Substitution Macros](#) <sup>[1660]</sup>
- [Constraint Field Substitution Macros](#) <sup>[1664]</sup>
- [Effort Field Substitution Macros](#) <sup>[1665]</sup>
- [File Field Substitution Macros](#) <sup>[1665]</sup>
- [File Import Field Substitution Macros](#) <sup>[1666]</sup>
- [Link Field Substitution Macros](#) <sup>[1667]</sup>
- [Linked File Field Substitution Macros](#) <sup>[1668]</sup>
- [Metric Field Substitution Macros](#) <sup>[1669]</sup>
- [Operation Field Substitution Macros](#) <sup>[1669]</sup>
- [Package Field Substitution Macros](#) <sup>[1671]</sup>
- [Parameter Field Substitution Macros](#) <sup>[1672]</sup>
- [Problem Field Substitution Macros](#) <sup>[1673]</sup>
- [Requirement Field Substitution Macros](#) <sup>[1673]</sup>
- [Resource Field Substitution Macros](#) <sup>[1674]</sup>
- [Risk Field Substitution Macros](#) <sup>[1675]</sup>
- [Scenario Field Substitution Macros](#) <sup>[1675]</sup>
- [Tagged Value Substitution Macros](#) <sup>[1676]</sup>
- [Template Parameter Substitution Macros](#) <sup>[1677]</sup>
- [Test Field Substitution Macros](#) <sup>[1677]</sup>
- [Function Macros](#) <sup>[1678]</sup>
- [Control Macros](#) <sup>[1683]</sup>
  - [List Macro](#) <sup>[1683]</sup>
  - [Branching Macros](#) <sup>[1684]</sup>
  - [Synchronization Macros](#) <sup>[1686]</sup>
  - [The PI Macro](#) <sup>[1687]</sup>
- [EASL Code Generation Macros](#) <sup>[1688]</sup>

#### Learn more

- [Intermediary Language](#) <sup>[2053]</sup>

#### 7.3.7.3.1 Template Substitution Macros

Template substitution macros correspond to [Base templates](#) <sup>[1633]</sup>. These macros result in the execution of the named template. By convention, template macros are named according to Pascal casing.

Structure: %<TemplateName>%

where <TemplateName> can be one of the templates listed below.

When a template is referenced from within another template, it is generated with respect to the elements currently in scope. The specific template is selected based on the stereotypes of the elements in scope.

As noted previously, there is an implicit hierarchy among the various templates. Some care should be taken in order to preserve a sensible hierarchy of template references. For example, it does not make sense to use the `%ClassInherits%` macro within any of the attribute or operation templates. Conversely, the *Operation* and *Attribute* templates are designed for use **within** the *ClassBody* template.

The CTF contains the following template substitution macros:

- Attribute
- AttributeDeclaration
- AttributeDeclarationImpl
- AttributeNotes
- Class
- ClassBase
- ClassBody
- ClassBodyImpl
- ClassDeclaration
- ClassDeclarationImpl
- ClassImpl
- ClassInherits
- ClassInterface
- ClassNotes
- ClassParameter
- File
- FileImpl
- ImportSection
- ImportSectionImpl
- InnerClass
- InnerClassImpl
- LinkedAttribute
- LinkedAttributeDeclaration
- LinkedAttributeNotes
- LinkedClassBase
- LinkedClassInterface
- Namespace
- NamespaceBody
- NamespaceDeclaration
- NamespaceImpl
- Operation
- OperationBody
- OperationBodyImpl
- OperationDeclaration
- OperationDeclarationImpl
- OperationImpl
- OperationNotes
- Parameter

#### 7.3.7.3.3.2 Field Substitution Macros

The **field substitution** macros provide access to data in your model. In particular, they are used to access data fields from:

- Packages
- Classes
- Attributes
- Operations
- Parameters

Field substitution macros are named according to Camel casing. By convention, the macro is prefixed with an abbreviated form of the corresponding model element. For example, attribute-related macros begin with **att**, as in the `%attName%` macro, to access the name of the attribute in scope.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

The following table lists a small number of project field substitution macros with a description of the result. See the *Learn more* section for tables listing groups of type-specific macros.

Macro Name	Description
eaDateTime	The current time with format: <i>DD-MMM-YYYY HH:MM:SS AM/PM</i> .
eaGUID	A unique GUID for this generation.



Macro Name	Description
eaVersion	Program Version (located in an Enterprise Architect dialog by selecting <b>Help   About EA</b> ).

### Learn more

- [Substitution Examples](#) <sup>[1649]</sup>
- [Attribute Field Substitution Macros](#) <sup>[1651]</sup>
- [Class Field Substitution Macros](#) <sup>[1652]</sup>
- [Code Generation Option Field Substitution Macros](#) <sup>[1655]</sup>
- [Connector Field Substitution Macros](#) <sup>[1660]</sup>
- [Constraint Field Substitution Macros](#) <sup>[1664]</sup>
- [Effort Field Substitution Macros](#) <sup>[1665]</sup>
- [File Field Substitution Macros](#) <sup>[1665]</sup>
- [File Import Field Substitution Macros](#) <sup>[1666]</sup>
- [Link Field Substitution Macros](#) <sup>[1667]</sup>
- [Linked File Field Substitution Macros](#) <sup>[1668]</sup>
- [Metric Field Substitution Macros](#) <sup>[1669]</sup>
- [Operation Field Substitution Macros](#) <sup>[1669]</sup>
- [Package Field Substitution Macros](#) <sup>[1671]</sup>
- [Parameter Field Substitution Macros](#) <sup>[1672]</sup>
- [Problem Field Substitution Macros](#) <sup>[1673]</sup>
- [Requirement Field Substitution Macros](#) <sup>[1673]</sup>
- [Resource Field Substitution Macros](#) <sup>[1674]</sup>
- [Risk Field Substitution Macros](#) <sup>[1675]</sup>
- [Scenario Field Substitution Macros](#) <sup>[1675]</sup>
- [Tagged Value Substitution Macros](#) <sup>[1676]</sup>
- [Template Parameter Substitution Macros](#) <sup>[1677]</sup>
- [Test Field Substitution Macros](#) <sup>[1677]</sup>

Field substitution macros can be used in one of two ways:

- Direct Substitution or
- Conditional Substitution

### Direct Substitution

This form directly substitutes the corresponding value of the element in scope into the output.

Structure: %<macroName>%

Where <macroName> can be any of the macros listed in the Field Substitution Macros tables.

### Examples

- `%className%`
- `%opName%`
- `%attName%`

### Conditional Substitution

This form of the macro enables alternative substitutions to be made depending on the macro's value.

Structure: `%<macroName> ( == " <text> " ) ? <subTrue> ( : <subFalse> ) %`

Where:

- `( )` denotes that values between the parentheses are optional
- `<text>` is a string representing a possible value for the macro
- `<subTrue>` and `<subFalse>` can be a combination of quoted strings and the keyword value; where the value is used, it is replaced with the macro's value in the output.

### Examples

- `%classAbstract=="T" ? "pure" : ""%`
- `%opStereotype=="operator" ? "operator" : ""%`
- `%paramDefault != "" ? " = " value : ""%`

The above three examples output nothing if the condition fails. In this case the false condition can be omitted, resulting in the following usage:.

- `%classAbstract=="T" ? "pure" %`
- `%opStereotype=="operator" ? "operator" %`
- `%paramDefault != "" ? " = " value%`

The third example of both blocks shows a comparison checking for a non-empty value or existence. This test can also be omitted.

- `%paramDefault ? " = " value : ""%`
- `%paramDefault ? " = " value%`

All of the above examples containing `paramDefault` are equivalent. If the parameter in scope had a default value of **10**, the output from each of them would normally be:

`= 10`

### Notes

- In a conditional substitution macro, any white space following `<macroName>` is ignored; if white space is required in the output, it should be included within the quoted substitution strings

### Learn more

- [Field Substitution Macros](#)<sup>[1648]</sup>

The following table lists each of the attribute field substitution macros with a description of the result.

Field substitution macros are named according to Camel casing.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description
<b>attAlias</b>	Attributes dialog: <b>Alias</b> .
<b>attAllowDuplicates</b>	Attributes Detail dialog: <b>Allow Duplicates</b> checkbox.
<b>attClassifierGUID</b>	The unique GUID for the classifier of the current attribute.
<b>attCollection</b>	Attributes Detail dialog: <b>Attribute is a Collection</b> checkbox.
<b>attConst</b>	Attributes dialog: <b>Const</b> checkbox.
<b>attContainerType</b>	Attributes Detail dialog: <b>Container Type</b> .
<b>attContainment</b>	Attributes dialog: <b>Containment</b> .
<b>attDerived</b>	Attributes dialog: <b>Derived</b> checkbox.
<b>attGUID</b>	The unique GUID for the current attribute.
<b>attInitial</b>	Attributes dialog: <b>Initial</b> .
<b>attIsEnumLiteral</b>	Attributes dialog: <b>Is Literal</b> checkbox.
<b>attIsID</b>	Attributes Detail dialog: <b>isID</b> checkbox.
<b>attLength</b>	Column dialog: <b>Length</b> .
<b>attLowerBound</b>	Attributes Detail dialog: <b>Lower Bound</b> .
<b>attName</b>	Attributes dialog: <b>Name</b> .

Macro Name	Description
<b>attNotes</b>	Attributes dialog: <b>Notes</b> .
<b>attOrderedMultiplicity</b>	Attributes Detail dialog: <b>Ordered Multiplicity</b> checkbox.
<b>attProperty</b>	Attributes dialog: <b>Property</b> checkbox.
<b>attQualType</b>	The attribute type qualified by the namespace path (if generating namespaces) and the classifier path (dot delimited). If the attribute classifier has not been set, is equivalent to the <i>attType</i> macro.
<b>attScope</b>	Attributes dialog: <b>Scope</b> .
<b>attStatic</b>	Attributes dialog: <b>Static</b> checkbox.
<b>attStereotype</b>	Attributes dialog: <b>Stereotype</b> .
<b>attType</b>	Attributes dialog: <b>Type</b> .
<b>attUpperBound</b>	Attributes Detail dialog: <b>Upper Bound</b> .
<b>attVolatile</b>	Attributes Detail dialog: <b>Transient</b> checkbox.

#### Learn more

- [Field Substitution Macros](#)<sup>[1648]</sup>

The following table provides a list of how to access each available Class property in the Code Generation and Transformation templates.

Field substitution macros are named according to Camel casing.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description	See also
<b>elemType</b>	The element type: Interface or Class.	
<b>classAbstract</b>	Class dialog: <b>Abstract</b> checkbox.	

Macro Name	Description	See also
<b>classAlias</b>	Class dialog: <b>Alias</b> .	
<b>classArguments</b>	Class Detail dialog: <b>C++ Templates: Arguments</b> .	
<b>classAuthor</b>	Class dialog: <b>Author</b> .	
<b>classBaseName</b>	Type Hierarchy dialog: <b>Class Name</b> (for use where no connector exists between child and base Classes).	
<b>classBaseScope</b>	The scope of the inheritance as reverse engineered. (For use where no connector exists between child and base Classes.)	
<b>classBaseVirtual</b>	The virtual property of the inheritance as reverse engineered. (For use where no connector exists between child and base Classes.)	
<b>classComplexity</b>	Class dialog: <b>Complexity</b> .	
<b>classCreated</b>	The date and time the Class was created.	
<b>classGUID</b>	The unique GUID for the current Class.	
<b>classHasConstructor</b>	Looks at the list of methods in the current object and, depending on the conventions of the current language, returns <b>T</b> if one is a default constructor. Typically used with the <b>genOptGenConstructor</b> macro.	<a href="#">Code Generation Option Field Substitution Macros</a> <small>1655</small>
<b>classHasCopyConstructor</b>	Looks at the list of methods in the current object and, depending on the conventions of the current language, returns <b>T</b> if one is a copy constructor. Typically used with the <b>genOptGenCopyConstructor</b> macro.	
<b>classHasDestructor</b>	Looks at the list of methods in the current object and, depending on the conventions of the current language, returns <b>T</b> if one is a destructor. Typically used with the <b>genOptGenDestructor</b> macro.	
<b>classHasParent</b>	<b>True</b> , if the Class in scope has one or more base Classes.	
<b>classImports</b>	Code Gen dialog: <b>Imports</b> .	

Macro Name	Description	See also
<b>classIsActive</b>	Class Advanced dialog: <b>Is Active</b> checkbox.	
<b>classIsAssociationClass</b>	<b>True</b> , if the Association is an AssociationClass connector.	
<b>classIsInstantiated</b>	<b>True</b> , if the Class is an instantiated template Class.	
<b>classIsLeaf</b>	Class Advanced dialog: <b>Is Leaf</b> checkbox.	
<b>classIsRoot</b>	Class Advanced dialog: <b>Is Root</b> checkbox.	
<b>classIsSpecification</b>	Class Advanced dialog: <b>Is Specification</b> checkbox.	
<b>classKeywords</b>	Class dialog: <b>Keywords</b> .	
<b>classLanguage</b>	Class dialog: <b>Language</b> .	
<b>classMacros</b>	A space separated list of macros defined for the Class.	
<b>classModified</b>	The date and time the Class was last modified.	
<b>classMultiplicity</b>	Class Advanced dialog: <b>Multiplicity</b> .	
<b>className</b>	Class dialog: <b>Name</b> .	
<b>classNotes</b>	Class dialog: <b>Note</b> .	
<b>classParamDefault</b>	Class Detail dialog.	
<b>classParamName</b>	Class Detail dialog.	
<b>classParamType</b>	Class Detail dialog.	
<b>classPersistence</b>	Class dialog: <b>Persistence</b> .	

Macro Name	Description	See also
<b>classPhase</b>	Class dialog: <b>Phase</b> .	
<b>classQualName</b>	The Class name prefixed by its outer Classes. Class names are separated by double colons (::).	
<b>classScope</b>	Class dialog: <b>Scope</b> .	
<b>classStereotype</b>	Class dialog: <b>Stereotype</b> .	
<b>classStatus</b>	Class dialog: <b>Status</b> .	
<b>classVersion</b>	Class dialog: <b>Version</b> .	

#### Learn more

- [Field Substitution Macros](#)<sup>[1648]</sup>

The following table lists each of the code generation option field substitution macros with a description of the result.

These macros operate on the source code generation options defined in the Source Code Engineering pages of the Options dialog (**Tools | Options | Source Code Engineering**).

Field substitution macros are named according to Camel casing.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description
genOptActionScriptVersion	ActionScript Specifications page: <b>Default Version</b> .
genOptCDefaultAttributeType	C Specifications page: <b>Default Attribute Type</b> .
genOptCGenMethodNotesInBody	C Specifications page: <b>Method Notes In Implementation</b> .
genOptCGenMethodNotesInHeader	C Specifications page: <b>Method Notes In Header</b> .
genOptCSynchNotes	C Specifications page: <b>Synchronize Notes in Generation</b> .
genOptCSynchCFile	C Specifications page: <b>Synchronise Implementation file in Generation</b> .

Macro Name	Description
genOptCDefaultSourceDirectory	C Specifications page: <b>Default Source Directory</b> .
genOptCNamespaceDelimiter	C Specifications page: <b>Namespace Delimiter</b> .
genOptCOperationRefParam	C Specifications page: <b>Reference as Operation Parameter</b> .
genOptCOperationRefParamStyle	C Specifications page: <b>Reference Parameter Style</b> .
genOptCOperationRefParamName	C Specifications page: <b>Reference Parameter Name</b> .
genOptCConstructorName	C Specifications page: <b>Default Constructor Name</b> .
genOptCDestructorName	C Specifications page: <b>Default Destructor Name</b> .
genOptCPPCommentStyle	C++ Specifications page: <b>Comment Style</b> .
genOptCPPDefaultAttributeType	C++ Specifications page: <b>Default Attribute Type</b> .
genOptCPPDefaultReferenceType	C++ Specifications page: <b>Default Reference Type</b> .
genOptCPPDefaultSourceDirectory	C++ Specifications page: <b>Default Source Directory</b> .
genOptCPPGenMethodNotesInHeader	C++ Specifications page: <b>Method Notes In Header</b> checkbox.
genOptCPPGenMethodNotesInBody	C++ Specifications page: <b>Method Notes In Body</b> checkbox.
genOptCPPGetPrefix	C++ Specifications page: <b>Get Prefix</b> .
genOptCPPHeaderExtension	C++ Specifications page: <b>Header Extension</b> .
genOptCPPSetPrefix	C++ Specifications page: <b>Set Prefix</b> .
genOptCPPSourceExtension	C++ Specifications page: <b>Source Extension</b> .



Macro Name	Description
genOptCPPSynchCPPFile	C++ Specifications page: <b>Synchronize Notes</b> .
genOptCPPSynchNotes	C++ Specifications page: <b>Synchronize CPP File</b> .
genOptCSDefaultAttributeType	C# Specifications page: <b>Default Attribute Type</b> .
genOptCSSourceExtension	C# Specifications page: <b>Default file extension</b> .
genOptCSGenDispose	C# Specifications page: <b>Generate Dispose</b> .
genOptCSGenFinalizer	C# Specifications page: <b>Generate Finalizer</b> .
genOptCSGenNamespace	C# Specifications page: <b>Generate Namespace</b> .
genOptCSDefaultSourceDirectory	C# Specifications page: <b>Default Source Directory</b> .
genOptDefaultAssocAttName	Attribute/Operations Specifications page: <b>Default name for associated</b> .
genOptDefaultConstructorScope	Object Lifetimes page: <b>Default Constructor Visibility</b> .
genOptDefaultCopyConstructorScope	Object Lifetimes page: <b>Default Copy Constructor Visibility</b> .
genOptDefaultDatabase	Code Editors page: <b>Default Database</b> .
genOptDefaultDestructorScope	Object Lifetimes page: <b>Default Destructor Constructor Visibility</b> .
genOptGenCapitalisedProperties	Source Code Engineering page: <b>Capitalize Attribute Names for Properties</b> checkbox.
genOptGenComments	Source Code Engineering page: <b>Generate Comments</b> checkbox.
genOptGenConstructor	Object Lifetimes page: <b>Generate Constructor</b> checkbox.
genOptGenConstructorInline	Object Lifetimes page: <b>Constructor Inline</b> checkbox.

Macro Name	Description
genOptGenCopyConstructor	Object Lifetimes page: <b>Generate Copy Constructor</b> checkbox.
genOptGenCopyConstructorInline	Object Lifetimes page: <b>Copy Constructor Inline</b> checkbox.
genOptGenDestructor	Object Lifetimes page: <b>Generate Destructor</b> checkbox.
genOptGenDestructorInline	Object Lifetimes page: <b>Destructor Inline</b> checkbox.
genOptGenDestructorVirtual	Object Lifetimes page: <b>Virtual Destructor</b> checkbox.
genOptGenImplementedInterfaceOps	Attribute/Operations Specifications page: <b>Generate methods for implemented interfaces</b> checkbox.
genOptGenPrefixBoolProperties	Source Code Engineering page: <b>Use is prefix for boolean property Get()</b> .
genOptGenRoleNames	Source Code Engineering page: <b>Autogenerate role names when creating code</b> .
genOptGenUnspecAssocDir	Source Code Engineering page: <b>Do not generate members where Association direction is unspecified</b> checkbox.
genOptJavaDefaultAttributeType	Java Specifications page: <b>Default attribute type</b> .
genOptJavaGetPrefix	Java Specifications page: <b>Get Prefix</b> .
genOptJavaDefaultSourceDirectory	Java Specifications page: <b>Default Source Directory</b> .
genOptJavaSetPrefix	Java Specifications page: <b>Set Prefix</b> .
genOptJavaSourceExtension	Java Specifications page: <b>Source code extension</b> .
genOptPHPDefaultSourceDirectory	PHP Specifications page: <b>Default Source Directory</b> .
genOptPHPGetPrefix	PHP Specifications page: <b>Get Prefix</b> .

Macro Name	Description
genOptPHPSetPrefix	PHP Specifications page: <b>Set Prefix</b> .
genOptPHPSourceExtension	PHP Specifications page: <b>Default file extension</b> .
genOptPHPVersion	PHP Specifications page: <b>PHP Version</b> .
genOptPropertyPrefix	Source Code Engineering page: <b>Remove prefixes when generating Get/Set properties</b> .
genOptVBMultiUse	VB Specifications page: <b>Multiuse</b> checkbox.
genOptVBPersistable	VB Specifications page: <b>Persistable</b> checkbox.
genOptVBDataBindingBehavior	VB Specifications page: <b>Data binding behavior</b> checkbox.
genOptVBDataSourceBehavior	VB Specifications page: <b>Data source behavior</b> checkbox.
genOptVBGlobal	VB Specifications page: <b>Global namespace</b> checkbox.
genOptVBCreatable	VB Specifications page: <b>Creatable</b> checkbox.
genOptVBExposed	VB Specifications page: <b>Exposed</b> checkbox.
genOptVBMTS	VB Specifications page: <b>MTS Transaction Mode</b> .
genOptVBNetGenNamespace	VB.Net Specifications page: <b>Generate Namespace</b> .
genOptVBVersion	VB Specifications page: <b>Default Version</b> .
genOptWrapComment	Source Code Engineering page: <b>Wrap length for comment lines</b> .

#### Learn more

- [Field Substitution Macros](#) 

The following table lists each of the **connector** field substitution macros with a description of the result.

Field substitution macros are named according to **Camel** casing. Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description	See also
<b>connectorAlias</b>	Connector Properties dialog: <b>Alias</b> .	
<b>connectorAssociationClassElementGUID</b>	The GUID of the connector's Association Class element.	
<b>connectorAssociationClassElementName</b>	The name of the connector's Association Class element.	
<b>connectorDestAccess</b>	Connector Properties dialog, Target Role tab: <b>Access</b> .	
<b>connectorDestAggregation</b>	Connector Properties dialog, Target Role tab: <b>Aggregation</b> .	
<b>connectorDestAlias</b>	Connector Properties dialog, Target Role tab: <b>Alias</b> .	
<b>connectorDestAllowDuplicates</b>	Connector Properties dialog, Target Role tab: <b>Allow Duplicates</b> checkbox.	
<b>connectorDestChangeable</b>	Connector Properties dialog, Target Role tab: <b>Changeable</b> .	
<b>connectorDestConstraint</b>	Connector Properties dialog, Target Role tab: <b>Constraint(s)</b> .	
<b>connectorDestContainment</b>	Connector Properties dialog, Target Role tab: <b>Containment</b> .	
<b>connectorDestDerived</b>	Connector Properties dialog, Target Role tab: <b>Derived</b> checkbox.	
<b>connectorDestDerivedUnion</b>	Connector Properties dialog, Target Role tab: <b>DerivedUnion</b> checkbox.	
<b>connectorDestElem*</b>	<p>A set of macros that access a property of the element at the target end of a connector. The * (asterisk) is a wildcard that corresponds to any Class substitution macro in the Class macro list. For example:</p> <ul style="list-style-type: none"> <li><i>connectorDestElemAlias (classAlias)</i></li> <li><i>connectorDestElemAuthor (classAuthor)</i></li> </ul>	<a href="#">Class Field Substitution Macros</a> <small>[1652]</small>

Macro Name	Description	See also
<b>connectorDestElemType</b>	The element type of the connector destination element. (Separate from the <i>connectorDestElem*</i> macros because there is no <i>classType</i> substitution macro.)	
<b>connectorDestFeature*</b>	A set of macros that access a property of the feature at the target end of a connector. The * (asterisk) is a wildcard that corresponds to any attribute or operation substitution macro in the Attribute macro or Operation macro list, depending on the <i>connectorDestFeatureType</i> .  For example: <ul style="list-style-type: none"> <li><i>connectorDestFeatureReturnClassifierGUID</i> - an Operation's return classifier GUID</li> <li><i>connectorDestFeatureContainment</i> - an Attribute's containment</li> </ul>	<a href="#">Attribute Field Substitution Macros</a> [1651]  <a href="#">Operation Field Substitution Macros</a> [1669]
<b>connectorDestFeatureType</b>	The type of the connector destination feature. <ul style="list-style-type: none"> <li><i>connectorDestFeatureType</i>="Attribute" or "Operation"</li> </ul>	
<b>connectorDestMemberType</b>	Connector Properties dialog, Target Role tab: <b>Member Type</b> .	
<b>connectorDestMultiplicity</b>	Connector Properties dialog, Target Role tab: <b>Multiplicity</b> .	
<b>connectorDestNavigability</b>	Connector Properties dialog, Target Role tab: <b>Navigability</b> .	
<b>connectorDestNotes</b>	Connector Properties dialog, Target Role tab: <b>Role Notes</b> .	
<b>connectorDestOrdered</b>	Connector Properties dialog, Target Role tab: <b>Ordered</b> checkbox.	
<b>connectorDestOwned</b>	Connector Properties dialog, Target Role tab: <b>Owned</b> checkbox.	
<b>connectorDestQualifier</b>	Connector Properties dialog, Target Role tab: <b>Qualifier(s)</b> .	
<b>connectorDestRole</b>	Connector Properties dialog, Target Role tab: <b>Role</b> .	
<b>connectorDestScope</b>	Connector Properties dialog, Target Role tab: <b>Target Scope</b> .	
<b>connectorDestStereotype</b>	Connector Properties dialog, Target Role tab: <b>Stereotype</b> .	

Macro Name	Description	See also
<b>connectorDirection</b>	Connector Properties: <b>Direction</b> .	
<b>connectorEffect</b>	Transition Constraints dialog: <b>Effect</b> .	
<b>connectorGuard</b>	Object Flow and Transition Constraints dialog: <b>Guard</b> .	
<b>connectorGUID</b>	The unique GUID for the current connector.	
<b>connectorIsAssociationClass</b>	<b>True</b> , if the connector is an <b>AssociationClass</b> connector.	
<b>connectorName</b>	Connector Properties: <b>Name</b> .	
<b>connectorNotes</b>	Connector Properties: <b>Notes</b> .	
<b>connectorSourceAccess</b>	Connector Properties dialog, Source Role tab: <b>Access</b> .	
<b>connectorSourceAggregation</b>	Connector Properties dialog, Source Role tab: <b>Aggregation</b> .	
<b>connectorSourceAlias</b>	Connector Properties dialog, Source Role tab: <b>Alias</b> .	
<b>connectorSourceAllowDuplicates</b>	Connector Properties dialog, Source Role tab: <b>Allow Duplicates</b> checkbox.	
<b>connectorSourceChangeable</b>	Connector Properties dialog, Source Role tab: <b>Changeable</b> .	
<b>connectorSourceConstraint</b>	Connector Properties dialog, Source Role tab: <b>Constraint(s)</b> .	
<b>connectorSourceContainment</b>	Connector Properties dialog, Source Role tab: <b>Containment</b> .	
<b>connectorSourceDerived</b>	Connector Properties dialog, Source Role tab: <b>Derived</b> checkbox.	
<b>connectorSourceDerivedUnion</b>	Connector Properties dialog, Source Role tab: <b>DerivedUnion</b> checkbox.	

Macro Name	Description	See also
<b>connectorSourceElem*</b>	<p>A set of macros that access a property of the element at the source end of a connector. The * (asterisk) is a wildcard that corresponds to any class substitution macro in the Class macro list. For example:</p> <ul style="list-style-type: none"> <li><i>connectorSourceElemAlias</i> (<i>classAlias</i>)</li> <li><i>connectorSourceElemAuthor</i> (<i>classAuthor</i>)</li> </ul>	<a href="#">Class Field Substitution Macros</a> <small>[1652]</small>
<b>connectorSourceElemType</b>	The element type of the connector source element. (Separate from the <i>connectorSourceElem*</i> macros because there is no <i>classType</i> substitution macro.)	
<b>connectorSourceFeature*</b>	<p>A set of macros that access a property of the feature at the source end of a connector. The * (asterisk) is a wildcard that corresponds to any attribute or operation substitution macro in the Attribute macro or Operation macro list, depending on the <i>connectorSourceFeatureType</i>. For example:</p> <ul style="list-style-type: none"> <li><i>connectorSourceFeatureCode</i> - Operation's Code</li> <li><i>connectorSourceFeatureInitial</i> - Attribute's Initial</li> </ul>	<a href="#">Attribute Field Substitution Macros</a> <small>[1651]</small> <a href="#">Operation Field Substitution Macros</a> <small>[1669]</small>
<b>connectorSourceFeatureType</b>	<p>The type of the connector source feature.</p> <ul style="list-style-type: none"> <li><i>connectorSourceFeatureType</i>="Attribute" or "Operation"</li> </ul>	
<b>connectorSourceMemberType</b>	Connector Properties dialog, Source Role tab: <b>Member Type</b> .	
<b>connectorSourceMultiplicity</b>	Connector Properties dialog, Source Role tab: <b>Multiplicity</b> .	
<b>connectorSourceNavigability</b>	Connector Properties dialog, Source Role tab: <b>Navigability</b> .	
<b>connectorSourceNotes</b>	Connector Properties dialog, Source Role tab: <b>Role Notes</b> .	
<b>connectorSourceOrdered</b>	Connector Properties dialog, Source Role tab: <b>Ordered</b> checkbox.	
<b>connectorSourceOwned</b>	Connector Properties dialog, Source Role tab: <b>Owned</b> checkbox.	
<b>connectorSourceQualifier</b>	Connector Properties dialog, Source Role tab: <b>Qualifier(s)</b> .	

Macro Name	Description	See also
<b>connectorSourceRole</b>	Connector Properties dialog, Source Role tab: <b>Role</b> .	
<b>connectorSourceScope</b>	Connector Properties dialog, Source Role tab: <b>Target Scope</b> .	
<b>connectorSourceStereotype</b>	Connector Properties dialog, Source Role tab: <b>Stereotype</b> .	
<b>connectorStereotype</b>	Connector Properties dialog: <b>Stereotype</b> .	
<b>connectorTrigger</b>	Transition Constraints dialog: <b>Trigger</b> .	
<b>connectorType</b>	The connector type; for example, <b>Association</b> or <b>Generalization</b> .	
<b>connectorWeight</b>	Object Flow Constraints dialog: <b>Weight</b> .	

### Learn more

- [Field Substitution Macros](#) <sup>1648</sup>

The following table lists each of the Constraint field substitution macros with a description of the result. Field substitution macros are named according to Camel casing.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description
<b>constraintName</b>	Class dialog, Constraints tab: <b>Name</b> .
<b>constraintNotes</b>	Class dialog, Constraints tab: <b>Notes</b> .
<b>constraintStatus</b>	Class dialog, Constraints tab: <b>Status</b> .
<b>constraintType</b>	Class dialog, Constraints tab: <b>Type</b> .
<b>constraintWeight</b>	Class dialog, Constraints tab: ordering (hand up/down) keys.



The following table lists each of the Effort field substitution macros with a description of the result.

Field substitution macros are named according to Camel casing.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description
effortName	Project Management window: <b>Effort</b> .
effortNotes	Project Management window: <b>Notes</b> (unlabelled).
effortTime	Project Management window: <b>Time</b> .
effortType	Project Management window: <b>Type</b> .

The following table lists each of the Class field substitution macros with a description of the result.

Field substitution macros are named according to Camel casing.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description
fileExtension	The file type extension of the file being generated.
fileName	The name of the file being generated.
fileNameImpl	The filename of the implementation file for this generation, if applicable.
fileHeaders	Code Gen dialog: <b>Headers</b> .
fileImports	Code Gen dialog: <b>Imports</b> . For supported languages this also includes dependencies derived from associations.
filePath	The full path of the file being generated.
filePathImpl	The full path of the implementation file for this generation, if applicable.

The following table lists each of the file import field substitution macros with a description of the result.

Field substitution macros are named according to Camel casing.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description
importClassName	The name of the Class being imported.
importFileName	The filename of the Class being imported.
importFilePath	The full path of the Class being imported.
importFromAggregation	<b>T</b> if the Class has an Aggregation connector to a Class in this file, <b>F</b> otherwise.
importFromAssociation	<b>T</b> if the Class has an Association connector to a Class in this file, <b>F</b> otherwise.
importFromAtt	<b>T</b> if an attribute of a Class in the current file is of the type of this Class, <b>F</b> otherwise.
importFromDependency	<b>T</b> if the Class has a Dependency connector to a Class in this file, <b>F</b> otherwise.
importFromGeneralization	<b>T</b> if the Class has a Generalization connector to a Class in this file, <b>F</b> otherwise.
importFromMeth	<b>T</b> if a method return type of a Class in the current file is the type of this Class, <b>F</b> otherwise.
importFromParam	<b>T</b> if an method parameter of a Class in the current file is of the type of this Class, <b>F</b> otherwise.
importFromRealization	<b>T</b> if the Class has a Realization connector to a Class in this file, <b>F</b> otherwise.
importFromTemplateBinding	<b>T</b> if the Class has a TemplateBinding connector to a Class in this file, <b>F</b> otherwise.
importInFile	<b>T</b> if the Class is in the current file, <b>F</b> otherwise.
importPackagePath	The package path with a '.' separator of the Class being imported.
ImportRelativeFilePath	The relative file path of the Class being imported from the file path of the file

Macro Name	Description
	being generated.

### Learn more

- [Field Substitution Macros](#) <sup>1648</sup>

If you want to provide access to data concerning connectors in the model, particularly Associations and Generalizations, you can use the **Link field substitution macros**. The macro names are in **Camel** casing. Macros that represent checkboxes return a value of **T** if the box is selected; otherwise the value is empty.

### Link field substitution macros

Macro Name	Description/Result	See also
<b>linkAttAccess</b>	Association Properties dialog, Target Role: <b>Access</b> .	
<b>linkAttAggregation</b>	Association Properties dialog, Source or Target Role: <b>Aggregation</b> .	
<b>linkAttCollectionClasses</b>	The collection appropriate for the linked attribute in scope.	
<b>linkAttContainment</b>	Association Properties dialog, Target Role: <b>Containment</b> .	
<b>linkAttName</b>	Association Properties dialog: <b>Target</b> .	
<b>linkAttNotes</b>	Association Properties dialog, Target Role: <b>Role Notes</b> .	
<b>linkAttOwnedByAssociation</b>	<b>True</b> , if the <b>Owned</b> checkbox on the Source or Target Role page of the Association Properties dialog is <b>not</b> selected.	
<b>linkAttOwnedByClass</b>	<b>True</b> , if the <b>Owned</b> checkbox on the Source or Target Role page of the Association Properties dialog <b>is</b> selected.	
<b>linkAttQualName</b>	The Association target qualified by the namespace path (if generating namespaces) and the classifier path (dot delimited).	
<b>linkAttRole</b>	Association Properties dialog, Target Role: <b>Role</b> .	

Macro Name	Description/Result	See also
<b>linkAttStereotype</b>	Association Properties dialog, Target Role: <b>Stereotype</b> .	
<b>linkAttTargetScope</b>	Association Properties dialog, Target Role: <b>Target Scope</b> .	
<b>linkCard</b>	Link Properties dialog, Target Role: <b>Multiplicity</b> .	
<b>linkGUID</b>	The unique GUID for the current connector.	
<b>linkIsAssociationClasses</b>	<b>True</b> , if the Association is an <b>AssociationClass</b> connector.	
<b>linkIsBound</b>	Returns <b>T</b> if any TemplateBindings are specified on the connector.	
<b>linkParamSubs</b>	Returns a comma-separated list of the arguments specified.	
<b>linkParentName</b>	Generalization Properties dialog: <b>Target</b> .	
<b>linkParentQualName</b>	The Generalization target qualified by the namespace path (if generating namespaces) and the classifier path (dot delimited).	
<b>linkStereotype</b>	The stereotype of the current connector.	
<b>linkVirtualInheritance</b>	Generalization Properties dialog: <b>Virtual Inheritance</b> .	

#### Learn more

- [Macros](#) <sup>[1646]</sup>
- [Field Substitution Macros](#) <sup>[1648]</sup>

The following table lists each of the Linked File field substitution macros with a description of the result.

Field substitution macros are named according to Camel casing.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description
linkedFileLastWrite	Class Properties dialog: <b>Last Write</b> .

Macro Name	Description
linkedFileNotes	Class Properties dialog: <b>Notes</b> .
linkedFilePath	Class Properties dialog: <b>File Path</b> .
linkedFileSize	Class Properties dialog: <b>Size</b> .
linkedFileType	Class Properties dialog: <b>Type</b> .

The following table lists each of the Metric field substitution macros with a description of the result.

Field substitution macros are named according to Camel casing.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description
metricName	Project Management dialog, Metrics tab: <b>Metric</b> field.
metricNotes	Project Management dialog, Metrics tab: (Notes) field.
metricType	Project Management dialog, Metrics tab: <b>Type</b> field.
metricWeight	Project Management dialog, Metrics tab: <b>Weight</b> field.

The **Operation field substitution macros** provide access to data concerning operations in the model. The macro names are in Camel casing. Macros that represent checkboxes return a value of **T** if the box is selected; otherwise the value is empty.

#### Operation field substitution macros

Macro Name	Description/Result
opAbstract	Operation dialog: <b>Virtual</b> checkbox.
opAlias	Operation dialog: <b>Alias</b> .
opBehavior	Operation Behavior dialog: <b>Behavior</b> .

Macro Name	Description/Result
opCode	Operation Behavior dialog: <b>Initial Code</b> .
opConcurrency	Operation dialog: <b>Concurrency</b> .
opConst	Operation dialog: <b>Const</b> checkbox.
opGUID	The unique GUID for the current operation.
opHasSelfRefParam	Scans the list of parameters in the current Operation, returning <b>T</b> if one type is the Class reference (this could be <b>ClassA*</b> or <b>ClassA&amp;</b> , depending on the value of the <b>genOptCOperationRefParamStyle</b> code generation option field substitution macro).
opImplMacros	A space-separated list of macros defined in the implementation of this operation.
opIsQuery	Operation dialog: <b>IsQuery</b> checkbox.
opMacros	A space-separated list of macros defined in the declaration for this operation.
opName	Operation dialog: <b>Name</b> .
opNotes	Operation dialog: <b>Notes</b> .
opPure	Operation dialog: <b>Pure</b> checkbox.
opReturnArray	Operation dialog: <b>Return Array</b> checkbox.
opReturnClassifierGUID	The unique GUID for the classifier of the current operation.
opReturnQualType	The operation return type qualified by the namespace path (if generating namespaces) and the classifier path (dot delimited). If the return type classifier has not been set, it is equivalent to the <i>opReturnType</i> macro.
opReturnType	Operation dialog: <b>Return Type</b> .
opScope	Operation dialog: <b>Scope</b> .

Macro Name	Description/Result
opStatic	Operation dialog: <b>Static</b> checkbox.
opStereotype	Operation dialog: <b>Stereotype</b> .
opSynchronized	Operation dialog: <b>Synchronized</b> checkbox.

### Learn more

- [Field Substitution Macros](#) <sup>[1648]</sup>
- [Code Generation Option Field Substitution Macros](#) <sup>[1656]</sup>

The following table lists each of the Package field substitution macros with a description of the result.

Field substitution macros are named according to Camel casing.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description
packageAbstract	Package dialog: <b>Abstract</b> .
packageAlias	Package dialog: <b>Alias</b> .
packageAuthor	Package dialog: <b>Author</b> .
packageComplexity	Package dialog: <b>Complexity</b> .
packageGUID	The unique GUID for the current package.
packageKeywords	Package dialog: <b>Keywords</b> .
packageLanguage	Package dialog: <b>Language</b> .
packageName	Package dialog: <b>Name</b> .
packagePath	The string representing the hierarchy of packages, for the Class in scope. Each package name is separated by a dot (.).

Macro Name	Description
packagePhase	Package dialog: <b>Phase</b> .
packageScope	Package dialog: <b>Scope</b> .
packageStatus	Package dialog: <b>Status</b> .
packageStereotype	Package dialog: <b>Stereotype</b> .
packageVersion	Package dialog: <b>Version</b> .

#### Learn more

- [Field Substitution Macros](#) <sup>1648</sup>

The following table lists each of the Parameter field substitution macros with a description of the result.

Field substitution macros are named according to Camel casing.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description
paramClassifierGUID	The unique GUID for the classifier of the current parameter.
paramDefault	Operation Parameters dialog: <b>Default</b> .
paramFixed	Operation Parameters dialog: <b>Fixed</b> checkbox.
paramGUID	The unique GUID for the current parameter.
paramIsEnum	<b>True</b> , if the parameter uses the <i>enum</i> keyword (C++).
paramKind	Operation Parameters dialog: <b>Kind</b> .
paramName	Operation Parameters dialog: <b>Name</b> .
paramNotes	Operation Parameters dialog: <b>Notes</b> .



Macro Name	Description
paramQualType	The parameter type qualified by the namespace path (if generating namespaces) and the classifier path (dot delimited). If the parameter classifier has not been set, is equivalent to the <i>paramType</i> macro.
paramType	Operation Parameters dialog: <b>Type</b> .

The following table lists each of the Problem field substitution macros with a description of the result.

Field substitution macros are named according to Camel casing.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description
problemCompletedBy	Maintenance dialog, Element Issues tab: <b>Completed by</b> .
problemCompletedDate	Maintenance dialog, Element Issues tab: <b>Completed</b> .
problemHistory	Maintenance dialog, Element Issues tab: <b>History</b> .
problemName	Maintenance dialog, Element Issues tab: <b>Name</b> .
problemNotes	Maintenance dialog, Element Issues tab: <b>Description</b> .
problemPriority	Maintenance dialog, Element Issues tab: <b>Priority</b> .
problemRaisedBy	Maintenance dialog, Element Issues tab: <b>Raised by</b> .
problemRaisedDate	Maintenance dialog, Element Issues tab: <b>Raised</b> .
problemStatus	Maintenance dialog, Element Issues tab: <b>Status</b> .
problemVersion	Maintenance dialog, Element Issues tab: <b>Version</b> .

The following table lists each of the Requirement field substitution macros with a description of the result.

Field substitution macros are named according to Camel casing.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description
requirementDifficulty	Properties dialog: Require tab: <b>Difficulty</b> .
requirementLastUpdated	Properties dialog: Require tab: <b>Last Update</b> .
requirementName	Properties dialog: Require tab: <b>Short Description</b> .
requirementNotes	Properties dialog: Require tab: <b>Notes</b> .
requirementPriority	Properties dialog: Require tab: <b>Priority</b> .
requirementStatus	Properties dialog: Require tab: <b>Status</b> .
requirementType	Properties dialog: Require tab: <b>Type</b> .

The following table lists each of the Resource field substitution macros with a description of the result.

Field substitution macros are named according to Camel casing.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description
resourceAllocatedTime	Project Management window, Resource Allocation tab: <b>Allocated Time</b> .
resourceEndDate	Project Management window, Resource Allocation tab: <b>End Date</b> .
resourceExpectedTime	Project Management window, Resource Allocation tab: <b>Expected Time</b> .
resourceExpendedTime	Project Management window, Resource Allocation tab: <b>Time Expended</b> .
resourceHistory	Project Management window, Resource Allocation tab: <b>History</b> .
resourceName	Project Management window, Resource Allocation tab: <b>Resource</b> .
resourceNotes	Project Management window, Resource Allocation tab: <b>Description</b> .
resourcePercentCompleted	Project Management window, Resource Allocation tab: <b>Completed(%)</b> .

Macro Name	Description
resourceRole	Project Management window, Resource Allocation tab: <b>Role</b> .
resourceStartDate	Project Management window, Resource Allocation tab: <b>Start Date</b> .

The following table lists each of the Risk field substitution macros with a description of the result.

Field substitution macros are named according to Camel casing.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description
riskName	Project Management window, Risks tab: <b>Risk</b> .
riskNotes	Project Management window, Risks tab: (Notes).
riskType	Project Management window, Risks tab: <b>Type</b> .
riskWeight	Project Management window, Risks tab: <b>Weight</b> .

The following table lists each of the Scenario field substitution macros with a description of the result.

Field substitution macros are named according to Camel casing.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description
scenarioGUID	The unique ID for a scenario. Identifies the scenario unambiguously within a model.
scenarioName	Properties dialog, Scenario tab: <b>Scenario</b> .
scenarioNotes	Properties dialog, Scenario tab: (Notes).
scenarioType	Properties dialog, Scenario tab: <b>Type</b> .

*Tagged Value* macros are a special form of field substitution macros, which provide access to element tags and the corresponding Tagged Values. They can be used in one of two ways:

- Direct Substitution
- Conditional Substitution

#### **Direct Substitution:**

This form of the macro directly substitutes the value of the named tag into the output.

Structure: %<macroName>: " <tagName> " %

<macroName> can be one of:

- attTag
- classTag
- connectorDestElemTag
- connectorDestTag
- connectorSourceElemTag
- connectorSourceTag
- connectorTag
- linkAttTag
- linkTag
- opTag
- packageTag
- paramTag

This corresponds to the tags for attributes, Classes, operations, packages, parameters, connectors with both ends, elements at both ends of connectors and connectors including the attribute end.

<tagName> is a string representing the specific tag name.

#### **Example**

```
%opTag: " attribute " %
```

#### **Conditional Substitution:**

This form of the macro mimics the conditional substitution defined for field substitution macros.

Structure: %<macroName>: " <tagName> " ( == " <test> " ) ? <subTrue> ( : <subFalse> ) %

Where:

- <macroName> and <tagName> are as defined above
- ( <text> ) denotes that <text> is optional
- <test> is a string representing a possible value for the macro
- <sub True> and <sub False> can be a combination of quoted strings and the keyword value; where the value is used, it gets replaced with the macro's value in the output

### Examples

```
%opTag: "opI n l i n e" ? "i n l i n e" : "" %
%opTag: "opI n l i n e" ? "i n l i n e" %
%cl assTag: "unsaf e" == "t r u e" ? "unsaf e" : "" %
%cl assTag: "unsaf e" == "t r u e" ? "unsaf e" %
```

Tagged Value macros use the same naming convention as field substitution macros.

### Learn more

- [Field Substitution Macros](#) 

If you want to provide access in a transformation template to data concerning the transformation of a Template Binding connector's binding parameter substitution in the model, you can use the **Template Parameter substitution macros**. The macro names are in Camel casing. Macros that represent checkboxes return a value of **T** if the box is selected; otherwise the value is empty.

### Template Parameter substitution macros

Macro Name	Description
parameterSubstitutionFormal	Template Binding Properties dialog, Binding Parameter tab, Parameter Substitution(s) panel: <b>Formal</b> Template Parameter name.
parameterSubstitutionActual	Template Binding Properties dialog, Binding Parameter tab, Parameter Substitution(s) panel: <b>Actual</b> parameter name/expression.
parameterSubstitutionActual Classifier	Template Binding Properties dialog, Binding Parameter tab, Parameter Substitution(s) panel: <b>Actual</b> parameter classifier.

### Learn more

- [Macros](#) 
- [Field Substitution Macros](#) 
- [Template Binding](#) 
- [Parameter Substitution](#) 
- The *OMG UML Specification (UML Superstructure Specification, v 2.4.1)* section 17.4.5

The following table lists each of the Test field substitution macros with a description of the result.

Field substitution macros are named according to Camel casing.

Macros that represent checkboxes return a value of **T** if the box is selected. Otherwise the value is empty.

Macro Name	Description
testAcceptanceCriteria	Testing window: <b>Acceptance Criteria</b> .
testCheckedBy	Testing window: <b>Checked By</b> .
testDateRun	Testing window: <b>Last Run</b> .
testClass	The Testing window tab (the type of test defined): Unit, Integration, System, Acceptance, Scenario.
testInput	Testing window: <b>Input</b> .
testName	Testing window: <b>Test</b> .
testNotes	Testing window: <b>Description</b> .
testResults	Testing window: <b>Results</b> .
testRunBy	Testing window: <b>Run By</b> .
testStatus	Testing window: <b>Status</b> .
testType	Testing window: <b>Type</b> .

#### Learn more

- [Field Substitution Macros](#) <sup>1648</sup>

#### 7.3.7.3.3 Function Macros

Function macros are a convenient way of manipulating and formatting various element data. Each function macro returns a result string. There are two primary ways to use the results of function macros:

- Direct substitution of the returned string into the output, such as: `%TO_LOWER(attName)%`
- Storing the returned string as part of a variable definition such as: `$name = %TO_LOWER(attName)%`

Function macros can take parameters, which can be passed to the macros as:

- String literals, enclosed within double quotation marks
- Direct substitution macros without the enclosing percent signs

- Variable references
- Numeric literals

Multiple parameters are passed using a comma-separated list.

Function macros are named according to the All-Caps style, as in: **%CONVERT\_SCOPE(opScope)%**

The available function macros are described below. Parameters are denoted by angle brackets, as in: **FUNCTION\_NAME(<param>)**.

**CONVERT\_SCOPE(<umlScope>):**

For use with supported languages, to convert <umlScope> to the appropriate scope keyword for the language being generated. The following table shows the conversion of <umlScope> with respect to the given language.

Language	Package	Public	Private	Protected
C++	public	public	private	protected
C#	internal	public	private	protected
Delphi	protected	public	private	protected
Java		public	private	protected
PHP	public	public	private	protected
VB	Protected	Public	Private	Protected
VB .Net	Friend	Public	Private	Protected

**COLLECTION\_CLASS(<language>):**

Gives the appropriate collection Class for the language specified for the current linked attribute.

**CSTYLE\_COMMENT(<wrap length>):**

Converts the notes for the element currently in scope to plain C-style comments, using /\* and \*/.

**DELPHI\_PROPERTIES(<scope>, <separator>, <indent>):**

Generates a Delphi property.

**DELPHI COMMENT(<wrap\_length>):**

Converts the notes for the element currently in scope to Delphi comments.

**EXEC\_ADD\_IN(<addin\_name>, <function\_name>, <prm\_1>, ..., <prm\_n>):**

Invokes an Enterprise Architect Add-In function, which can return a result string.

<addin\_name> and <function\_name> specify the names of the Add-In and function to be invoked.

Parameters to the Add-In function can be specified via parameters <prm\_1> to <prm\_n>.

```
$result = %EXEC_ADD_IN( " MyAddIn" , " ProcessOperation" , classGUID, opGUID) %
```

Any function that is to be called by the *EXEC\_ADD\_IN* macro must have two parameters: an *EA.Repository* object, and a *Variant* array that contains any additional parameters from the *EXEC\_ADD\_IN* call. Return type should be *Variant*.

```
Public Function ProcessOperation( Repository As EA.Repository, args As Variant ) As Variant
```

**FIND(<src>, <subString>):**

Position of the first instance of <subString> in <src>; -1 if none.

**GET\_ALIGNMENT():**

Returns a string where all of the text on the current line of output is converted into spaces and tabs.

**JAVADOC COMMENT(<wrap\_length>):**

Converts the notes for the element currently in scope to *javadoc*-style comments.

**LEFT(<src>, <count>):**

The first <count> characters of <src>.

**LENGTH(<src>):**

Length of <src>. Returns a string.

**MID(<src>, <start>)****MID(<src>, <start>, <count>):**

Substring of <src> starting at <start> and including <count> characters. Where <count> is omitted the rest of the string is included.

**PI(<option>, <value>, {<option>, <value>}):**

Sets the PI for the current template to <value>. Valid <values> are:



- `"\n"`
- `"\t"`
- `"\"`
- `"\"`

`<option>` controls when the new PI takes effect. Valid values for `<option>` are:

- *I, Immediate*: the new PI is generated before the next non-empty template line
- *N, Next*: the new PI is generated after the next non-empty template line

Multiple pairs of options are allowed in one call. An example of the situation where this would be used is where one keyword is always on a new line, as illustrated below:

```
%PI = " " %
%classAbstract ? "abstract" %
%i f classTag: "macro" != " " %
%PI ( "I", "\n", "N", " ") %
%classTag: "macro" %
%endif %
class
%className%
```

For more details, see the [description of PI](#)<sup>[1687]</sup>.

#### **PROCESS END OBJECT(<template name>):**

Enables the Classes that are one Class further away from the base Class, to be transformed into objects (such as attributes, operations, packages, parameters and columns) of the base Class. `<template_name>` refers to the working template that temporarily stores the data.

#### **REMOVE DUPLICATES(<source>, <separator>):**

Where `<source>` is a `<separator>` separated list; this removes any duplicate or empty strings.

#### **REPLACE(<string>, <old>, <new>):**

Replaces all occurrences of `<old>` with `<new>` in the given string `<string>`.

#### **RESOLVE OP NAME():**

Resolves clashes in interface names where two method-from interfaces have the same name.

#### **RESOLVE QUALIFIED TYPE()**

#### **RESOLVE QUALIFIED TYPE(<separator>)**

#### **RESOLVE QUALIFIED TYPE(<separator>, <default>):**

Generates a qualified type for the current attribute, linked attribute, linked parent, operation, or parameter. Enables the specification of a separator other than `.` and a default value for when some value is required.

#### **RIGHT(<src>, <count>):**

The last *<count>* characters of *<src>*.

**TO LOWER(<string>):**

Converts *<string>* to lower case.

**TO UPPER(<string>):**

Converts *<string>* to upper case.

**TRIM(<string>)**

**TRIM(<string>, <trimChars>):**

Removes trailing and leading white spaces from *<string>*. If *<trimChars>* is specified, all leading and trailing characters in the set of *<trimChars>* are removed.

**TRIM LEFT(<string>)**

**TRIM LEFT(<string>, <trimChars>):**

Removes the specified leading characters from *<string>*.

**TRIM RIGHT(<string>)**

**TRIM RIGHT(<string>, <trimChars>):**

Removes the specified trailing characters from *<string>*.

**VB COMMENT(<wrap length>):**

Converts the notes for the element currently in scope to Visual Basic style comments.

**WRAP COMMENT(<comment>, <wrap length>, <indent>, <start string>):**

Wraps the text *<comment>* at width *<wrap length>* putting *<indent>* and *<start string>* at the beginning of each line.

```
$behavior = %WRAP_COMMENT( opBehavior, "40", " ", " / " ) %
```

*<wrap length>* must still be passed as a string, even though **WRAP\_COMMENT** treats this parameter as an integer.

**WRAP LINES(<text>, <wrap length>, <start string>(, <end string> )):**

Wraps *<text>* as designated to be *<wrap length>*, adding *<start string>* to the beginning of every line and *<end string>* to the end of the line if it is specified.

**XML COMMENT(<wrap length>):**

Converts the notes for the element currently in scope to XML-style comments.

#### 7.3.7.3.3.4 Control Macros

Control macros are used to control the processing and formatting of the templates. The basic types of control macro include:

- The *list* macro, for generating multiple element features, such as attributes and operations
- The branching macros, which form *if-then-else* constructs to conditionally execute parts of a template
- The PI macro for formatting new lines in the output, which takes effect from the next non-empty line
- A PI function macro that enables setting PI to a variable and adds the ability to set the PI that is generated before the next line
- The synchronization macros

In general, control macros are named according to Camel casing.

#### Learn more

- [List Macro](#) <sup>[1683]</sup>
- [Function Macros](#) <sup>[1678]</sup>
- [Branching Macros](#) <sup>[1684]</sup>
- [Synchronization Macros](#) <sup>[1686]</sup>
- [The PI Macro](#) <sup>[1687]</sup>

If you need to loop or iterate through a set of Objects that are contained within or are under the current object, you can do so using the %list macro. This macro performs an iterative pass on all the objects in the scope of the current template, and calls another template to process each one.

The basic structure is:

```
%list=<TemplateName> @separator=<string> @ndent=<string> ( <conditions>
) %
```

where <string> is a double-quoted literal string and <TemplateName> can be one of the following template names:

- Attribute
- AttributeImpl
- Class
- ClassBase
- ClassImpl
- ClassInterface
- Constraint
- Custom Template (custom templates enable you to define your own templates)
- Effort
- InnerClass
- InnerClassImpl
- LinkedFile
- Metric
- Namespace
- Operation

- OperationImpl
- Parameter
- Problem
- Requirement
- Resource
- Risk
- Scenario
- Test

`<conditions>` is optional and looks the same as the conditions for *if* and *elseif* statements.

### Example

In a Class transform, the Class might contain multiple Attributes; the following example calls the Attribute transform and outputs the result of processing the transform for each attribute of the Class in scope. The resultant list separates its items with a single new line and indents them two spaces respectively. If the Class in scope had any *stereotyped* attributes, they would be generated using the appropriately specialized template.

```
%list="Attribute" @separator="\n" @indent="  "%
```

The *separator* attribute, denoted above by `@separator`, specifies the space that should be used between the list items, excluding the last item in the list.

The *indent* attribute, denoted by `@indent`, specifies the space by which each line in the generated output should be indented.

### Special Cases

There are some special cases to consider when using the `%list` macro:

- If the *Attribute* template is used as an argument to the `%list` macro, this also generates attributes derived from Associations by executing the appropriate *LinkedAttribute* template
- If the *ClassBase* template is used as an argument to the `%list` macro, this also generates Class bases derived from links in the model by executing the appropriate *LinkedClassBase* template
- If the *ClassInterface* template is used as an argument to the `%list` macro, this also generates Class bases derived from links in the model by executing the appropriate *LinkedClassInterface* template
- If *InnerClass* or *InnerClassImpl* is used as an argument to the `%list` macro, these Classes are generated using the *Class* and *ClassImpl* templates respectively; these arguments direct that the templates should be processed based on the inner Classes of the Class in scope

### Learn more

- [Create Custom Templates](#) <sup>[1701]</sup>
- [Customize Base Templates](#) <sup>[1702]</sup>
- [Control Macros](#) <sup>[1683]</sup>

Branching macros provide if-then-else constructs. The CTF supports a limited form of branching through the following macros:

- *if*

- *elseif*
- *else*
- *endif*
- *endTemplate* (which exits the current template)

The basic structure of the *if* and *elseif* macros is:

```
%i f <test> <operator> <test>%
```

where *<operator>* can be one of:

- ==
- !=

and *<test>* can be one of:

- a string literal, enclosed within double quotation marks
- a direct substitution macro, without the enclosing percent signs
- a variable reference

Branches can be nested, and multiple conditions can be specified using one of:

- *and* or
- *or*

When specifying multiple conditions, *and* and *or* have the same order of precedence, and conditions are processed left to right.

If conditional statements on strings are case sensitive, "a String" does not equal "A STRING". Hence in some situations it is better to set the variable *\$str= TO\_LOWER(variable)* or *TO\_UPPER(variable)* then compare to a specific case.

Macros are not supported in the conditional statements. It is best to assign the results of a macro (string) to a variable, and then use the variable in the comparison.

```
$fldType = %TO_LOWER($parameter1)%
$COMMENT = "Use the first 4 characters for Date and Time field types"
$fldType4 = %LEFT($fldType, 4)%
%i f $fldType4=="date"%
Datetime
%endif%
```

This takes a parameter of value "Datetime", "DATETIME" or "Date", and returns "Datetime".

The *endif* or *endTemplate* macros must be used to signify the end of a branch. In addition, the *endTemplate* macro causes the template to return immediately, if the corresponding branch is being executed.

#### **Example 1:**

```
%i f elementType == "Interface" %
;
%else%
%OperationBody%
%endif%
```

In this case:

- If the *elemType* is "Interface" a semi-colon is returned
- If the *elemType* is not "Interface", a template called Operation Body is called

### Example 2:

```
$bases=" Cl assBase"
$I n t e r f a c e s="" %
%if $bases != "" and $I n t e r f a c e s != "" %
: $bases, $I n t e r f a c e s
%el self $bases != "" %
: $bases
%el self $I n t e r f a c e s != "" %
: $I n t e r f a c e s
%endif %
```

In this case the text returned is : Cl assBase.

### Conditions using Boolean Value:

When doing branching using conditions that involve a system checkbox (boolean fields), such as Attribute.Static (*attStatic*) the conditional statement would be written as:

```
%if attStatic == "T" %
```

For example:

```
%if attCollection=="T" or attOrderedMultiplicity == "T"%
%endTemplate%
```

### Learn more

- [Attribute Field Substitution Macros](#) <sup>[1652]</sup>

The *synchronization macros* are used to provide formatting hints to Enterprise Architect when inserting new sections into the source code, during forward synchronization. The values for synchronization macros must be set in the **File** templates.

The structure for setting synchronization macros is:

```
%<name>=<value>%
```

where <name> can be one of the macros listed below and <value> is a literal string enclosed by double quotes.

Macro Name	Description
<b>synchNewClassNotesSpace</b>	Space to append to a new Class note. Default value: \ n.
<b>synchNewAttributeNotesSpace</b>	Space to append to a new attribute note. Default value: \ n.
<b>synchNewOperationNotesSpace</b>	Space to append to a new operation note. Default value: \ n.

Macro Name	Description
<b>synchNewOperationBodySpace</b>	Space to append to a new operation body. Default value: \ n .
<b>synchNamespaceBodyIndent</b>	Indent applied to Classes within non-global namespaces. Default value: \ t .

The PI (Processing Instruction) macro provides a means of defining the separator text to be inserted between the code pieces (which represent entities) that are generated using a template.

The structure for setting the Processing Instruction is:

```
%PI =<value>%
```

where <value> is a literal string enclosed by double quotes, with the following options:

- " \ n " - New line (the default)
- " " - Space
- " \ t " - Tab
- " " - Null

By default, the PI is set to generate a new line (\ n) for each non-empty substitution, which behavior can be changed by resetting the *PI* macro. For instance, a Class's Attribute declaration in simple VB code would be generated to a single line statement (with no new lines). These properties are derived from the Class-Attribute Properties in the model to generate, for example:

```
Private Const PrintFormat As String = "Portrait"
```

The template for generating this starts with the PI being set to a space rather than a new line:

```
%PI=" "%
%CONVERT_SCOPE(attScope)%
%endIf%
%i f attConst == "T"%
Const
%endIf%
```

On transforming this, *attScope* returns the VB keyword "Private" and *attConst* returns "Const" on the same line spaced by a single space (fitting the VB Class.Attribute definition example above).

Alternatively, when generating a Class you might want the Class declaration, the notes and Class body all separated by double lines. In this case the %PI is set to "/ n/ n" to return double line spacing:

```
%PI="\n\n"%
%ClassDeclaration%
%ClassNotes%
%ClassBody%
```

#### **PI Characteristics:**

- Blank lines have no effect on the output

- Any line that has a macro that produces an empty result does not result in a PI separator (space/new line)
- The last entry does not return a PI; for example, %Classbody% (above) does not have a double line added after the body

#### Learn more

- [Function Macros](#)

#### 7.3.7.3.3.5 EASL Code Generation Macros

Enterprise Architect provides two Enterprise Architect Simulation Library (EASL) code generation macros to generate code from behavioral models. These are:

- *EASL\_GET* and
- *EASLList*

#### EASL\_GET

The *EASL\_GET* macro is used to retrieve a property or a collection of an EASL object. The EASL objects and the properties and collections for each object are identified in the [EASL Collections](#) and [EASL Properties](#) topics.

#### Syntax

```
$result = %EASL_GET( <<Property>>, <<Owner ID>>, <<Name>> )
```

where:

- <<Property>> is either "Property" or "Collection"
- <<OwnerID>> is the ID of the owner object for which the property/collection is to be retrieved
- <<Name>> is the name of the property or Collection being accessed
- \$result is the returned value; this is "" if not a valid property

#### Example

```
$sPropName = %EASL_GET( " Property", $context, " Name" ) %
```

#### EASLList

The *EASLList* macro is used to render each object in an EASL collection using the appropriate template.

#### Syntax

```
$result = %EASLList=<<TemplateName>> @separator=<<Separator>>
          @indent=<<indent>> @owner=<<OwnedID>>
          @collection=<<CollectionName>> @option1=<<OPTI ON1>>
          @option2=<<OPTI ON2>>..... @optionN=<<OPTI ONN>>%
```



where:

- <<TemplateName>> is the name of any [behavioral model template](#)<sup>[1689]</sup> or [custom template](#)<sup>[1701]</sup>
- <<Separator>> is a list separator (such as “\n”)
- <<indent>> is any indentation to be applied to the result
- <<owner>> is the ID of the object that contains the required collection
- <<CollectionName>> is the name of the required collection
- <<OPTION1>....<<OPTION99>> are miscellaneous options that might be passed on the template; each option is given as an additional input parameter to the template
- \$result is the resultant value; this is “” if not a valid collection

### **Example**

```
$sStates = %EASLList="State" @separator="\n" @indent="\t"
           @owner=$StateMachineGUI D @collection="States"
@option=$sOption%
```

### **Behavioral Model Templates**

- Action
- Action Assignment
- Action Break
- Action Call
- Action Create
- Action Destroy
- Action If
- Action Loop
- Action Opaque
- Action Parallel
- Action RaiseEvent
- Action RaiseException
- Action Switch
- Behavior
- Behavior Body
- Behavior Declaration
- Behavior Parameter
- Call Argument
- Decision Action
- Decision Condition
- Decision Logic
- Decision Table
- Guard
- Property Declaration
- Property Notes

- Property Object
- State
- State CallBack
- State Enumerate
- State EnumeratedName
- StateMachine
- StateMachine HistoryVar
- Transition
- Transition Effect
- Trigger

This topic lists the EASL collections for each of the EASL objects, as retrieved by the [EASL\\_GET](#)<sup>[1688]</sup> code generation macro.

**Action:**

Collection Name	Description
<b>Arguments</b>	The Action's arguments.
<b>SubActions</b>	The sub-actions of the Action.

**Behavior:**

Collection Name	Description
<b>Actions</b>	The Behavior's Actions.
<b>Nodes</b>	The Behavior's nodes.
<b>Parameters</b>	The Behavior's parameters.
<b>Variables</b>	The Behavior's variables.

**Classifier:**

Collection Name	Description
<b>AllStateMachines</b>	All State Machines for the Classifier.

Collection Name	Description
<b>AsynchProperties</b>	The asynchronous properties of the Classifier.
<b>AsynchTriggers</b>	The asynchronous triggers of the Classifier.
<b>Behaviors</b>	The behaviors of the Classifier.
<b>Properties</b>	The properties of the Classifier.
<b>TimedProperties</b>	The timed properties of the Classifier.
<b>TimedTriggers</b>	The timed triggers of the Classifier.
<b>Triggers</b>	All triggers of the Classifier.

**Construct:**

Collection Name	Description
<b>AllChildren</b>	The Construct's children.
<b>ClientDependencies</b>	The client dependencies on the Construct.
<b>StereoTypes</b>	The stereotypes of the Construct.
<b>SupplierDependencies</b>	The supplier dependencies on the Construct.

**Node:**

Collection Name	Description
<b>IncomingEdges</b>	The Node's incoming edges.
<b>OutgoingEdges</b>	The Node's outgoing edges.
<b>SubNodes</b>	The sub-nodes of the Node.

Collection Name	Description

**State:**

Collection Name	Description
<b>DoBehaviors</b>	The State's Do behaviors.
<b>EntryBehaviors</b>	The State's Entry behaviors.
<b>ExitBehaviors</b>	The State's Exit behaviors.

**StateMachine:**

Collection Name	Description
<b>AllFinalStates</b>	The State Machine's final States.
<b>AllStates</b>	All States within the State Machine, including those within Submachine States.
<b>DerivedTransitions</b>	The State Machine's derived transitions with the associated valid effect.
<b>States</b>	The States within the State Machine.
<b>Transitions</b>	The transitions within the State Machine.
<b>Vertices</b>	The State Machine's vertices.

**Transition:**

Collection Name	Description
<b>Effects</b>	The Transition's effects.
<b>Guards</b>	The Transition's guards.

Collection Name	Description
<b>Triggers</b>	The Transition's triggers.

**Trigger:**

Collection Name	Description
<b>TriggeredTransitions</b>	The triggered transitions associated with the Trigger.

**Vertex:**

Collection Name	Description
<b>DerivedOutgoingTransitions</b>	The Vertex's derived outgoing transitions after traversing the pseudo-nodes.
<b>IncomingTransitions</b>	The Vertex's incoming transitions.
<b>OutgoingTransitions</b>	The Vertex's outgoing transitions.

This topic lists the EASL properties for each of the EASL objects, as retrieved by the [EASL\\_GET](#)<sup>[1688]</sup> code generation macro.

**Action**

Property Name	Description
<b>Behavior</b>	The Action's associated behavior ( <i>Call Behavior Action</i> or <i>Call Operation Action</i> ).
<b>Body</b>	The Action's body.
<b>Context</b>	The Action's context.
<b>Guard</b>	The Action's guard.
<b>IsFinal</b>	A check on whether the action is a final Action.

Property Name	Description
<b>IsGuarded</b>	A check on whether the action is a guarded Action.
<b>IsInitial</b>	A check on whether the action is an initial Action.
<b>Kind</b>	The Action's kind.
<b>Next</b>	The Action's next action.
<b>Node</b>	The Action's associated node in the graph.

#### Argument

Property Name	Description
<b>Parameter</b>	The ID of the Argument's associated parameter.
<b>Value</b>	The default value of the argument.

#### Behavior

Property Name	Description
<b>InitialAction</b>	The Behavior's initial action.
<b>isReadOnly</b>	The isReadOnly of the Behavior.
<b>isSingleExecution</b>	The isSingleExecution of the Behavior.
<b>Kind</b>	The kind of Behavior.
<b>ReturnType</b>	The return type of the Behavior.
<b>Specification</b>	The specification of the Behavior.

**CallEvent**

Property Name	Description
<b>Operation</b>	The operation of the CallEvent.

**ChangeEvent**

Property Name	Description
<b>ChangeExpression</b>	The change expression of the ChangeEvent.

**Classifier**

Property Name	Description
<b>HasBehaviors</b>	A check on whether the Classifier has behavioral models (Activity and Interaction).
<b>Language</b>	The Classifier's language.
<b>StateMachine</b>	The State Machine of the Classifier.

**Condition**

Property Name	Description
<b>Expression</b>	The Condition's expression.
<b>Lower</b>	The Condition's lower value.
<b>Upper</b>	The Condition's upper value.

**Construct**

Property Name	Description
<b>GetTaggedValue</b>	The Property's Tagged Value.

Property Name	Description
<b>IsStereotypeApplied</b>	A check on whether a particular stereotype is applied to the Property.
<b>Notes</b>	Notes on the Property.
<b>UMLType</b>	The UML type of the Property.
<b>Visibility</b>	The visibility of the Property.

#### Edge

Property Name	Description
<b>From</b>	The ID of the node from which the Edge arises.
<b>To</b>	The ID of the node at which the Edge is targeted.

#### EventObject

Property Name	Description
<b>EventKind</b>	The event kind of the Event Object.

#### Instance

Property Name	Description
<b>Classifier</b>	The classifier of the Instance.
<b>Value</b>	The value of the Instance.

#### Parameter



Property Name	Description
<b>Direction</b>	The direction of the Parameter.
<b>Type</b>	The type of the Parameter.
<b>Value</b>	The value of the parameter.

### Primitive

Property Name	Description
<b>FQName</b>	The FQ name of the Primitive.
<b>ID</b>	The ID of the Primitive.
<b>Name</b>	The name of the Primitive.
<b>ObjectType</b>	The object type of the Primitive.
<b>Parent</b>	The IDParent of the Primitive.

### PropertyObject

Property Name	Description
<b>BoundSize</b>	The bound size of the PropertyObject (if it is a collection).
<b>ClassifierStereoType</b>	The stereotype of the PropertyObject's classifier.
<b>IsAsynchProp</b>	A check on whether the PropertyObject is an asynchronous property.
<b>IsCollection</b>	A check on whether the PropertyObject is a collection.
<b>IsOrdered</b>	A check on whether the PropertyObject is ordered (if it is a collection).
<b>IsTimedProp</b>	A check on whether the PropertyObject is a timed property.

Property Name	Description
<b>Kind</b>	The PropertyObject's kind.
<b>LowerValue</b>	The PropertyObject's lower value (if it is a collection).
<b>Type</b>	The PropertyObject's type.
<b>UpperValue</b>	The PropertyObject's upper value (if it is a collection).
<b>Value</b>	The PropertyObject's value.

#### SignalEvent

Property Name	Description
<b>Signal</b>	The signal of the SignalEvent.

#### State

Property Name	Description
<b>HasSubMachine</b>	A check on whether the State is a Submachine state.
<b>IsFinalState</b>	A check on whether the State is a final state.
<b>SubMachine</b>	Get the ID of the Submachine contained by the State (if applicable).

#### StateMachine

Property Name	Description
<b>HasSubMachineState</b>	A check on whether the State Machine has a Submachine state.
<b>InitialState</b>	The State Machine's initial state.

Property Name	Description
<b>SubMachineState</b>	The State Machine's Submachine state.

#### TimeEvent

Property Name	Description
<b>When</b>	The 'when' property of the TimeEvent.

#### Transition

Property Name	Description
<b>HasEffect</b>	A check on whether the transition has a valid effect.
<b>IsDerived</b>	A check on whether the transition is a derived transition.
<b>IsTranscend</b>	A check on whether the transition transcends from one State Machine (Submachine state) to another.
<b>IsTriggered</b>	A check on whether the transition is triggered.
<b>Source</b>	The Transition's source.
<b>Target</b>	The Transition's target.

#### Trigger

Property Name	Description
<b>AsynchDestinationState</b>	The asynchronous destination state of the Trigger (if it is an asynchronous trigger).
<b>DependentProperty</b>	The ID of the property associated with the Trigger.
<b>Event</b>	The Trigger's event.

Property Name	Description
<b>Name</b>	The Trigger's name.
<b>Type</b>	The Trigger's type.

### Vertex

Property Name	Description
<b>IsHistory</b>	A check on whether the vertex is a history state.
<b>IsPseudoState</b>	A check on whether the vertex is a pseudo state.
<b>PseudoStateKind</b>	The Vertex's pseudo-state kind.

#### 7.3.7.3.4 Call Templates From Templates

Using function calls with parameters, you can call templates from other templates, whether standard templates or user-defined templates created within your project. Also, called templates can return a value, and can be called recursively.

### Examples

A call statement returning a parameter to a variable:

```
$sSource = %StateEnumeratedName($Source)%
```

A call statement to a template that has parameters:

```
%RuleTask($GUID, $index)%
```

Using the \$parameter statement in the called template:

```
$GUID = $parameter1
$index = $parameter2
```

Templates support recursive calls, such as the following recursive call on the template RuleTask:

```
$GUID = $parameter1
$index = $parameter2

%PI = ""%

$null = "Initialize condition and action object"

$count = %BR_GET("RuleCount")%
```

```

%if $count==" " or $count == $index%
%ComputeRule($GUID)%
\n
%endTemplate%

%Rule($index)%
\n
$index = %MATH_ADD($index, "1")%
%RuleTask($GUID, $index)%

```

#### Learn more

- [Create Custom Templates](#)<sup>[1701]</sup>
- [Customize Base Templates](#)<sup>[1702]</sup>
- [The Code Template Editor in MDG Development](#)<sup>[1701]</sup>

### 7.3.7.4 The Code Template Editor in MDG Development

The following topics describe how you use the Code Template Editor window to create custom templates:

- [Create Custom Templates](#)<sup>[1701]</sup>
- [Customize Base Templates](#)<sup>[1702]</sup>
- [Add New Stereotyped Templates](#)<sup>[1703]</sup>

The Code Template Editor provides the facilities of the **Common Code Editor**, including Intelli-sense for the code generation template macros. For more information on Intelli-sense and the Common Code Editor, see the *Editing Source Code* topic.

#### Learn more

- [Code Template Framework](#)<sup>[1631]</sup>
- [Code Generation Template Macros](#)<sup>[1646]</sup>
- [Editing Source Code](#)<sup>[2146]</sup>

#### 7.3.7.4.1 Create Custom Templates

Enterprise Architect provides a wide range of templates that define how code elements are generated. If these are not sufficient for your purposes - for example, if you want to generate code in a language not currently supported by Enterprise Architect - you can create completely new custom templates. You can also add **stereotype overrides** to your custom templates; for example, you might list all of your parameters and their notes in your method notes.

**Access** **Tools | Source Code Generation Templates (Ctrl+Shift+P)**

#### Create custom templates using the Code Templates Editor

Step	Description	See also
1	In the <b>Language</b> field, click on the drop-down arrow and select the appropriate programming language.	

2	Click on the <b>Add New Custom Template</b> button. The Create New Custom Template dialog displays.	
3	In the <b>Template Type</b> field, click on the drop-down arrow and select the appropriate modeling object.  The <b>&lt;None&gt;</b> option requires special treatment; it enables the definition of a function macro that doesn't actually apply to any of the types, but must be called as a function to define variables <i>\$parameter1</i> , <i>\$parameter2</i> and so on for each value passed in.	
4	In the <b>Template Name</b> field, type an appropriate name. Click on the <b>OK</b> button.	
5	On the Code Templates Editor tab, the new template is included in the Templates list, with the value <b>Yes</b> in the <b>Modified</b> field.  The template is called <i>&lt;Template Type&gt;__&lt;Template Name&gt;</i> .  Note the double underscore character between the template type and template name.	
6	Select the template from the Templates list and edit the contents in the <b>Template</b> field to meet your requirements.	<a href="#">Code Template Syntax</a> <sup>[1643]</sup>
7	Click on the <b>Save</b> button.  This stores the new template, which is now available from the list of templates for use. You can also add a stereotype override to the template, if you wish.	<a href="#">Add New Stereotyped Templates</a> <sup>[1703]</sup>

### Notes

- For a custom language, you must define the *File* template so that it can call the *Import Section*, *Namespace* and *Class* templates, and any other templates that you decide are applicable

### Learn more

- [Developing Programming Languages](#)<sup>[1581]</sup>

#### 7.3.7.4.2 Customize Base Templates

Enterprise Architect provides a wide range of templates that define how code elements are generated. If you want to change the way a code element is generated, you can customize the appropriate templates. Your changes might be to the effect of the template itself, or to its calls to other templates. You can also add **stereotype overrides** to your customized templates; for example, you might list all of your parameters and their notes in your method notes.

When you **customize** a **system-provided** (base) template, you effectively create a **copy** of the template that is used in preference to the original. All subsequent changes are to that copy, and the original base template is hidden. If you subsequently delete the copy it can no longer override the original, which is then brought into use again.

**Access** **Tools | Source Code Generation Templates** ( Ctrl+Shift+P )

#### Customize a base template

Step	Description	See also
1	On the Code Template Editor, in the <b>Language</b> field, click on the drop-down arrow and select the programming language for which you want to customize the base templates.	
2	In the Templates list, click on the base template to edit.	
3	Update the template.	<a href="#">Code Template Syntax</a> <sup>[1643]</sup>
4	Click on the <b>Save</b> button to store your changes.	
5	Repeat steps 2 to 4 for each of the relevant base templates you want to customize.	
6	If you wish, add one or more stereotype overrides to any of the templates.	<a href="#">Add New Stereotyped Templates</a> <sup>[1703]</sup>

#### **7.3.7.4.3 Add New Stereotyped Templates**

Sometimes it is useful to define a specific code generation template for use with elements of a given stereotype. This enables different code to be generated for elements, depending on their stereotype. Enterprise Architect provides some default templates, which have been specialized for commonly used stereotypes in supported languages. For example the Operation Body template for C# has been specialized for the *property* stereotype, so that it automatically generates its constituent *get* and *set* methods. Users can override the default stereotyped templates as described in the *Override Default Templates* topic. Additionally users can define templates for their own stereotypes, as described below.

**Access** **Tools | Source Code Generation Templates** ( Ctrl+Shift+P )

#### Add a new stereotyped template using the Code Template Editor

Step	Description	See also
1	Select the appropriate language, from the <b>Language</b> list.	

Step	Description	See also
2	Select one of the base templates, from the <b>Templates</b> list.	
3	Click on the <b>Add New Stereotyped Override</b> button. The New Template Override dialog displays.	
4	Select the required <b>Feature</b> and/or <b>Class</b> stereotype. Click on the <b>OK</b> button.	
5	The new stereotyped template override displays in <b>Stereotype Overrides</b> list, marked as modified.	
6	Make the required modifications in the Code Templates Editor.	
7	Click on the <b>Save</b> button to store the new stereotyped template in the project file. Enterprise Architect can now use the stereotyped template, when generating code for elements of that stereotype.	

#### Notes

- Class and feature stereotypes can be combined to provide a further level of specialization for features; for example, if properties should be generated differently when the Class has a stereotype *MyStereotype*, then both *property* and *MyStereotype* should be specified in the New Template Override dialog

#### 7.3.7.4.4 Override Default Templates

Enterprise Architect has a set of built-in or default code generation templates. The Code Templates Editor enables you to modify these default templates, hence customizing the way in which Enterprise Architect generates code. You can choose to modify any or all of the base templates to achieve your required coding style.

Any templates that you have overridden are stored in the project file. When generating code, Enterprise Architect first checks whether a template has been modified and if so, uses that template. Otherwise the appropriate default template is used.

**Access** **Tools | Source Code Generation Templates (Ctrl+Shift+P)**

#### Reference

Override a default code generation template using the Code Templates Editor, as indicated below:



Field/Button	Description	See also
<b>Language</b>	Select the appropriate language from the list.	
<b>Templates</b>	Select one of the base templates from the list.	
<b>Stereotype Overrides</b>	If the base template has stereotyped overrides, you can select one of these from the list.	
<b>&lt;Other fields&gt;</b>	Make any other modifications required.	
<b>Save</b>	Click on this button to store the modified version of the template to the project file. The template is marked as modified.	

When generating code, Enterprise Architect now uses the overriding template, instead of the default template.

### 7.3.8 Grammar Framework

Enterprise Architect provides reverse engineering support for a number of popular programming languages. However, if the language you are using is not supported, you can **write your own grammar** for it, using the in-built **Grammar Editor**. You can then incorporate the grammar into an MDG technology to provide both reverse engineering and code synchronization support for your target language.

The framework for writing a grammar and importing it into Enterprise Architect is the direct complement to the **Code Template Framework**. While code templates are for converting a model to a textual form, grammars are required to convert text to a model. Both are required to synchronize changes into your source files.

An example language source file and an example Grammar for that language are provided in the *Code Samples* directory, which you can access from your installation directory (the default location is *C:\Program Files\Sparx Systems\EA*). Two other grammar files are also provided, illustrating specific aspects of developing Grammars.

#### Components

Component	Description	See also
<b>Grammar Syntax</b>	<p>Grammars define how a text is to be broken up into a structure, which is necessary when you are converting code into a UML representation. At the simplest level, a grammar is instructions for breaking up an input to form a structure.</p> <p>Enterprise Architect uses a variation of Backus–Naur Form (nBNF) to include processing instructions, the execution of which returns structured information from the parsed results in the form of an Abstract Syntax Tree (AST), which is used to generate a UML representation.</p>	<a href="#">Grammar Syntax</a> <small>[1706]</small>

Component	Description	See also
<b>Grammar Editor</b>	The Grammar Editor is an in-built editor that you can use to open, edit, validate and save grammar files.	<a href="#">Editing Grammars</a> <sup>[1721]</sup>
<b>Grammar Debugging</b>	You can debug the grammar files you create using two facilities: <ul style="list-style-type: none"> <li>The <b>Parser</b>, which generates the AST for the Grammar</li> <li>The <b>Profiler</b>, which also parses the Grammar and generates the AST but which exposes the Profiling pathway to show exactly what happened at each step of the process</li> </ul>	<a href="#">Parsing AST Results</a> <sup>[1722]</sup> <a href="#">Profiling Grammar Parsing</a> <sup>[1722]</sup>

#### Learn more

- [Add Code Modules](#) <sup>[1552]</sup> (to an MDG Technology)
- [Example Grammars](#) <sup>[1723]</sup>

### 7.3.8.1 Grammar Syntax

Grammars define how a text is to be broken up into a structure, which is exactly what is needed when you are converting code into a UML representation. At the simplest level, a grammar is just instructions for breaking up an input to form a structure. Enterprise Architect uses a variation of **Backus–Naur Form (BNF)** to express a grammar in a way that allows it to convert the text to a UML representation. What the grammar from Enterprise Architect offers over a pure BNF is the addition of **processing instructions**, which allow structured information to be returned from the parsed results in the form of an **Abstract Syntax Tree (AST)**. At the completion of the AST, Enterprise Architect will process it to produce a UML model.

#### Syntax

Syntax	Detail	See also
<b>Comments</b>	Comments have the same form as in many programming languages. <pre>// You can comment to the end of a line by adding two /s. /* You can comment multiple lines by adding a / followed by a *. The comment is ended when you add a * followed by a /. */</pre>	
<b>Instructions</b>	Instructions specify the key details of how the grammar works. They are generally included at the top of the grammar, and resemble function calls in most programming languages.	<a href="#">Grammar Instructions</a> <sup>[1707]</sup>
<b>Rules</b>	Rules make up the body of a grammar. A rule may have one or more definitions separated by pipe delimiters ( ).  For a rule to pass, any single complete definition must pass. Rules are terminated with the semi-colon character (;).	<a href="#">Grammar Rules</a> <sup>[1708]</sup>

<b>Definitions</b>	A definition is one of the paths a rule can take. Each definition is made up of one or more <b>terms</b> .	
<b>Definition Lists</b>	<p>A definition list corresponds to one or more <b>sets of terms</b>. These will be evaluated in order until one succeeds. If none succeed then the containing rule fails. Each definition is separated by a   character.</p> <p>This is a simple rule with three definitions:</p> <pre>&lt;greeting&gt; ::= "hello"   "hi"   ["good"] "morning";</pre>	
<b>Terms</b>	A term can be a reference to a rule, a specific value, a range of values, a sub-rule or a command.	<a href="#">Grammar Terms</a> <small>[1709]</small>
<b>Commands</b>	<p>Like instructions, commands resemble function calls. They serve two main purposes:</p> <ul style="list-style-type: none"> <li>To process tokens in a specific way or</li> <li>To provide a result to the caller</li> </ul>	<a href="#">Grammar Commands</a> <small>[1709]</small> <a href="#">Context Labels</a> <small>[1711]</small>

Learn more

- [Grammar Framework](#) [1705]
- [Editing Grammars](#) [1721]
- [Parsing AST Results](#) [1722]
- [Profiling Grammar Parsing](#) [1722]

**7.3.8.1.1 Grammar Instructions**

**Instructions** specify the key details of how the grammar works. They are generally included at the top of the grammar, and resemble **function calls** in most programming languages.

Instructions

Instruction	Description
<b>caseSensitive()</b>	One of these two instructions is expected to specify if <b>token matching</b> needs to be case sensitive or not. For example, languages in the BASIC family are case insensitive while languages in the C family are case sensitive.
<b>caseInsensitive()</b>	
<b>delimiters (DelimiterRule: Expression)</b>	The delimiters instruction tells the <b>lexical analyzer</b> which rule to use for delimiter discovery. Delimiters are used during keyword analysis, and can be defined as the characters that may be used immediately before or after language keywords.
<b>lex(TokenRule: Expression)</b>	The <b>lex instruction</b> tells the lexical analyzer the name of the root rule to use for its analysis.

Instruction	Description
<b>parse(RootRule: Expression)</b> <b>parse(RootRule: Expression, SkipRule: Expression)</b>	The <b>parse instruction</b> tells the parser the name of the root rule to use for its processing. The optional second argument specifies a skip (or escape) rule which is generally used to handle comments.

#### Learn more

- [Grammar Syntax](#)<sup>[1706]</sup>

### 7.3.8.1.2 Grammar Rules

**Rules** are run to break up text into structure. A rule is made up of one or more **definitions**, each of which is made up of one or more **terms**.

#### Types of Rule

Rule	Description
<b>Named rules</b>	A name, followed by a definition list. For example: <code>&lt;rule&gt; ::= &lt;term1&gt; &lt;term2&gt;   "-" &lt;term1&gt;;</code>
<b>Inline Rules</b>	Inside a definition, a rule defined within parentheses. These act in exactly the same way as if they were a named rule being called by a term. For example: <code>&lt;rule&gt; ::= (&lt;inline&gt;);</code>
<b>Optional Rules</b>	Inside a definition, a rule defined within square brackets. This rule succeeds even if the contents fail. For example: <code>&lt;rule&gt; ::= [&lt;inline&gt;];</code>
<b>Repeating Rules</b>	Inside a definition, a term followed by a plus sign. This rule matches the inner rule once or more than once. For example: <code>&lt;rule&gt; ::= &lt;inline&gt;+;</code> <code>rule ::= (&lt;term1&gt; &lt;term2&gt;)+;</code>
<b>Optional Repeating Rules</b>	Inside a definition, a rule followed by a star. This rule matches the inner rule zero or more times, meaning it succeeds even if the inner rule never succeeds. For example: <code>&lt;rule&gt; ::= &lt;inline&gt;*;</code> <code>rule ::= (&lt;term1&gt; &lt;term2&gt;)*;</code>

Learn more

- [Grammar Terms](#)<sup>[1709]</sup>

### 7.3.8.1.3 Grammar Terms

Terms identify where tokens are consumed.

Types of Term

Type	Description
<b>Concrete terms</b>	Quoted strings. e.g. "class"
<b>Unicode characters</b>	A lexer-only term, having the prefix of U+0x followed by a hexadecimal number. e.g. U+0x1234
<b>Ranges</b>	A lexer-only term, matching any character between the two characters specified. e.g. "a".. "z"    or    U+0x1234..U+2345
<b>References</b>	The name of another rule, in angled brackets. The token will match if that rule succeeds. e.g. <anotherRule>
<b>Commands</b>	A call to a specific command.

Learn more

- [Grammar Syntax](#)<sup>[1706]</sup>

### 7.3.8.1.4 Grammar Commands

**Commands**, like Instructions, resemble function calls. They serve two main purposes:

- To process tokens in a specific way or
- To provide a result to the caller



Commands

Command	Description
<b>attribute(Name: String, Value:</b>	Creates an attribute on the current AST node. The attribute will be created with the <b>Name</b> specified in the grammar source, and will be given the value of all

Command	Description
<b>Expression)</b>	tokens consumed as a part of executing the <b>Value</b> expression.  This command produces the AST node attributes that Enterprise Architect operates on in code engineering.
<b>attributeEx(Name: String)</b> <b>attributeEx(Name: String , Value: String)</b>	Creates an attribute on the current AST node without consuming any tokens. The attribute will be created with the <b>Name</b> specified in the grammar source, and with either an empty value or the value specified by the optional <b>Value</b> argument.  This command produces the AST node attributes that Enterprise Architect operates on in code engineering.
<b>node(Name: String, Target: Expression)</b>	Creates an AST node under the current AST node (the nodes that Enterprise Architect operates on in code engineering). The node will be created with the <b>Name</b> specified in the grammar source.
<b>token(Target: Expression)</b>	Creates a token during lexical analysis for processing during parsing. The value of the token will be the value of all characters consumed as a result of executing the <b>Target</b> expression.
<b>keywords()</b>	Matches any keyword from the parser rules set for the language.
<b>skip(Target: Expression)</b> <b>skip(Target: Expression, Escape: Expression)</b>	Consumes input data (characters when lexing, and tokens when parsing) until the <b>Target</b> expression is matched. The optional <b>Escape</b> expression can be used to handle instances such as escaped quotes within strings.
<b>skipBalanced(Origin: Expression, Target: Expression)</b> <b>skipBalanced(Origin: Expression, Target: Expression, Escape: Expression)</b>	Consumes input data (characters or tokens) until the <b>Target</b> expression is matched and the nesting level reaches zero. If the <b>Origin</b> expression is matched during this process, the nesting level is increased. If the <b>Target</b> expression is matched, the nesting level is decreased. When the nesting level reaches zero, the command exits with success. An optional <b>Escape</b> expression can be provided.
<b>skipEOF()</b>	Consumes all remaining data (characters or tokens) until the end of the file.
<b>fail()</b>	Causes the parser to fail the current rule, including any remaining definitions.
<b>warning()</b>	Inserts a warning into the resulting AST.
<b>except(Target: Expression,</b>	Consumes input data that matches the <b>Target</b> expression, but fail on data that matches the <b>Exception</b> expression. This operates somewhat similar to, but

Command	Description
<b>Exception: Expression)</b>	exactly the opposite of, the <b>skip</b> command.
<b>preProcess(Target: Expression)</b>	Evaluates an expression and uses that pre-processed data in multiple definitions. This is most useful within expression parsing, where the same left hand side expression will be evaluated against a number of operators. This command reduces the work the parser must do to make this happen.

#### Learn more



- [Grammar Instructions](#) 
- [AST Nodes](#) 

#### 7.3.8.1.5 AST Nodes

In defining a grammar, you would use **AST nodes** and **AST node attributes** that can be recognized in code engineering in Enterprise Architect, in the AST results that are returned by the **attribute**, **attributeEx** and **node** commands. These nodes and attributes are identified in the following tables. Any others will be ignored in code engineering.

#### FILE Node

The FILE node represents a file. It isn't mapped to anything, but contains all the required information.

Multiplicity	Contained Nodes/ Attributes	Description
<b>0..*</b>	PACKAGE	See <a href="#">PACKAGE Node</a> 
<b>0..*</b>	CLASS	See <a href="#">CLASS Node</a> 
<b>0..*</b>	IMPORT	
<b>0..*</b>	<i>COMMENT</i>	Field labels as part of a skip rule will be at the root level; the code generator looks for comments of this sort by position relative to the node.
<b>0..1</b>	<i>INSERT_POSITION</i>	This gives the position where new Classes, packages and method implementations can be inserted into the file. If it is not found, the code generator will automatically insert new items immediately after the last one is found in code.

#### PACKAGE node

The PACKAGE node corresponds to a namespace or equivalent in the file. When importing with *package per namespace*, Enterprise Architect will create a package directly under the import for this and place all Classes within it. When not importing namespaces, Enterprise Architect will look for Classes under this point, but it will do nothing with this node.

Additionally, if you are generating with namespaces enabled (see the *Code Options* topics for generic languages) a generated Class will not match a Class in code unless they are under the same package structure.

Contained in nodes: [FILE](#)<sup>[1711]</sup>

Multiplicity	Contained Nodes/Attributes	Description
1	NAME	See <a href="#">NAME Node</a> <sup>[1718]</sup> <sup>[1718]</sup>
0..*	CLASS	See <a href="#">CLASS Node</a> <sup>[1712]</sup>
0..*	PACKAGE	
0..1	OPEN_POSITION	Gives the position where the package body opens. This can also be used as an insert position.
0..1	INSERT_POSITION	Gives the position where new Classes and packages can be inserted into the file. If it is not found, the code generator will automatically insert new items immediately after the last one is found in code.
0..1	SUPPRESS	Prevents indenting when inserting into this package.

### CLASS/INTERFACE Node

The CLASS (or INTERFACE) node is the most important in code generation. It is brought in as Class (or Interface) Objects.

See *Class DECLARATION* (below) and [Class BODY](#)<sup>[1713]</sup>.




Contained in Nodes: [FILE](#)<sup>[1711]</sup>, [PACKAGE](#)<sup>[1711]</sup>, [Class BODY](#)<sup>[1713]</sup>

### CLASS Declaration








Contained in Nodes: CLASS/INTERFACE

Multiplicity	Contained Nodes/Attributes	Description
1	NAME	See <a href="#">NAME Node</a> <sup>[1718]</sup>



Multiplicity	Contained Nodes/Attributes	Description
0..*	PARENT	See <a href="#">PARENT Node</a>  .
0..*	TAG	See <a href="#">TAG Node</a>  .
0..1	DESCRIPTION	See <a href="#">DESCRIPTION Node</a>  .
1	NAME	The name of the Class. If there is a node NAME, that will overwrite this attribute.
0..1	SCOPE	The UML Scope of the Class - <b>Public</b> , <b>Private</b> , <b>Protected</b> or <b>Package</b> .
0..1	ABSTRACT	If present, indicates that this is an abstract Class.
0..1	VERSION	The version of the Class.
0..1	STEREOTYPE	The stereotype that Enterprise Architect should assign to the Class. This does not support multiple stereotypes.
0..1	ISLEAF	
0..1	MULTIPLICITY	
0..1	LANGUAGE	Generally, you do not need to set this.
0..1	NOTE	Generally not used as it is covered by the comments above the Class.
0..1	ALIAS	
0..*	MACRO	Adds a numbered Tagged Value that Enterprise Architect can use to round trip macros.

**Class BODY Node**Contained in Nodes: [CLASS/INTERFACE](#) .

Multiplicity	Contained Nodes/Attributes	Description
0..*	METHOD	See <a href="#">METHOD Node</a>  .
0..*	ATTRIBUTE	See <a href="#">ATTRIBUTE Node</a>  .
0..*	FIELD	See <a href="#">FIELD Node</a>  .
0..*	CLASS	See <a href="#">CLASS Node</a>  .
0..*	SCOPE	See <a href="#">SCOPE Node</a>  .
0..*	PROPERTY	
0..*	TAG	See <a href="#">TAG Node</a>  .
0..*	PARENT	See <a href="#">PARENT Node</a>  .
0..1	<i>OPEN_POSITION</i>	Gives the position where the Class body opens. This can also be used as an insert position.
0..1	<i>INSERT_POSITION</i>	Gives the position where new Class members can be inserted into the file. If it is not found, the code generator will automatically insert new items immediately after the last one is found in code.

### SCOPE Node

This is an optional feature for languages resembling C++ that have blocks that specify the scope of elements. The language needs to have a name specified that is used for the scope of all elements in the block. In all other respects it behaves identically to the **Class BODY** node.

Contained in Nodes: [Class BODY](#) .

Multiplicity	Contained Nodes/Attributes	Description
1	<i>NAME</i>	Used as the scope for all methods and attributes contained within the scope.

### METHOD Node

Contained in Nodes: [Class BODY](#)<sup>[1713]</sup>, [SCOPE](#)<sup>[1714]</sup>

Multiplicity	Contained Nodes/Attributes	Description
1	Method DECLARATION	See <i>Method DECLARATION Node</i> (below).

### Method DECLARATION Node

Contained in Nodes: [METHOD](#)<sup>[1714]</sup>

Multiplicity	Contained Nodes/Attributes	Description
0..1	TYPE	See <a href="#">TYPE Node</a> <sup>[1716]</sup> .
0..*	PARAMETER	See <a href="#">PARAMETER Node</a> <sup>[1717]</sup> .
0..*	TAG	See <a href="#">TAG Node</a> <sup>[1720]</sup> .
0..1	DESCRIPTION	See <a href="#">DESCRIPTION Node</a> <sup>[1720]</sup> .
0..1	MULTI PARAMETER	Supports Delphi's parameter list style of declaration; see <a href="#">PARAMETER Node</a> <sup>[1717]</sup> . This is the equivalent of <a href="#">FIELD</a> <sup>[1717]</sup> .
1	NAME	
0..1	TYPE	
0..1	SCOPE	
0..1	ABSTRACT	
0..1	STEREOTYPE	
0..1	STATIC	
0..1	CONST or CONSTANT	

Multiplicity	Contained Nodes/Attributes	Description
0..1	<i>PURE</i>	
0..1	<i>ISQUERY</i>	
0..1	<i>ARRAY</i>	
0..1	<i>SYNCHRONIZED</i>	
0..*	<i>MACRO</i>	
0..1	<i>CSHARPIMPLEMENTS</i>	Specifies special behavior for C#.
0..1	<i>BEHAVIOR</i>	Provides support for Aspect J, using behavior.
0..1	<i>SHOWBEHAVIOR</i>	Provides support for Aspect J, using behavior, and shows the reverse-engineered behavior on the diagram.

### ATTRIBUTE Node

Contained in Nodes: [Class BODY](#)<sup>[1713]</sup>, [SCOPE](#)<sup>[1714]</sup>

Multiplicity	Contained Nodes/Attributes	Description
1	TYPE	See <a href="#">TYPE Node</a> <sup>[1718]</sup> .
0..*	TAG	See <a href="#">TAG Node</a> <sup>[1720]</sup> .
0..1	DESCRIPTION	See <a href="#">DESCRIPTION Node</a> <sup>[1720]</sup> .
1	NAME	
0..1	TYPE	
0..1	SCOPE	

Multiplicity	Contained Nodes/ Attributes	Description
0..1	<i>DEFAULT</i>	
0..1	<i>CONTAINER</i> or <i>ARRAY</i>	
0..1	<i>CONTAINMENT</i>	Reference or value.
0..1	<i>STEREOTYPE</i>	
0..1	<i>STATIC</i>	
0..1	<i>CONST</i> or <i>CONSTANT</i>	
0..1	<i>ORDERED</i>	
0..1	<i>LOWBOUND</i>	
0..1	<i>HIGHBOUND</i>	
0..1	<i>TRANSIENT</i> or <i>VOLATILE</i>	

### **FIELD Node**

A field corresponds to **multiple** attribute declarations in one. Anything not defined in the Declarators but defined in the field itself will be set for each declarator. Everything supported in an attribute is supported in the field. If no declarators are found then this works in the same way as an attribute.

Contained in Nodes: [Class BODY](#)<sup>[1713]</sup>, [SCOPE](#)<sup>[1714]</sup>

Multiplicity	Contained Nodes/ Attributes	Description
0..*	DECLARATOR	See <a href="#">ATTRIBUTE Node</a> <sup>[1716]</sup> .

### **PARAMETER Node**

Contained in Nodes: [Method DECLARATION](#)<sup>[1715]</sup>, [TEMPLATE](#)<sup>[1719]</sup>

Multiplicity	Contained Nodes/Attributes	Description
1	TYPE	See <a href="#">TYPE Node</a> <sup>[1718]</sup> .
0..*	TAG	See <a href="#">TAG Node</a> <sup>[1720]</sup> .
0..1	DESCRIPTION	See <a href="#">DESCRIPTION Node</a> <sup>[1720]</sup> .
0..1	NAME	
0..1	TYPE	
0..1	KIND	Expected to be <b>in</b> , <b>inout</b> , <b>out</b> or <b>return</b> .
0..1	DEFAULT	
0..1	FIXED	
0..1	ARRAY	

### NAME Node

Contained in Nodes: [PACKAGE](#) <sup>[1711]</sup>, [Class DECLARATION](#) <sup>[1712]</sup>

Multiplicity	Contained Nodes/Attributes	Description
1	NAME	
0..*	QUALIFIER	
0..*	NAMEPART	An alternative to using NAME and QUALIFIER. A string of values, all except the last one taken as qualifiers. The last one is taken as the Name.

### TYPE Node

Contained in Nodes: [Method DECLARATION](#) <sup>[1715]</sup>, [ATTRIBUTE](#) <sup>[1716]</sup>, [PARAMETER](#) <sup>[1717]</sup>

Multiplicity	Contained Nodes/Attributes	Description
<b>0..1</b>	<i>TEMPLATE</i>	The entire text of the template is the name of the type. Only used if <i>NAME</i> is undefined. See <i>TEMPLATE Node</i> , below.
<b>1</b>	<i>NAME</i>	
<b>0..*</b>	<i>QUALIFIER</i>	
<b>0..*</b>	<i>NAMEPART</i>	An alternative to using <i>NAME</i> and <i>QUALIFIER</i> . A string of values, all except the last one taken as qualifiers. The last one is taken as the Name.

**TEMPLATE Node**Contained in Nodes: [TYPE](#)<sup>[1718]</sup>

Multiplicity	Contained Nodes/Attributes	Description
<b>0..*</b>	<i>PARAMETER</i>	See <a href="#">PARAMETER Node</a> <sup>[1717]</sup> .
<b>1</b>	<i>NAME</i>	

**PARENT Node**Contained in Nodes: [Class DECLARATION](#)<sup>[1712]</sup>

Multiplicity	Contained Nodes/Attributes	Description
<b>0..1</b>	<i>TYPE</i>	Has the value <b>Parent</b> , <b>Implements</b> or <b>VirtualP</b> .
<b>1</b>	<i>NAME</i>	
<b>0..*</b>	<i>QUALIFIER</i>	
<b>0..*</b>	<i>NAMEPART</i>	An alternative to using <i>NAME</i> and <i>QUALIFIER</i> . A string of values, all except the last one taken as qualifiers. The last one is taken as the Name.

Multiplicity	Contained Nodes/Attributes	Description
<b>0..1</b>	<i>INSTANTIATION</i>	

**TAG Node**

Contained in Nodes: [Class DECLARATION](#)<sup>[1712]</sup>, [Method DECLARATION](#)<sup>[1715]</sup>, [ATTRIBUTE](#)<sup>[1716]</sup>, [PARAMETER](#)<sup>[1717]</sup>

Multiplicity	Contained Nodes/Attributes	Description
<b>1</b>	<i>NAME</i>	
<b>0..*</b>	<i>VALUE</i>	
<b>0..1</b>	<i>MEMO</i>	
<b>0..1</b>	<i>NOMEMO</i>	
<b>0..1</b>	<i>GROUP</i>	

**DESCRIPTION Node**

Contained in Nodes: [Class DECLARATION](#)<sup>[1712]</sup>, [Method DECLARATION](#)<sup>[1715]</sup>, [ATTRIBUTE](#)<sup>[1716]</sup>, [PARAMETER](#)<sup>[1717]</sup>

Multiplicity	Contained Nodes/Attributes	Description
<b>0..*</b>	<i>VALUE</i>	

**Learn more**

- [Language Options](#)<sup>[2262]</sup>
- [Grammar Commands](#)<sup>[1709]</sup>



### 7.3.8.2 Editing Grammars

If you need to write and edit a grammar for **code imported in a new programming language**, you can do so using the in-built **Grammar Editor**.

**Access**    **Analyzer | Grammar Editor**

#### Create and Edit Grammar

Field/Button	Action	See also
<b>Open Grammar</b>	Display a browser through which you can locate and open the file containing the grammar you want to edit.	
<b>Save</b>	Save the current file.	
<b>Validate Grammar</b>	The grammar validation will run a series of tests on the current grammar to ensure its validity. Errors and warnings will be displayed informing you of both errors that will make the grammar unusable, and conditions where you may get unexpected results.	
<b>Help</b>	Display this Help topic.	

#### Context Menu Options

Field/Button	Action	See also
<b>Open File</b>	Display a browser through which you can locate and open the file containing the grammar you want to edit.	
<b>Validate</b>	The grammar validation will run a series of tests on the current grammar to ensure its validity. Errors and warnings will be displayed informing you of both errors that will make the grammar unusable, and conditions where you may get unexpected results.	
<b>Language</b>	The Grammar Editor defaults to normal Backus–Naur Form (nBNF). The <b>mBNF</b> option is also available.	
<b>Line Numbers</b>	Turn line numbers on or off in the grammar editor.	

#### Learn more

- [Grammar Framework](#)  1705

- [Grammar Syntax](#)<sup>[1706]</sup>
- [Parsing AST Results](#)<sup>[1722]</sup>
- [Profiling Grammar Parsing](#)<sup>[1722]</sup>

### 7.3.8.3 Parsing AST Results

The Abstract Syntax Tree (AST) is the code that Enterprise Architect sees as it processes a grammar.

You parse the text in the bottom half of the Grammar Editor window and review what is displayed as a result. You can either open a file or paste text in. If you have pasted text that corresponds to something that cannot appear at the file level (such as Operation Parameters) you can select an alternative rule to use as a starting point. The parse will then commence from that rule.

**Access**    **Analyzer | Grammar Editor > Grammar Debugger > AST Results**

#### Toolbar Options

Option	Action	See also
<b>Open File</b>	Open a sample input file to test against.	
<b>Parse</b>	Perform the parse operation. If the parse is successful, the AST Results tab will contain the resulting AST.	
<b>Select Rule</b>	This drop down allows you to select an alternative root rule for processing your sample source.	
<b>Help</b>	Display this Help topic.	

#### Learn more

- [Grammar Framework](#)<sup>[1705]</sup>
- [Grammar Syntax](#)<sup>[1706]</sup>
- [Editing Grammars](#)<sup>[1721]</sup>
- [Profiling Grammar Parsing](#)<sup>[1722]</sup>

### 7.3.8.4 Profiling Grammar Parsing

When you parse a grammar that you have created, it might show errors that you cannot immediately diagnose. To help you resolve such errors, you can review the process that the parser followed to generate the AST you can see, using the **Grammar Profiler**.

You again parse the text in the bottom half of the Grammar Editor window, but this time the tree shows **each rule** that the parser attempted, where it got to and if it passed or not. Rules for opening a file, pasting a file and setting the starting rule remain the same.

**Access**    **Analyzer | Grammar Editor > Grammar Debugger > Profiler Results**

### Toolbar Options

Option	Action	See also
<b>Open File</b>	Display a browser through which you can locate and open the file containing the grammar you want to edit.	
<b>Parse</b>	Perform the parse operation. If the parse is successful, the AST Results tab will contain the resulting AST, and the Profile Results tab will contain debug information regarding the path that the parser took through your grammar. The profile data is extremely useful when debugging a new grammar.	
<b>Select Rule</b>	If you want to use a different root rule for processing your sample source, click on the drop-down arrow and select the alternative rule.	
<b>Help</b>	Display this Help topic.	

### Notes

- Because profiling can take a very long time for large files, the Profile Results tab is not filled if you are not displaying that tab when you begin parsing

### Learn more

- [Grammar Framework](#)<sup>[1705]</sup>
- [Grammar Syntax](#)<sup>[1706]</sup>
- [Editing Grammars](#)<sup>[1721]</sup>
- [Parsing AST Results](#)<sup>[1722]</sup>

### **7.3.8.5 Example Grammars**

The Code Samples directory set up by the Enterprise Architect installer contains an example Grammar that you can load into the Grammar editor to review, and into the Grammar Debugger to parse and profile.

The Grammar example consists of two files:

- **test.ssl** - a simple sample language source file, in the style of C, and
- **ssl.nbnf** - a grammar for the simple sample language

The example illustrates:

- Tokenization (using the Lexer)
- Creation of a package
- Creation of a Class or Interface
- Creation of an attribute
- Creation of an operation (with parameters)
- Importing comments

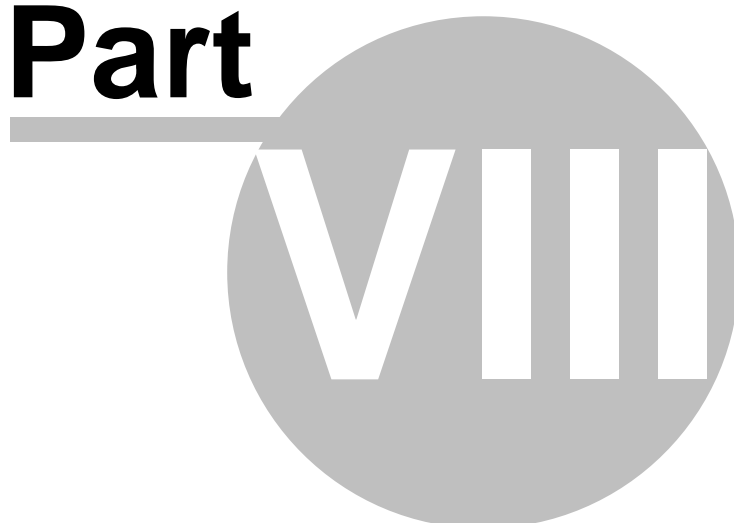
The Code Samples directory also contains two other Grammar files that you can examine:

- **Expressions Sample.nBNF** - this illustrates how expression parsing is set up and processed, with detailed comment text providing explanations
- **CSV Sample.nBNF** - an example grammar for processing CSV files

Learn more

- [Grammar Syntax](#)

**Part**



## 8 Requirement Models

**Requirements** are essentially what a system, application or business process is required to do. A requirement can be:

- Broad and high level, defining - for example - that a process is necessary to update a particular database
- More specialized and detailed, recording the expectation that - for example - a system call must always be acknowledged by return

Detailed requirements can be organized into a hierarchy culminating in a high-level requirement, so that satisfying each of the detailed requirements results in meeting the higher-level requirements and ultimately the top-level requirement. This hierarchical structure helps manage the complexity of large systems with thousands of requirements and many processes being developed to implement those requirements.

### Gathering Requirements

Gathering requirements is typically the first step in developing a solution, be it for developing either a system or a process. Requirements are gathered from all parties expected to use, maintain or benefit from the solution, and are organized into groups, functional areas and hierarchies as necessary. They can be transcribed into a spreadsheet or a requirements gathering or management tool, or they can be created within an integrated modeling tool such as Enterprise Architect.

The management of requirements is one of the more problematic disciplines in software development, for reasons such as:

- Diverse group input into the requirements
- Organizational boundary divisions
- Tool boundary divisions
- Volatility of requirements
- Imprecision and ambiguity in natural languages

These can cause issues with:

- Traceability and
- Integration with change and configuration management systems

Enterprise Architect can reduce or eliminate these problems in Requirements Management.

### Requirements Management and Enterprise Architect

Enterprise Architect is one of the few UML tools that integrate Requirements Management with other software development disciplines in the core product, by defining requirements within the model. Within Enterprise Architect, you can:

- **Create** and **view** requirements as entities and properties directly in the model, as simple **text descriptions** or as **diagrammatic representations** of the elements and their organization
- Collate the requirements in an external CSV file and then **import** them into your model
- Detail **use cases** and scenarios directly in the model
- Enter **standard attributes** (properties) for each requirement, such as difficulty, status and type, and **define your own attributes** (properties)

- **Trace** requirements to Use Cases, business rules, test cases and analysis artifacts (using, for example, the Relationship Matrix)
- Trace and view the impact of **changes** on requirements (through, for example, the **Traceability** window) and **review the changes** themselves
- Create customer-quality MS Word and HTML **reports** on requirements







### Notes

- All of these features are illustrated by examples in the *EAExample.eap* model, provided as part of your Enterprise Architect installation in the Enterprise Architect Program Files directory:

..\Program Files\Sparx Systems\EA

- If your project team are not using Enterprise Architect to **manage** Requirements, they can still access, use and work with them via the Cloud, from an **Open Services for Lifecycle Collaboration (OSLC)** client tool

### Learn more

- [Requirements Management with Enterprise Architect](#) (Online Resource)
- [Specification Manager](#) 
- [Requirements](#) 
- [Model Requirements](#) 
- [Create Requirements](#) 
- [Requirement Properties](#) 
- [Extend Requirement Properties](#) 
- [Connect Requirements](#) 
- [Import Requirements Using CSV](#) 
- [View Requirements](#) 
- [Trace Use Of Requirements](#) 
- [Traceability Window](#) 
- [Relationship Matrix](#) 
- [Manage Requirement Changes](#) 
- [Report on Requirements](#) 
- [OSLC Requirements Management](#) 

## 8.1 Specification Manager

In developing a model, you can use a wide range of Enterprise Architect tools to create the model specification. For the preliminary specification, such as defining Requirements, you can adopt one of two general methods:

- Diagram-based, creating UML-defined element icons in, for example, Requirements diagrams, and building the model graphically, or
- Text-based, creating elements as simple text descriptions, sketching out the elements, relationships and properties in a broad definition that you - or team members with more specific roles - can refine and develop into a more complete model

Your 'sketch pad' for the second approach is the **Specification Manager**.

The **Specification Manager** is a simple **document-based** interface to a selected Package in the model, providing the means of creating and reviewing elements as **text** representations of objects in the model, using a process that is familiar to all users including business professionals and management who might not have expertise in model development.

The Specification Manager is also an interactive reporting tool for displaying, in icon form, the status of various other features against each entry and providing access to edit each feature.

In the Specification Manager you can:

- Work on the element entries as **paragraph text** in the same way as you would edit a document using a word processor; this includes copying or dragging text from an external file into the Specification Manager, and importing data from an external spreadsheet (.csv) file
- Switch easily from element to element in a **single mouse click** or **key stroke**
- Show and in-place **edit** the current status of a number of properties of each entry
- Directly interface to a range of Enterprise Architect tools that you can quickly invoke to display, edit and discuss the properties of the selected Package or element; these tools span the range of Project Management and Model Management facilities, including:
  - Traceability (Traceability window and Relationship Matrix)
  - Change Management (Baselines and Auditing)
  - Maintenance
  - Testing
  - Risk Management
  - Documentation (document format, web format, and spreadsheet format)
  - Metrics
  - Review and Discussion (Team Review, Element Discussions and Model Mail)

Once any tool window is open, the details reflect the currently-selected element in the Specification Manager, so you can quickly run through the properties of a number of elements just by selecting them in sequence in the list

The Specification Manager makes it very easy to develop a simple, visual description of your model for initial specification, ongoing development and program management, with immediate access to a range of types of detailed information on any selected element, should you need this detail.

### Learn more

- [Specification Manager - Overview](#) 



- [Using the Specification Manager](#)<sup>[173]</sup>

### 8.1.1 Specification Manager - Overview

The **Specification Manager** is a **document-based** interface to a selected Package in the model, providing the means of creating and reviewing elements as **text** representations of model objects in that Package. You can rapidly define or edit properties of the elements in the Specification Manager window itself, and also immediately open a range of windows and dialogs to add or review further properties and characteristics specific to each element. A key feature of the Specification Manager is the speed and ease with which you can create, filter and review a large number of elements from one point, without necessarily developing or examining complex detail on each element.

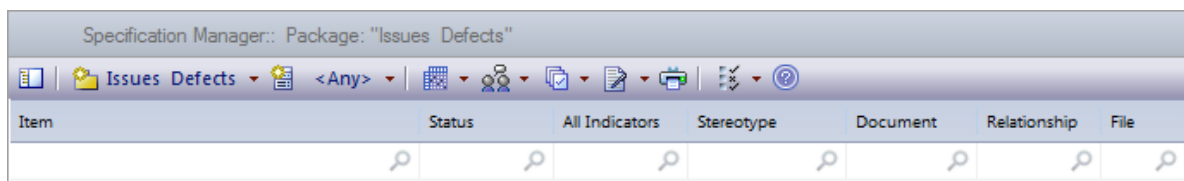
**Access**    **View | Specification Manager**

#### The Specification Manager View

The Specification Manager initially displays as a tabbed view in the application work area, containing a:

- Toolbar
- List header bar of column names, and
- Blank work area

The selected Package defaults to the first top-level View or Package in your model, and the element type defaults to any type held in the Package.



If you prefer, you can drag the Specification Manager tab out of the application workspace and use it as a floating or docked window.

To begin you:

- Select an existing Package to review, or create a new Package, using the second button in the Specification Manager toolbar
- If you want to select a specific type of element to review, use the third button in the Specification Manager toolbar - this also provides the option of switching back to displaying elements of **any** type in the Package

You can also adjust the **list header bar** to include the columns that show the information of interest to you, and the **display Filter bar** to further isolate particular elements for review. For an existing Package, the Specification Manager might now resemble this:

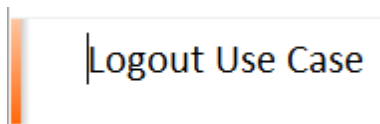
Specification Manager:: Package: "Log in/out Use Cases"						
Item	Status	All Indicators	Stereotype	Document	Relationship	
<b>1 Admin Logout</b>	Approved					
<b>1.1 AdminClose</b>	Implemented					
<b>2 Logout Use Case</b>	Approved					
<b>3 User Login</b>	Proposed		Performance			
User logins can have different access permissions.						

The display shows:

- The element names and any **notes** they might have (you can configure the display to hide the notes)
- The elements in parent/child relationships
- The text values of each **property** represented by a column, where those values have been set; you can quickly edit any of the values
- Icons in several columns, which indicate that the element has an associated object, file or linked document; you can double-click on these icons to display the object properties or document in the window in which it was created and, if necessary, you can edit the details

After you become familiar with what each indicator icon represents, you can display them all within the single **All Indicators** column to save space.

When you click on an element in the display, it is marked by a **vertical colored line** in the left margin to indicate that the element is selected, and a vertical or horizontal line against the element **property** you clicked on, to indicate that the focus is on that property. In the illustration, the **name** of the **Logout Use Case** element is selected. If you click again on the element property, the margin line changes color, to indicate that the property is open for **editing**. For example:



### Adding, displaying and changing information

When you create or review an element on the Specification Manager, you operate through one of three areas:

- The Specification Manager **toolbar**, which provides options to:
  - Change the Package or type of element under review
  - Open tools to create, display or review the properties, characteristics and associated records of the

Package or specific elements in the package

- Configure the **appearance** of the Specification Manager display and the types of information you might develop from the data in the display
- The **display area** itself, where you can:
  - Perform **in-place editing** by directly editing text (such as Version, Phase or element notes) or selecting alternative values from drop-down lists (such as Difficulty, Status or Priority)
  - Double-click on an indicator icon and review the record or document attached to the element
- The **element context menu**, which you can display by right-clicking anywhere on an entry in the display; this provides options for:
  - Creating and deleting elements
  - Editing element properties (including adding and reviewing **extended properties**, or **Tagged Values**)
  - Locating the element in the Project Browser and
  - Refreshing the display

#### Notes

- In the Corporate and extended editions of Enterprise Architect, a User Security system can be applied that restricts or enables access to a range of operations and functions; if you cannot access a function in the Specification Manager, check with your System Administrator or Security Administrator to see if you have access permissions to work with that function

#### Learn more

- [Specification Manager](#)<sup>[1728]</sup>
- [Using the Specification Manager](#)<sup>[1731]</sup>
- [Traceability](#)<sup>[1744]</sup>
- [Reviewing Elements](#)<sup>[1747]</sup>
- [Managing Changes](#)<sup>[1752]</sup>
- [Reporting](#)<sup>[1754]</sup>
- [Specification Manager Configuration](#)<sup>[1756]</sup>






### **8.1.2 Using the Specification Manager**

When you begin working in the Specification Manager, there are three general areas of activity you work through:



- Setting up the Specification Manager to suit your work requirements
- Creating and/or organizing the elements in the View
- Performing whatever work you need to do on the elements

The Specification Manager is extremely versatile, and you can use it very simply to create a set of basic elements or at varying levels of complexity to develop and manage those elements in your project, as you need.

#### Initial tasks

Task	Action	See also
<b>Specify the type of element to work on</b>	<p>Either:</p> <ul style="list-style-type: none"> <li>• Work with the default (all element types) or</li> <li>• Select a specific type of element</li> </ul> <p>You specify the element type first because when you come to browse for or select the Specification Package, if you have specified a particular element type the system <b>automatically filters</b> the Package selection to only those that contain elements of that type.</p>	<a href="#">Select Specification Type</a> 
<b>Select the Package to work in</b>	<p>Either:</p> <ul style="list-style-type: none"> <li>• Work with the default Package</li> <li>• Select another existing Package in the model, or</li> <li>• Create a new Package in the model</li> </ul>	<a href="#">Select Specification Package</a> 
<b>Set up List Header and Filter bars</b>	<p>Use the Field Chooser dialog to organize the columns that contain the type of information you want to work with.</p> <p>Some of the columns contain indicators that represent records and documents associated with an element. You can access the item represented by an indicator, by clicking on it.</p>	<a href="#">Customize Columns</a>  <a href="#">Indicator Columns</a> 
<b>Configure the display</b>	<p>As you begin, or at any other time in your use of the Specification Manager, you can change the appearance of the display to, for example, display any hierarchy of Packages under the selected Package in a separate panel, use smaller or larger text font, partially or totally hide Notes text, or show the element names in bold.</p> <p>You can further configure the display and the element definition by including level numbering, automatic naming and Project Glossary entries, and by applying customized properties such as additional Requirement Types and Tagged Value Types.</p>	<a href="#">Specification Manager Configuration</a> 

### Organize elements

Task	Action	See also
<b>Add new elements</b>	Create new elements in the model as direct members of the Package or as child elements of those members.	<a href="#">Adding Elements</a> 
<b>Delete elements</b>	You can filter out elements from the display by type or according to the value of various properties. However, if an element does not belong in the Package and model, you can delete it completely.	<a href="#">Deleting Elements</a> 

Task	Action	See also
<b>Reorganize elements</b>	<p>The Specification Manager reflects the sequence of elements in the Package as shown in the Project Browser.</p> <p>To put the elements in the Specification Manager in a different order, move them in the Project Browser.</p>	<a href="#">Move Objects Within a Package</a> <sup>[57]</sup>

### Review the elements

Task	Action	See also
<b>Create and Review Relationships</b>	<p>You can examine existing relationships and create new ones using the <b>Relationship Matrix</b>, creating and using specific Matrix <b>Profiles</b>.</p> <p>You can also review existing relationships by opening the <b>Traceability window</b>, either from the Toolbar or by double-clicking on the blue arrow displayed in the <b>Relationship</b> and <b>All Indicators</b> columns. If the element has no relationships, the blue arrow is not shown.</p>	<a href="#">Create Matrix Profile</a> <sup>[1744]</sup> <a href="#">Open Relationship Matrix</a> <sup>[1746]</sup> <a href="#">Traceability</a> <sup>[1744]</sup>
<b>Discuss and review elements amongst project members</b>	<p>It is possible to create review documents and have recorded text message discussions on the development of your elements, using the Team Review, Element Discussion and Model Mail facilities.</p> <p>If an element has an existing Element Discussion, this is represented by a 'speech bubble' icon in the <b>Discussion</b> and <b>All Indicators</b> columns. You can open the discussion by double-clicking on that icon.</p>	<a href="#">Reviewing Elements</a> <sup>[1747]</sup> <a href="#">Create Review Document</a> <sup>[1749]</sup> <a href="#">View Review Document</a> <sup>[1751]</sup>
<b>Make, Review and Manage Changes</b>	<p>You can easily <b>view and edit</b> elements directly in the Specification Manager or by displaying the appropriate window such as the Properties window or <b>Tagged Values</b> window.</p> <p>You can also <b>monitor</b> changes to the selected Package or element made anywhere in the system, using the <b>Baselines</b> and <b>Auditing</b> Facilities.</p> <p>Further, you can <b>manage</b> changes made by other users, opening the <b>Project Management</b> window to quickly assign <b>resources</b> and <b>risks</b> to the work on the element. You can review work on the elements in the Package by displaying the Specification Manager in another tab in the form of the <b>Project Gantt</b> chart.</p> <p><b>Maintenance</b> and <b>Testing</b> are integrated and easily available to help improve software quality from the beginning of your project.</p> <p>Existing records of these items are indicated by icons in</p>	<a href="#">Editing Elements</a> <sup>[1740]</sup>  <a href="#">Managing Changes</a> <sup>[1752]</sup>  <a href="#">Indicator Columns</a> <sup>[1743]</sup>

Task	Action	See also
	the corresponding column, and can be quickly opened by double-clicking on the icon.	
<b>Generate reports</b>	<p>If you should need to extract and/or publish the information in the Specification Manager, you can do so in a range of forms such as:</p> <ul style="list-style-type: none"> <li>• Simply printing the content of the view</li> <li>• Generating standard or customized document reports in RTF, DOCX or PDF format</li> <li>• Publishing on the web in HTML, or</li> <li>• Exporting to a spreadsheet tool in CSV format</li> </ul>	<a href="#">Reporting</a> <sup>[1754]</sup>

### Notes

- As you work on the information on the Specification Manager, periodically press **F5** to refresh the view to include any updates performed by other users elsewhere in the system
- In the Corporate and extended editions of Enterprise Architect, a User Security system can be applied that restricts or enables access to a range of operations and functions; if you cannot access a function in the Specification Manager, check with your System Administrator or Security Administrator to see if you have access permissions to work with that function

### 8.1.2.1 Select Specification Type

When you start to use the Specification Manager you can work on all element types in the Package (the default) or identify a **specific type of element** to work on, using the third button in the Specification Manager toolbar. You can select any type of element supported by Enterprise Architect or the MDG Technologies that are integrated with the system.

You specify the element type first because when you go on to locate the Package to work in, if you have identified a specific type or element the system automatically filters for Packages that already contain that type of element.

**Access** **View | Specification Manager |** 

#### Select the element type

Click on the drop-down arrow at the right-hand end of the button. A short menu of element types displays.

Click on the name of a specific element type to select that type **and** the **Limit Display to Selected Type** option.

To revert to listing any type of element in the Package, click on either the:

- **<Any>** option or
- **Limit Display to Selected Type** option again to **deselect** it

If the specific element type you require is not shown, click on the:

- **Structural** option to select from a short list of the most common structural element types, such as Class
- **Behavioral** option to select from a short list of the most common behavioral element types, such as Activity, or
- **Project Management** option to select from a list of the Project Management element types

If the element-type is a custom type created using a Profile, or is from another modeling language (such as a SysML requirement), click on the **Other** option and expand the appropriate technology sub-menus to locate the element type.

On the final submenu for each technology, you can select the **Activate <type name> Toolbox** option to use **that** submenu as the default shortlist of types offered when you click on the original drop-down arrow. This is useful if you intend to work within a specific technology for an extended period.

If you want to return to the initial element type shortlist, click on the **Activate Default Toolbox** option at the top of the drop-down list.

#### Learn more

- [Using the Specification Manager](#)<sup>[1731]</sup>
- [Select Specification Package](#)<sup>[1735]</sup>

### 8.1.2.2 Select Specification Package


When you start to work with the Specification Manager, you identify the **Package** in which to build your specification. When you open the facility, the Specification Package defaults to the first top-level Package in your model, as indicated by the second button in the window toolbar. If this is the Package in which you intend to work, you can go on to configure the display and create or review elements.

You might want to use a different Package that:

- Already exists in the model and already contains elements of the specified type
- Already exists in the model but contains no elements of the specified type
- You will create underneath the current Specification Package, OR
- You will create underneath a different Package elsewhere in the model

**Access**    **View | Specification Manager**

#### Locate existing Package containing specification elements

Step	Action	See also
1	Click on the drop-down arrow at the right hand end of the second toolbar button (indicated by the  icon).	
2	Select the <b>Open Specification Package</b> option.  The Select a Specification Package dialog displays. This dialog presents your model hierarchy, filtered to show only those Packages that already contain elements of the type you selected as the Specification element type.	<a href="#">Select Specification Type</a> <sup>[1734]</sup>

Step	Action	See also
3	To expand a Package, click on the arrow next to the Package name, or click on the <b>Expand All</b> button.	
4	Click on the required Package, and click on the <b>OK</b> button.  The Package name displays in the second button in the toolbar, and the elements in the Package are shown in the main body of the Specification Manager window, filtered to the specified type if appropriate.	<a href="#">Specification Manager - Overview</a> <sup>[1729]</sup>
5	Customize the columns and/or add elements to the Package.	<a href="#">Customize Columns</a> <sup>[1738]</sup> <a href="#">Adding Elements</a> <sup>[1738]</sup>

#### Locate existing empty Package

It is possible that you want to create your specification in a Package that has been set up for this purpose but does not yet contain any elements (or any of the specified type). In this case the Package would not be included in the Select a Specification Package dialog (above). To select this Package for the Specification Manager:

- Double click on the Package name in the Project Browser

The Package name displays in the second button in the toolbar.

If this does not work as described, display the Options dialog (**Tools | Options**), select the General page and click on the **Shows Properties** radio button in the Double Click on Browser panel.

#### Create new Package

This method creates a new Package underneath the current Specification Package.

Step	Action	See also
1	Click on the drop-down arrow at the right hand end of the first toolbar button, and select the <b>New Package</b> option.  The New Model Package dialog displays.	
2	In the <b>Name</b> field, type the name of the new Package. Optionally, you can also select the check boxes to automatically create and open a new diagram in the Package.	<a href="#">Add a Package</a> <sup>[772]</sup>



Step	Action	See also
3	Click on the <b>OK</b> button.  The new Package becomes the current Specification Package, named in the window toolbar. The Package is also added to the Project Browser, underneath the previous Specification Package.	
4	Customize the columns and/or add elements to the Package.	<a href="#">Customize Columns</a> <sup>[1738]</sup> <a href="#">Adding Elements</a> <sup>[1738]</sup>

### Create new Package elsewhere in the Model

This method creates a new Package underneath a different parent Package in the model.

Step	Action	See also
1	Click on the drop-down arrow at the right hand end of the second toolbar button, and select the <b>Add Package To</b> option.  The Browse Project dialog displays, exposing all the Packages in the model.	
2	Browse through the model and click on the existing Package under which to create the new Package.	
3	Click on the <b>OK</b> button.  The New Model Package dialog displays.	
4	In the <b>Name</b> field, type the name of the new Package. Optionally, you can also select the check boxes to automatically create and open a new diagram in the Package.	<a href="#">Add a Package</a> <sup>[772]</sup>
5	Click on the <b>OK</b> button.  The new Package becomes the current Specification Package, named in the window toolbar. The Package is also added to the Project Browser, underneath the selected parent Package.	
6	Customize the columns and/or add elements to the Package.	<a href="#">Customize Columns</a> <sup>[1738]</sup> <a href="#">Adding Elements</a> <sup>[1738]</sup>

### Learn more

- [Using the Specification Manager](#)<sup>[1731]</sup>

### 8.1.2.3 Customize Columns

The Specification Manager can display a wide range of properties for the elements in the selected Specification Package, each property being shown in a separate column. Using the **Field Chooser** dialog, you can customize the display of the columns so that you show as many or as few as you need, in an appropriate sequence. You can also use full or partial property values in the **Filter bar** as a means of filtering the display to show only those elements that have property values that match the filters.

**Access**    **View | Specification Manager | Right-click column headings | Field Chooser**  
**View | Specification Manager | Right-click column headings | Toggle Filter Bar**

### Using the Field Chooser

Drag the column headings that you want to include out of the Field Chooser and onto the header bar, in the position you want the column to occupy in the sequence of column headings. You can drag an existing heading from its current position in the bar to another position in the sequence, including making it the final column.


If you do not want to include a column that is already in the heading bar, drag it up or down out of the bar. It is automatically returned to the Field Chooser dialog.

### Using the Filter bar

In the Filter bar, you define a text string that any displayed values in the corresponding column must include.

In the **filter field** for a column, type the character or string that must form part of the value shown in each row of that column. As you type the string, element rows in the Specification Manager that do **not** match the filter are hidden from display.

To clear the text in the filter field, click on the green cross at the right of the field.

You can display or hide the Filter bar using either the right-click context menu or the  icon, and selecting the **Toggle Filter Bar** option.

### Learn more


- [Customizing the Search View \(Column headers\)](#)<sup>[708]</sup>
- [Customizing the Search View \(Filter Bar\)](#)<sup>[709]</sup>

### 8.1.2.4 Adding Elements

The Specification Manager is an excellent model development tool, through which you can **create elements** in the Package you are reviewing in a number of ways. You can add new elements either:

- Directly to the Package or
- As child elements of an element in the Package

The process of adding an element varies depending on whether you have set a specific element type as the **Specification Type**, or set the display to list all types of element in the Package. If you import a set of element data from an external spreadsheet (.csv) file, this setting will also filter the imported information to show elements of the specified type or all types.

**Access** **View | Specification Manager:**  **(Create new Element)**  
**View | Specification Manager | right-click window | Add New Element**  
**View | Specification Manager | Ctrl+N**  
**View | Specification Manager | right-click existing element | Add New Child** or  
**View | Specification Manager | click existing element | Ctrl+Shift+N**

### Add element of predefined Specification Type

If you have set the Specification Type to a particular type of element:

- An element of that type is immediately added to the Specification Manager; the element has either the system default element name or a name that you have set up under your own naming conventions
- The name is highlighted and, if you want to change it, you can immediately type in an alternative name
- If the element is a child of an existing element, the entry is added underneath the parent, indented and in a smaller font; the parent element is given a +/- expansion box

### Add element of any Specification Type

If the Specification Manager is listing elements of any type, or if you have set the Specification Type field to **<Any>**, the New Element dialog displays. You can use this dialog to create an element of any type that the model supports.

### Notes

- If you add an element of the defined Specification Type to the selected Package in the **Project Browser**, it is also added to the Specification Manager
- In the Corporate and extended editions of Enterprise Architect, a User Security system can be applied that restricts or enables access to a range of operations and functions; if you cannot access a function in the Specification Manager, check with your System Administrator or Security Administrator to see if you have access permissions to work with that function

### Learn more

- [Select Specification Type](#)<sup>[1734]</sup>
- [Set Auto Naming and Auto Counters](#)<sup>[904]</sup>
- [Add Elements Directly To Packages](#)<sup>[903]</sup>
- [Reporting](#)<sup>[1754]</sup> (for import from a CSV file)

## **8.1.2.5 Deleting Elements**

The Specification Manager provides a direct view of a Package in your model, and reflects the contents of the Package as it appears in the Project Browser. If you **delete** an element from the Specification Manager, it is removed **from the model** and therefore also disappears from the Project Browser. Any **child objects** that the element has are **also deleted** from the model.

**Access** **View | Specification Manager | right-click on element | Delete Selected** or  
**Click on element | Ctrl+D**

### Delete an element

When you select to delete an element in the Specification Manager, a prompt displays to confirm that you want to delete the element from the model.

Click on the **Yes** button. The element is removed from the Specification Manager and all other windows and diagrams in which it appears.

### Notes

- You cannot select more than one separate element at a time in the Specification Manager, and therefore you cannot delete multiple elements there; you can, however, select and delete multiple elements from the Project Browser, which also removes them all from the **Specification Manager** in one operation
- In the Corporate and extended editions of Enterprise Architect, a User Security system can be applied that restricts or enables access to a range of operations and functions; if you cannot access a function in the Specification Manager, check with your System Administrator or Security Administrator to see if you have access permissions to work with that function

### Learn more

- [Delete Elements from Diagram and Model](#)<sup>[92]</sup>

## 8.1.2.6 Editing Elements







When you have created elements in the Specification Manager, you can add and review the properties and characteristics of each element to whatever extent you need. You could simply set some basic properties such as the element version, status and general description, or you could develop extended properties (Tagged Values) or add a Linked Document to add more detailed and extensive information on the element.

**Access** **View | Specification Manager | Click twice on element**

### Editing tasks

Task	Action	See also
<b>Display/Edit Element Properties</b>	<p>Where a column in the Specification Manager contains a property value, you can double-click on the value and either:</p> <ul style="list-style-type: none"> <li>• Type a different value (such as the element name) or</li> <li>• Click on the field to display a drop-down arrow and select a value from the list</li> </ul> <p>Alternatively, click on the value and press <b>F2</b>.</p> <p>Otherwise, you can right-click on the item and select the:</p> <ul style="list-style-type: none"> <li>• <b>Properties</b> menu option (or press <b>Alt+Enter</b>) to display the Properties <b>dialog</b>, or</li> </ul>	<p><a href="#">Properties Dialog</a><sup>[95]</sup></p>



© 1998-2014 Sparx Systems Pty Ltd

Task	Action	See also
<b>Review extended properties on an element</b>	<p>Right-click on the element item and select the <b>Extended Properties</b> option (or click on the item and press <b>Ctrl+Shift+6</b>).</p> <p>The <b>Tagged Values window</b> displays, through which you can view, add, edit and delete extended properties (Tagged Values) on the selected element.</p> <p>Once the Tagged Values window is open, you can click on any element in the Specification Manager and immediately switch the focus of the Tagged Values window to that element.</p>	<a href="#">Tagged Values</a> 
<b>Create, edit and delete a linked document on an element</b>	<p>There are several ways in which you can create documentation on an element in the Specification Package, through the window <b>Toolbar</b> options.</p> <p>You can also create a <b>linked document</b> on the selected element, by:</p> <ul style="list-style-type: none"> <li>Right-clicking on the element line and selecting the <b>Create Linked Document</b> option, or</li> <li>Clicking on the element line and pressing <b>Ctrl+Alt+D</b></li> </ul> <p>The New Linked Document From Template dialog displays, from which you select any template you want to base the document on, or <b>None</b> for a free-style document with no template.</p> <p>When you click on the <b>OK</b> button, the Linked Document Editor view becomes active, and you can create a document describing any aspect of the selected element.</p> <p>If the element already has a linked document, the menu options available change to:</p> <ul style="list-style-type: none"> <li><b>Edit Linked Document</b> (<b>Ctrl+Alt+D</b>) and</li> <li><b>Delete Linked Document</b></li> </ul>	<a href="#">Reviewing Elements</a>  <a href="#">Reporting</a>  <a href="#">Linked Documents</a>   <a href="#">Edit Linked Documents</a>   <a href="#">Replace or Delete Documents</a> 

### Notes

- You can also copy-and-paste text from a text file into the Specification Manager, using the text to fill the element name or notes text area of an existing element entry
- In the Corporate and extended editions of Enterprise Architect, a User Security system can be applied that restricts or enables access to a range of operations and functions; if you cannot access a function in the Specification Manager, check with your System Administrator or Security Administrator to see if you have access permissions to work with that function









### Learn more

- [Adding Elements](#) 
- [Deleting Elements](#) 

### 8.1.2.7 Indicator Columns

An element specification can have any of a number of resources associated with it. You can obtain a quick indication that resources exist for the element by displaying the corresponding column (selected from the Field Chooser dialog) on the Specification Manager; if the resource exists, an icon displays in the column against that element. When an indicator is displayed, you can double-click on the icon to **immediately access** the record or the first in a set of records for the element, in the corresponding window. You can use these windows to edit, delete or (where appropriate) add to the records against the element.

#### Columns

Column Header	Indicator or	Indicates that...	See also
<b>Relationship</b>		The selected element has at least one relationship to another element, which is shown in the Traceability window.	<a href="#">The Traceability Window</a> <sup>[725]</sup>
<b>Resource</b>		There is at least one resource assigned to a task on the element, in the Project Management window.	<a href="#">Resource Allocation</a> <sup>[512]</sup>
<b>Risk</b>		There is at least one Risk item defined for the element, in the Project Management window.	<a href="#">Risk Management</a> <sup>[519]</sup>
<b>Maintenance</b>		There is at least one Maintenance item defined for the element, in the Maintenance window.	<a href="#">Working on Maintenance Items</a> <sup>[2623]</sup>
<b>Test</b>		There is at least one Test item defined for the element, in the Testing window.	<a href="#">Working On Test Records</a> <sup>[2605]</sup>
<b>File</b>		The selected element has at least one associated file, defined on the File page of the element Properties dialog.	<a href="#">Associated Files</a> <sup>[988]</sup>
<b>Discussion</b>		There is a discussion thread on this element, in the Element Discussions window.	<a href="#">Element Discussions</a> <sup>[365]</sup>
<b>Document</b>		There is a linked document for this element.	<a href="#">Linked Documents</a> <sup>[1044]</sup>
<b>All Indicators</b>	-	Any or all of the above items exist for the element, as indicated by the corresponding icon.	

#### Learn more

- [Specification Manager - Overview](#) <sup>[1729]</sup>
- [Using the Specification Manager](#) <sup>[1731]</sup>

- [Customize Columns](#)<sup>[1738]</sup>

### 8.1.3 Traceability

As you create and set up the elements in your Specification Package, using the Specification Manager, it is critical to:

- **Trace** any **existing** relationships those elements have to the elements in the model that realize them, or are realized by them
- Follow the relationships of such elements further, to **check** how changes to a source element or its relationships might affect **other relationship chains** in which the target elements are involved
- Create **new** relationships, where you can do so quickly and with an immediate view of how elements as a group are connected across the model

The two most significant tools for this are the **Traceability window** and the **Relationship Matrix**.

When you display the Traceability window, you can review every relationship in which the element selected in the Specification Manager is the source or target, the details of each of the related elements, and the relationships in which each of **those** elements are involved.

In the context of the Specification Manager, the Relationship Matrix provides a view of the relationships between the elements within the Specification Package, or between the elements of the Specification Package and a different Package. You can also use it to quickly create many relationships between the elements, defining a simple organization within the Specification Package and defining its relationships with other Packages.

**Access**    **View | Specification Manager:**  | **Open Relationship Matrix**

**View | Specification Manager:**  | **Traceability**

**Ctrl+Shift+4 (Traceability window)**

**Click on**  **in the Relationship or All Indicators columns (Traceability window)**

#### Learn More

- [Specification Manager](#)<sup>[1728]</sup>
- [Relationship Matrix](#)<sup>[727]</sup>
- [Create Matrix Profile](#)<sup>[1744]</sup> from within the Specification Manager
- [Open Relationship Matrix](#)<sup>[1746]</sup> from within the Specification Manager
- [The Traceability Window](#)<sup>[725]</sup>

#### 8.1.3.1 Create Matrix Profile

In using the Specification Manager you work on a specific Package and element type. If you want to use a Relationship Matrix **profile** to look more closely at the relationships of elements in that Package, it is therefore more logical to define the profile from the information **already set** in the Specification Manager, before you open the Relationship Matrix. The system helps you to do this.

**Access**    **View | Specification Manager:**  | **Matrix Profiles | Create new Profile**

#### Create the Matrix Profile



Field/Option/ Button	Action	See also
<b>Profile Name</b>	Type in a name to identify the profile.	
<b>Source</b>	Displays the name of the current Specification Package, as the source Package in the Relationship Matrix.  If you need to, you can click on the ( ... ) <b>Browse</b> button and select a different Package.	
<b>Type</b>	(For the Source Package) Displays the name of the current Specification element type. If you need to, you can click on the ( ... ) <b>Browse</b> button and select a different element type.	
<b>Target</b>	Defaults to the name of the current Specification Package as the target Package of elements in the relationships to examine.  If you want to examine relationships to elements in a different target Package, click on the <b>Browse</b> button ( ... ) and, on the Browse Project dialog, locate and select that Package.  Click on the <b>OK</b> button to put the Package name in the <b>Target</b> field.	
<b>Type</b>	(For the Target Package) Defaults to <b>&lt;All&gt;</b> target element types.  If you want to examine relationships to a specific type of element, click on the drop down arrow, scroll through the list and click on the appropriate element type name.	
<b>Relationship</b>	Defaults to <b>Realization</b> as the type of relationship to display.  If you want to examine relationships of a different type, click on the drop down arrow, scroll through the list and click on the appropriate relationship type name.	
<b>Direction</b>	Defaults to <b>Source -&gt; Target</b> as the direction of the relationships to examine.  If you want to examine relationships in the opposite direction, or in both directions, click on the drop-down arrow and select the appropriate option.	
<b>OK</b>	Click on this button to create the profile and close the dialog.  The Relationship Matrix automatically opens, applying the Profile that you have just defined.	
<b>Cancel</b>	Click on this button to discard any values you have entered, and to return to the Specification Manager.	

### Notes

- After you have created a Matrix profile, the name of the profile is added to a list of profiles that include the Specification Package and Specification element type, under the **Create new Profile** menu option; when you click on the profile name, the Relationship Matrix immediately displays with that profile in effect
- In the Corporate and extended editions of Enterprise Architect, a User Security system can be applied that restricts or enables access to a range of operations and functions; if you cannot access a function in the Specification Manager, check with your System Administrator or Security Administrator to see if you have access permissions to work with that function

### Learn more

- [Relationship Matrix](#)<sup>[727]</sup>
- [Matrix Profiles](#)<sup>[735]</sup>

## 8.1.3.2 Open Relationship Matrix

If you want to quickly create relationships on the elements in your Specification Package, or review the relationships that already exist, you can do so very easily using the Relationship Matrix. By opening the Relationship Matrix from within the Specification Manager, you can use the parameters already set in the Specification Manager to automatically configure the Relationship Matrix.

**Access**    **View | Specification Manager:** 

### Open the Relationship Matrix

Option	Action	See also
<b>Open Relationship Matrix</b>	Open the Relationship Matrix with the Specification Package set as the Source Package and the Specification Type element type as the Source element type.	
- As Source	Other fields default from the last settings of the Relationship Matrix; you can change any fields as necessary.	
- As Target	Open the Relationship Matrix with the Specification Package set as the Target Package and the Specification Type element type as the Target element type.  Other fields default from the last settings of the Relationship Matrix; you can change any fields as necessary.	<a href="#">Set Source and Target Package</a> <sup>[731]</sup>  <a href="#">Set Element Type</a> <sup>[729]</sup>
- As Both	Open the Relationship Matrix with the Specification Package set as both the Source Package and the Target Package, and the Specification Type element type as the Source and Target element type.  Other fields default from the last settings of the Relationship Matrix; you can change any fields as necessary.	

Option	Action	See also
<b>Matrix Profiles</b> <b>- Create New Profile</b>  <b>- &lt;Profile names&gt;</b>	Define a Matrix profile with the Specification Package set as the Source Package and the Specification element type set as the Source element type, and automatically open the Relationship Matrix with this profile in effect.	<a href="#">Create Matrix Profile</a> <sup>[1744]</sup>
	Select an existing Matrix profile from the list of profiles that include the Specification Package and Specification element type, and automatically open the Relationship Matrix with this profile in effect.	
<b>Previous Settings</b>	<p>Open the Relationship Matrix using the previously-processed settings; that is, the last settings under which changes were saved to the repository.</p> <p>These settings could have been defined <b>during</b> your use of the Specification Manager, or in the Relationship Matrix <b>independent</b> of any use of the Specification Manager.</p>	

**Notes**

- In the Corporate and extended editions of Enterprise Architect, a User Security system can be applied that restricts or enables access to a range of operations and functions; if you cannot access a function in the Specification Manager, check with your System Administrator or Security Administrator to see if you have access permissions to work with that function

**Learn more**





- [Relationship Matrix](#)<sup>[727]</sup>

**8.1.4 Reviewing Elements**

When you are managing or working as a member of a project team, it is useful to have mechanisms for reviewing the work you are doing and keeping some record of those reviews, whether they be casual discussions between peers or coordinated, structured and formal reviews. The Specification Manager provides a number of facilities that support the full range of review types, focussed on the elements that are defined in the Specification Package.

**Review facilities**

Facility	Description	See also
<b>Review Documents</b>	<p>It is possible to create any number of <b>Review Documents</b> on the elements of the Specification Package, both to formalize and guide specific review processes and to present the element information to be reviewed.</p> <p>The documents are stored in and easily accessed from the Team Review feature.</p>	<a href="#">Create Review Document</a> <sup>[1749]</sup>

Facility	Description	See also
<b>Bind Package to Team Review Folder</b>	By default, Review Documents are stored in folders under a special category of the Team Review, called <b>Formal Reviews</b> . You can, however, generate those documents elsewhere in the Team Review, by binding the Specification Package to whichever Team Review folder (topic) is most appropriate to the Package.	<a href="#">Create Review Document</a> <sup>[1748]</sup>
<b>Team Review</b>	<p>The <b>Team Review</b> tools provide a number of other facilities that can be of use in discussing the development and features of an element, such as the <i>Resources</i> folder or the ability to develop a sequence of communications on a specific element, on a group of elements, or on some broader aspect of the model development.</p> <p>The Specification Manager provides direct access to the Team Review, through this menu path:</p> <p><b>View   Specification Manager:</b>  <b>  Team Review</b></p>	<a href="#">Team Review Tools</a> <sup>[343]</sup>
<b>Element Discussions</b>	<p>Using the <b>Element Discussions</b> facility, you can initiate or respond in a general, ad-hoc discussion amongst your team members, on a specific element. The facility maintains a record of each initial comment or question, and every response to that comment.</p> <p>There is no limit on the number of different discussions, on the number of responses, or on the length of each item, although the facility is better suited to short queries and responses.</p> <p>When there is a discussion on an element, the  icon displays in the <b>Discussion</b> and <b>All Indicators</b> columns on the Specification Manager. Simply double-click on the icon to display the discussion.</p> <p>You can also access the Element Discussion window through this menu path:</p> <p><b>View   Specification Manager:</b>  <b>  Element Discussions</b></p>	<a href="#">Element Discussions</a> <sup>[365]</sup>
<b>Model Mail</b>	<p>The Model Mail facility is a personal, project-internal email service. You can use it to send emails to specific individuals or groups on your project, concerning the Specification Manager contents or functions, or any other subject you need to discuss.</p> <p>You can access Model Mail through this menu path:</p> <p><b>View   Specification Manager:</b>  <b>  Model Mail</b></p> <p>If you want to discuss a particular element, click on it in the Specification Manager and use the Model Mail <b>Create   Insert Quick Link</b> context menu option to create a hyperlink in the mail message to the element.</p>	<a href="#">Model Mail</a> <sup>[565]</sup>

### 8.1.4.1 Create Review Document

When you are managing the work of a project team, it is useful to have a mechanism for coordinating and guiding the work that the team members are required to do on the elements in the Specification Package, such as a **review document**. You can generate this type of document quickly and simply within the Specification Manager, to be distributed, accessed and read through the **Team Review** facility.

There are, broadly, two types of review document that you can generate:

- A specification document that contains the **material to be reviewed** - the individual features of each element - such as a Model Overview, List of Issues or Maintenance Report, and
- A management or review document that **guides** the reader in **applying** tests, checks and procedures across the Package, such a Phase Review, Test Plan or Training Plan

You can generate both kinds of document from one of a range of templates, or create a document without using any template. The templates are all those available through the Report Generator (specification documents) or Linked Document Editor (management documents), whether system-provided or user-designed.



The **specification** documents automatically structure information from the Specification Manager according to the selected template and should not require further editing to serve their purpose (although it is possible to edit them if you wish). They are therefore termed '**auto-generated documents**'.

By default, Review Documents are stored in folders (Topics) in a special category of the Team Review, called **Formal Reviews**. You can, however, generate those documents **elsewhere** in the Team Review, by binding the Specification Package to whichever Team Review folder is most appropriate to the Package. For example, if you have a modular structure in the Team Review, you might want to hold the Review Documents for a Specification Package in the same section of the Team Review as folders containing discussions and documents on the part of the model containing that Package.




**Access**    **View | Specification Manager:**  | **Create Review Document**  
                  **View | Specification Manager:**  | **Auto Generate Document**  
                  **View | Specification Manager:**  | **Bind Package to Team Review Folder**

#### Create a Review Document

Field/Option/ Button	Action	See also
<b>Name</b>	On the Review Document dialog, type a name for the document.  This is the name that you use to recognize and open the document in the Team Review folders.	<a href="#">View Review Document</a> <sup>[175]</sup>
<b>RTF Template</b>	Click on the drop-down arrow and select either: <ul style="list-style-type: none"> <li>• <b>None</b> to create a free-format document yourself, or</li> <li>• A document template name, to create a document of that type</li> </ul>	
<b>OK</b>	Click on this button to open the document in the Document Editor, which you can use to populate the document with text and images,	<a href="#">Custom Template</a>

Field/Option/ Button	Action	See also
	<p>format it and print it.</p> <p>At the same time:</p> <ul style="list-style-type: none"> <li>• The Team Review window opens</li> <li>• The <b>Formal Reviews</b> category is created (if it does not already exist) in the Team Review</li> <li>• A folder is created with the same name as the Specification Package (if it does not already exist) in the Formal Reviews category</li> <li>• The review document name is added to this folder</li> </ul> <p>If you have bound the Package to a different Team Review folder (see below), the Review Document is created in that folder instead of under the Formal Reviews category.</p>	<p><a href="#">Design Options</a>  (for editing facilities)</p> <p><a href="#">Team Review Tools</a> </p>

#### Auto-generate a Review Document

Field/Option/ Button	Action	See also
<b>Name</b>	<p>On the Review Document dialog, type a name for the document.</p> <p>This is the name that you use to recognize and open the document in the Team Review folders.</p>	<p><a href="#">View Review Document</a> </p>
<b>RTF Template</b>	Click on the drop-down arrow and select a document report system-template name, to create a document of that type	
<b>OK</b>	<p>Click on this button to generate the document through the Document Report Generator, containing the information currently displayed in the Specification Manager, within the parameters of the report template.</p> <p>At the same time:</p> <ul style="list-style-type: none"> <li>• The Team Review window opens</li> <li>• The <b>Formal Reviews</b> category is created (if it does not already exist) in the Team Review</li> <li>• A folder is created with the same name as the Specification Package (if it does not already exist) in the Formal Reviews category</li> <li>• The review document name is added to this folder</li> </ul> <p>The document is displayed in the Team Review tab for any further editing and modification you need, although the document is effectively ready for immediate use.</p> <p>If you have bound the Package to a different Team Review folder (see below), the Review Document is created in that folder instead of under the Formal Reviews category.</p>	<p><a href="#">Team Review Tools</a> </p> <p><a href="#">Custom Template Design Options</a>  (for editing facilities)</p>

Field/Option/ Button	Action	See also

#### Bind Package to different Team Review folder

Step	Action	See also
1	On the Select a Topic to Link dialog, click on the expansion box next to the appropriate category name to display the topic folders it contains.	
2	Click on the appropriate topic folder name.	
3	Click on the <b>OK</b> button. When you: <ul style="list-style-type: none"> <li>Create a Review Document, it will now be automatically stored in the selected topic folder</li> <li>Select the <b>Show Current Review Folder</b> option, the Team Review window will open with the selected folder name highlighted and expanded</li> <li>Select the <b>Current Reviews</b> option, the submenu will list the documents held in the selected folder</li> </ul>	<a href="#">View Review Document</a> <sup>[1751]</sup>

#### Notes

- For a given Specification Package, you can generate as many review documents as you require
- Before you generate a Review Document from the Specification Manager, you could run the system Spell Checker over the Specification Package to remove any errors in the text

#### Learn more

- [Reporting](#)<sup>[1754]</sup> (Spell Checker option)
- [Custom Document Templates](#)<sup>[2681]</sup>
- [Create Linked Document Templates](#)<sup>[1098]</sup>

### 8.1.4.2 View Review Document

In the Specification Manager, you can generate a range of **documents** that either **guide** a structured review of the elements you have listed, or provide the actual **material** - the descriptions, features and properties of the elements - to be reviewed. These documents are, by default, stored in the **Formal Reviews** category of the **Team Review**, from which point they can be accessed in a number of ways. Whichever access path you select, the document opens in the application work area and you can read the document online, edit it and/or print it.

If you want to edit or print the document, you can do so using the facilities of the common document editor, which is used in creating and editing both documents and document templates.

The Specification Manager also provides a facility for changing the default folder into which the Review Documents are generated. and from which you can select the documents to open.

**Access** **View | Specification Manager:**  |

- **Current Reviews | <document name>**
- **Show Current Review Folder | double-click <document name> in Team Review OR**
- **Team Review | Formal Reviews | <Package name> | double-click <document name> OR**

**View | Team Review | Formal Reviews | <Package name> | double-click <document name>**

#### Notes

- If you have changed the default folder for the Package (by binding the Package to a folder other than those in the Formal Reviews category), the **Current Reviews** and **Show Current Review Folder** options will automatically show the new default folder and the documents it contains; if you access the Team Review window directly, you would need to know the name and location of the new default folder

#### Learn more

- [Create Review Document](#) <sup>[1749]</sup>
- [Team Review Tools](#) <sup>[343]</sup>
- [Custom Template Design Options](#) <sup>[2704]</sup>

## 8.1.5 Managing Changes

If you are responsible for a team of people who are working on the specification, you can manage their work and monitor the changes that they are making within the Specification Manager. You can use the Specification Manager toolbar options and column indicators to open a number of management and monitoring tools, keeping the focus on the specification work rather than having to establish the focus in each management tool separately.

**Access** **View | Specification Manager:** 

#### Management and Monitoring facilities

Facility	Description	See also
<b>Manage Package Baselines</b>	Click on this option to open the <b>Package Baselines</b> dialog, which you can use to capture the state of development of the Specification Package at any appropriate point in time, as a <b>Baseline</b> . You can then compare the current state of the Package with a previously-captured Baseline and establish the changes that have been made since that Baseline was made.  You can also roll back any of the changes that you do not want to keep.	<a href="#">Package Baselines</a> <sup>[457]</sup> <a href="#">Manage Baselines</a> <sup>[467]</sup>
<b>View Audit Trail</b>	Click on this option to display the <b>Audit View</b> and the <b>Audit History</b> tab of the System Output window.  You can use the Audit View initially to turn on and configure Auditing, and then to perform audit checks on changes to the	<a href="#">Auditing</a> <sup>[446]</sup> <a href="#">The Audit View</a> <sup>[457]</sup> <a href="#">The System Output Window</a> <sup>[169]</sup>



Facility	Description	See also
	elements and connectors in the Specification Package (which might include changes to objects other than those of the Specification element type). The results of the audit check are displayed both in the Audit View and in the System Output window.	
<b>Project Management</b>	<p>Click on this option to display the <b>Resource Allocation</b> tab of the <b>Project Management</b> window. You can use this tab to define and allocate work for the people (resources) working on the Specification Package, either at Package level or at element level (select an element on the Specification Manager before defining the work).</p> <p>You can also select the <b>Risk</b>, <b>Effort</b> and <b>Metrics</b> tabs to define other aspects of work on an element.</p> <p>Once you have allocated a resource or defined a risk, an icon displays on the element row in the corresponding column of the Specification Manager, and in the <b>All Indicators</b> column. You can double-click on the icon to redisplay the Project Management window and review the record, and to add more records if necessary.</p>	<p><a href="#">The Project Management Window</a> <sup>[510]</sup></p> <p><a href="#">Resource Allocation</a> <sup>[512]</sup></p> <p><a href="#">Risk Management</a> <sup>[519]</sup></p> <p><a href="#">Effort Management</a> <sup>[518]</sup></p> <p><a href="#">Metrics</a> <sup>[520]</sup></p> <p><a href="#">Indicator Columns</a> <sup>[1743]</sup></p>
<b>Maintenance</b>	<p>Click on this option to display the <b>Defects</b> tab of the <b>Maintenance</b> window, on which you can create a record to define a problem or fault in the Specification Package or a selected element within it.</p> <p>You can also select the <b>Changes</b>, <b>Issues</b> and <b>Tasks</b> tabs to create other Maintenance records associated with the Package or selected element.</p> <p>When you create one of these records, an icon displays on the element row in the <b>Maintenance</b> and <b>All Indicators</b> columns of the Specification Manager. You can double-click on the icon to redisplay the Maintenance window and review the record, and to add more records if necessary.</p>	<p><a href="#">Maintenance</a> <sup>[2619]</sup></p> <p><a href="#">Indicator Columns</a> <sup>[1743]</sup></p>
<b>Testing</b>	<p>Click on this option to display the <b>Unit (Test)</b> tab of the <b>Testing</b> window, on which you can create a record to define a Unit Test for the Specification Package or a selected element within it.</p> <p>You can also select the other test type tabs to create other test records associated with the Package or selected element.</p> <p>When you create one of these records, an icon displays on the element row in the <b>Test</b> and <b>All Indicators</b> columns of the Specification Manager. You can double-click on the icon to redisplay the Testing window and review the record, and to add more records if necessary.</p>	<p><a href="#">Testing</a> <sup>[2604]</sup></p> <p><a href="#">Indicator Columns</a> <sup>[1743]</sup></p>
<b>View Package as Gantt</b>	Click on this option to display the <b>Package Browser</b> view in <b>Gantt</b> format, for the Specification Package. This view lists all the elements in the Package, showing their assigned resources and the progress of work on their separate tasks, both in text on the left hand side and as a graphical timeline on a calendar on the right.	<a href="#">Element View</a> <sup>[542]</sup>

Facility	Description	See also
	You can review the element properties or resource details by double-clicking on the item.	

### Notes

- If a record or entry does not display on the Specification Manager when you save the record, press **F5** to refresh the display
- Package Baselines and Auditing are available in the Corporate and extended editions of Enterprise Architect
- In the Corporate and extended editions of Enterprise Architect, a User Security system can be applied that restricts or enables access to a range of operations and functions; if you cannot access a function in the Specification Manager, check with your System Administrator or Security Administrator to see if you have access permissions to work with that function

### Learn more

- [Specification Manager - Overview](#)<sup>[1729]</sup>
- [Using the Specification Manager](#)<sup>[1731]</sup>

## 8.1.6 Reporting

As you develop the elements in the Specification Package, you can produce a number of documents and reports to either disseminate the information you have compiled or to extract additional information on the elements from other aspects of model development and management, that might not be immediately visible in the Specification Manager.

You can also export the data you have created to a .csv file and perhaps onwards to another tool, or import element information created outside Enterprise Architect.

**Access** **View | Specification Manager:** 

### Review facilities

Facility	Description	See also
<b>Spell-check the Package</b>	<p>Before generating and/or printing any information from the Specification Manager, you could run the system Spell Checker over the Specification Package to remove any errors that might exist in the element names and text.</p> <p>Select the <b>Spell Check Current Package</b> option to display the Spell Check: &lt;Package Name&gt; dialog, and click on the <b>Start</b> button.</p>	<a href="#">Using the Spell Checker</a> <sup>[553]</sup>
<b>Print the element list</b>	<p>A print-out of the Specification Manager element list is a simplified version of the screen display:</p> <ul style="list-style-type: none"> <li>• All elements are included, both parent and (indented) child</li> </ul>	

Facility	Description	See also
	<p>elements</p> <ul style="list-style-type: none"> <li>• The columns are narrowed to fit the page width (if necessary)</li> <li>• The indicator icons and element notes are not printed</li> <li>• The font size and type face are uniform across the report</li> </ul> <p>To generate the print-out:</p> <ol style="list-style-type: none"> <li>1. Click on the <b>Print List</b> option and, in the Generate Report dialog type a name for the document.</li> <li>2. Click on the <b>OK</b> button and specify the printer settings you require.</li> <li>3. Click on the <b>OK</b> button to print the list.</li> </ol>	
<b>Generate documentation in report document format</b>	<p>Enterprise Architect provides a comprehensive report generation facility, through which you can create a range of reports on different aspects of your model. You can apply this facility to the elements of a Specification Package, to generate a report in any of these formats:</p> <ul style="list-style-type: none"> <li>• <b>RTF</b></li> <li>• <b>PDF</b></li> <li>• <b>DOCX</b></li> </ul> <p>Select the <b>Generate Documentation (RTF/PDF)</b> option.</p> <p>The Generate Documentation dialog displays, with the <b>Root Element</b> field set to <b>Selected Elements</b> (the Specification Package elements, as the target contents of the report).</p>	<p><a href="#">Document Reports</a> <sup>[2640]</sup></p> <p><a href="#">Generate Documentation</a> <sup>[2641]</sup></p>
<b>Generate documentation in web page format (HTML)</b>	<p>As you develop the Specification Package elements you can publish them on the web, where the outline structure closely mirrors the Package structure and makes it very simple to explore on-line. You export the Package to web pages, where the web report provides an easy to use, highly detailed, Javascript-based structure. In addition, any hyperlinks in the elements make browsing to related information very simple.</p> <p>Select the <b>Publish as HTML</b> option.</p> <p>The Publish as HTML dialog displays, with the Specification Package name in the <b>Package</b> and <b>Title</b> fields.</p>	<p><a href="#">Web Reports</a> <sup>[2741]</sup></p> <p><a href="#">Create a Web Report</a> <sup>[2741]</sup></p>
<b>Generate spreadsheet file (CSV) and import from a spreadsheet</b>	<p>In Enterprise Architect you can import data from a spreadsheet into a Package and export element data from a Package into a .csv file.</p> <p>You can perform both of these operations within the Specification Manager, to bring in an existing set of element definitions to the Specification Package or transfer the Enterprise Architect element definitions to a different location.</p> <p>Select the <b>CSV Import/Export</b> option.</p>	<p><a href="#">CSV Import and Export</a> <sup>[491]</sup></p>

Facility	Description	See also
	<p>As part of either operation, you need to create a CSV Specification file.</p> <p>Whilst your display might be filtered to show a specific type of element, the operations work on the Package itself and incorporate all types of element within the Package.</p> <p>After you perform an import, refresh the display (press <b>F5</b>) to show the imported elements.</p>	
<b>Generate Quality Assurance and Metrics reports</b>	<p>Enterprise Architect provides several Quality Assurance reports for checking the development and impact of development of the elements in a Package. You can access these within the Specification Manager, so that the Specification Package is automatically the target of the report.</p> <p>You can also view and adjust the metrics report should you want to monitor the impact on the project metrics of the work that the elements in the specification might entail.</p>	<p><a href="#">Testing Details Report</a> <sup>[2736]</sup></p> <p><a href="#">Implementation Details Report</a> <sup>[2737]</sup></p> <p><a href="#">Dependency Details Report</a> <sup>[2739]</sup></p> <p><a href="#">Maintenance Report</a> <sup>[2740]</sup></p> <p><a href="#">Use Case Estimation</a> <sup>[584]</sup></p> <p><a href="#">Estimating Project Size</a> <sup>[588]</sup></p>

#### Notes

- In the Corporate and extended editions of Enterprise Architect, a User Security system can be applied that restricts or enables access to a range of operations and functions; if you cannot access a function in the Specification Manager, check with your System Administrator or Security Administrator to see if you have access permissions to work with that function

### 8.1.7 Specification Manager Configuration

As you work with the Specification Manager, you can tailor the display to provide more precise information, and to better represent the information. The Specification manager provides options to:

- Create reference data to work from
- Display a separate panel showing the Package Hierarchy underneath the Specification Package
- Modify the appearance of the entries, including the amount of Notes text to show
- Apply level numbering and automatic element naming and numbering

#### Learn more

- [Specification Manager - Overview](#) <sup>[1729]</sup>
- [Appearance Options](#) <sup>[1757]</sup>
- [Model Options](#) <sup>[1758]</sup>
- [Package Options](#) <sup>[1759]</sup>


### 8.1.7.1 Appearance Options

It is possible to tailor the appearance of the Specification Manager view to better display the information you want to examine.

**Access** **View | Specification Manager:**



#### Adjust the appearance of the display

Aspect	Description	See also
<b>Display of Notes</b>	<p>In the Specification Manager you can choose to hide, check, or create and edit Notes text on the screen. If not hidden, any Notes for an element are displayed underneath the element name.</p> <p>Click on either:</p> <ul style="list-style-type: none"> <li>• <b>Hide Notes</b> to not show the Notes at all; the element rows close up</li> <li>• <b>Notes Preview</b> to show the first line of Notes text for each element; in this mode, you cannot edit the Notes text in the Specification Manager, only in the Notes window</li> </ul> <p>or</p> <ul style="list-style-type: none"> <li>• <b>Full Notes</b> to show the complete Notes text for each element; in this mode you can create, edit and delete Notes text in the Specification Manager and perform the same operations as are available in the Notes window context menu</li> </ul>	<a href="#">Notes</a> 
<b>Display element names in bold</b>	Click on the <b>Bold Requirement Name</b> option. All element names, including the names of child elements, are displayed in bold.	
<b>Enable the use of Expansion boxes on parent elements</b>	<p>If an element has child elements, you can configure the Specification Manager to either:</p> <ul style="list-style-type: none"> <li>• Hide the child elements but provide an expansion box that you can click on to open and collapse the element hierarchy (select the <b>Allow Collapsible Regions</b> option), or</li> <li>• Automatically display the child elements indented underneath the parent element (deselect the <b>Allow Collapsible Regions</b> option)</li> </ul>	
<b>Change Font Sizes of display</b>	<p>Across the Specification Manager view, the various display components such as parent element names, child element names Notes/property values and column headings each have their own font size. You can select to increase or decrease the font sizes <b>as a set</b> across the display, relative to the current sizes.</p> <p>Select the <b>Font Size</b> option and either:</p> <ul style="list-style-type: none"> <li>• <b>Small</b></li> <li>• <b>Medium</b> or</li> </ul>	

Aspect	Description	See also
	<ul style="list-style-type: none"> <li>• <b>Large</b></li> </ul>	

### 8.1.7.2 Model Options

The Specification Manager helps you to develop the elements in a Package from 'the ground up'. You use standard, or reference, data in doing this, such as the names of team members who can create and edit model data, or the automatic naming system to apply in creating your elements. Because you can start from an early stage in the model's development, some of the reference data you want to use might not exist. The Specification Manager, therefore, also makes it easier to call up the facilities for defining the types of data you need.

Access   **View | Specification Manager:**  | **Configure**

#### Configure data for development

Data Type/Option	Description	See also
<b>Project Glossary</b>	<p>It is possible to create a Glossary of terms and definitions used throughout the Project, and to automatically link to the definition of a term where it occurs in the model text.</p> <p>Click on this menu option to to display the Project Glossary view, where you initially set up the Glossary.</p>	<a href="#">Project Glossary View</a> <sup>[534]</sup>
<b>Requirement Types</b>	<p>Requirement elements can be of one of seven system-defined types, as shown in the <b>Stereotype</b> column of the Specification Manager, if you have selected to work on Requirements.</p> <p>If these types are not sufficient for your needs, you can define alternatives on the Requirement tab of the General Types dialog.</p> <p>Click on this menu option to display the General Types dialog.</p>	<a href="#">Requirement Types</a> <sup>[1166]</sup>
<b>Project Authors</b>	<p>A Project Author is a system user who has responsibility for creating modeling structures and data in a project. Initially there are no Project Authors defined on the system, and you might need to identify the users who are in your development team.</p> <p>Click on this menu option to display the People dialog, on which you define the Project Authors.</p>	<a href="#">Project Authors</a> <sup>[1153]</sup>
<b>Auto Names and Counters</b>	<p>When you start to create elements in your model, the system automatically gives them a name based on the element type and the number of elements of that type that already exist. For example, if you create a series of Requirement elements the default names are <b>Requirement1</b>, <b>Requirement2</b>, <b>Requirement3</b> and so on. You then edit these default names.</p> <p>Alternatively, you can <b>define your own default naming and numbering system</b> for specific element types, for either the element name, the element alias, or both.</p>	<a href="#">Set Auto Naming and Auto Counters</a> <sup>[904]</sup>

Data Type/Option	Description	See also
	Click on this menu option to display the Auto Name Counters dialog, on which you define your naming and numbering system.	
<b>Extended Property Types</b>	<p>You can add Extended Properties (Tagged Values) to your elements in the Specification Manager, using the <b>Extended Properties</b> option on the element content menu.</p> <p>However, you might want to use a type of Tagged Value that is not currently available; in this case, you can create that type using the Tagged Value Types tab of the UML Types dialog.</p> <p>Click on this menu option to display the UML Types dialog.</p>	<p><a href="#">Editing Elements</a> <sup>[1740]</sup></p> <p><a href="#">Tagged Value Types</a> <sup>[1150]</sup></p>

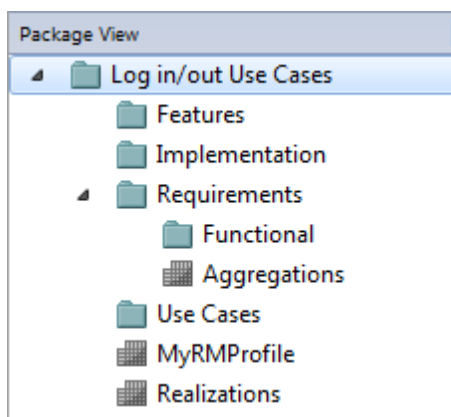
### Notes

- In the Corporate and extended editions of Enterprise Architect, a User Security system can be applied that restricts or enables access to a range of operations and functions; if you cannot access a function in the Specification Manager, check with your System Administrator or Security Administrator to see if you have access permissions to work with that function

### 8.1.7.3 Package Options

As you define the elements in your specification you can make them easier to reference by giving them more appropriate element names and/or numbers, either as the elements are created or retrospectively when a naming and numbering system has been finalized. You can also make the elements easier to locate and view by enabling level-numbering and indenting, helping you to organize and review the sequence and arrangement of the elements in the Specification Package. If you have been exploring other areas of the Project Browser, you can return focus to the Specification Package in the Project Browser.

You can also display a Package View panel to the left of the main view, showing the Package hierarchy of the **current** Specification Package. This also includes any Relationship Matrix profiles that exist in each Package.



**Access** **View | Specification Manager:**  <package name>

**View | Specification Manager:**  (for Package View)

Configure Package options

Option	Description	See also
<b>Locate Current in Project Browser</b>	Click on this option to highlight the name of the Specification Package in the Project Browser.	
<b>Apply Auto Naming to Elements</b>	<p>If you had to create elements before setting up an auto-naming and/or auto-numbering system, you can select this option to apply the system to the existing elements retrospectively.</p> <p>If you set up an auto-naming and/or auto-numbering system before creating your elements, that system is automatically applied as you create the elements.</p>	<a href="#">Apply Auto Naming to Existing Elements</a> <sup>[906]</sup>  <a href="#">Set Auto Naming and Auto Counters</a> <sup>[904]</sup>
<b>Level Numbering - Enable</b>	<p>Click on this option to apply level numbering to the elements in the Specification Package and indent each successive level, both in the Specification Manager and in the Project Browser. Nested elements are numbered hierarchically; that is:</p> <pre> 1 Name 2 Name   2.1 Name     2.1.1 Name     2.1.2 Name 3 </pre> <p>The option in the Specification Manager is the same as the option in the Project Browser Package context menu; turning the option on or off in one window also turns the option on or off in the other.</p>	<a href="#">Model Requirements</a> <sup>[1765]</sup>  <a href="#">Package Options in the Project Browser</a> <sup>[651]</sup>
<b>- Indent Description</b>	<p>If you have enabled level numbering, this option is available to indent the element <b>description</b> under each element name, to align with the element name. If the option is not selected, the text is aligned with the left-hand edge of the level number. For example:</p> <p><b>Option ON:</b></p> <pre> 1 Name   Description 2 Name   Description   2.1 Name     Description       2.1.1 Name         Description       2.1.2 Name         Description 3 Name   Description </pre> <p><b>Option OFF:</b></p>	



Option	Description	See also
	1 Name <i>Description</i> 2 Name <i>Description</i> 2.1 Name <i>Description</i> 2.1.1 Name <i>Description</i> 2.1.2 Name <i>Description</i> 3 Name <i>Description</i>	

### Package View panel

Option	Description	See also
<b>Expand or collapse the hierarchy</b>	Click on the white (expand) or black (collapse) arrows to the right of the Package name.	
<b>Locate a Package in Project Browser</b>	Right-click on the Package name and select the <b>Find in Project Browser</b> option.	
<b>Make a Package the focus of the display</b>	Click on the Package name in the hierarchy.	
<b>Make a Package the Root Specification Package</b>	Right-click on the Package name and select the <b>Set Package as Root</b> option. The display adjusts to show the selected Package at the top level of the hierarchy.	
<b>Add a new child Package</b>	Right-click on the Package name and select the <b>New Package</b> option. The New Model Package dialog displays, on which you start to define the new Package.	<a href="#">Add a Package</a> [772]
<b>Refresh the View</b>	Press <b>F5</b> .	
<b>Open Relationship</b>	Right-click on the Package name and select the <b>Open Relationship Matrix</b> option. A further submenu displays, from which you select to	<a href="#">Open Relationship</a>

Option	Description	See also
<b>Matrix for a Package</b>	make the Package the source, target or both in the Relationship Matrix.	<a href="#">Matrix</a> <sup>[1746]</sup>
<b>Enact a Relationship Matrix Profile</b>	Double-click on the Relationship Matrix profile name. It is enacted and displayed in the main View.	

### Notes

- In the Corporate and extended editions of Enterprise Architect, a User Security system can be applied that restricts or enables access to a range of operations and functions; if you cannot access a function in the Specification Manager, check with your System Administrator or Security Administrator to see if you have access permissions to work with that function

### Learn more

- [Specification Manager](#)<sup>[1728]</sup>
- [Select Specification Package](#)<sup>[1735]</sup>
- [Specification Manager Configuration](#)<sup>[1756]</sup>

## 8.2 Requirements

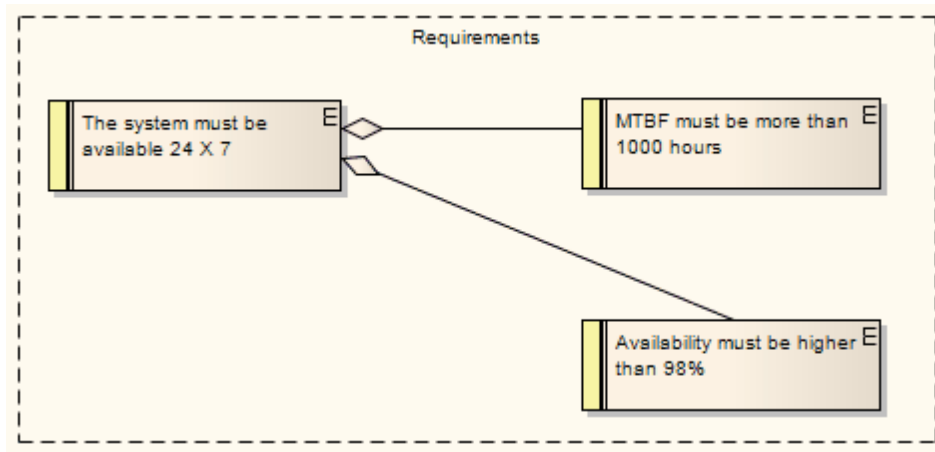
### Description

As an analysis step, often it is desirable to capture simple system requirements. These are eventually realized by Use Cases.

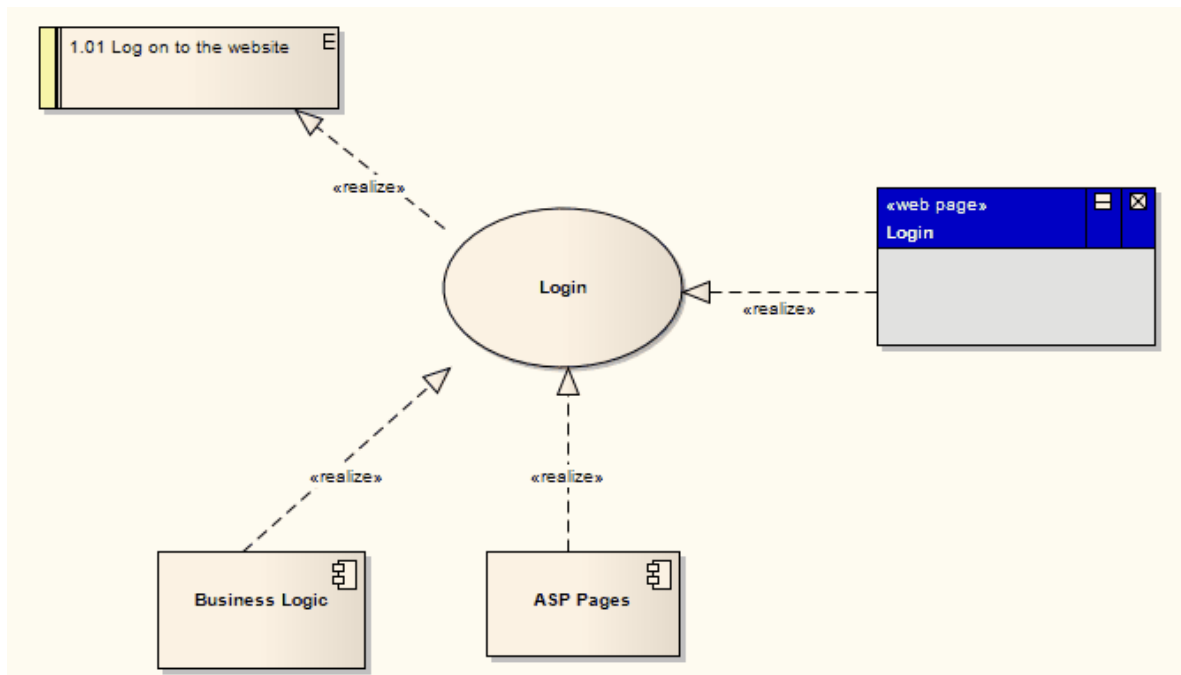
In the initial requirement gathering phase, cataloging requirements can be achieved using the Requirement extension on a Custom diagram.

### Examples

Requirements can be aggregated to create a hierarchy. The diagram below illustrates how this might be done.



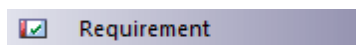
In the following diagram, a **requirement** that a user can log into a website is implemented by the *Login Use Case*, which in turn is implemented by the *Business Logic*, *ASP Pages* and *Login Web Page constructions*. Using this approach, you can easily model quite detailed and complex dependencies and implementation relationships.



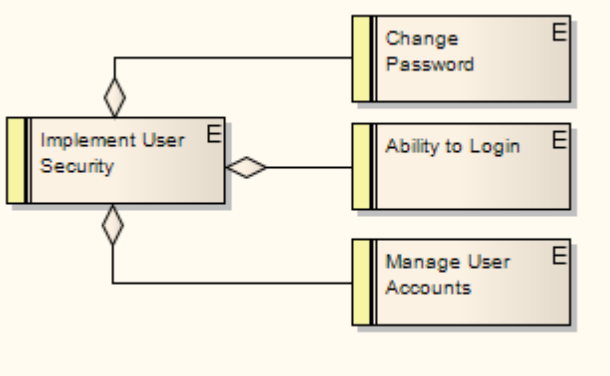
### Notes

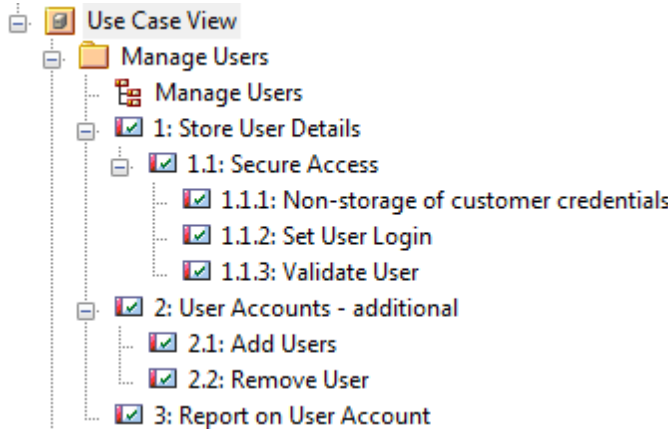
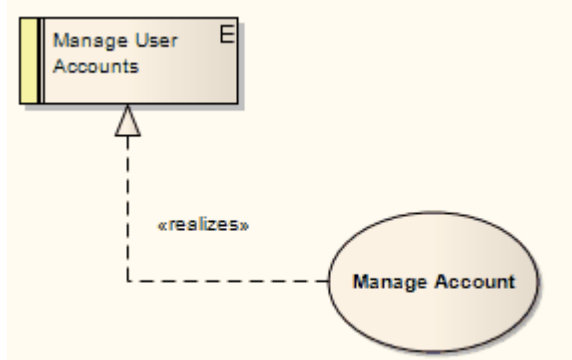
- External requirements can be displayed with or without an identifying **E** (for External) in the top right corner of the element; to toggle the display of this letter, select or deselect the **Show stereotype icon for requirements** checkbox on the Options dialog, Objects page
- The colors on Requirement elements identify the status of the requirement; you change the status - and hence color - on the element Properties dialog, and set the color for each status on the Status Types dialog

### Toolbox icon



## Topics

Topic	Detail	See also
<b>Represent Requirements</b>	<p>In Enterprise Architect, a requirement can be modeled as an:</p> <ul style="list-style-type: none"> <li>• <b>External Requirement</b> - an expectation of the system or process, what the system or process must provide, modeled as an element; for example, a business requirement or a stakeholder request - Requirements at this level have their own properties and are reported on separately in document reports</li> <li>• <b>Internal requirement</b> – a responsibility of an existing element, what the element must do or accomplish, defined as a property of the element</li> </ul> <p>Requirements Management in Enterprise Architect is primarily concerned with external Requirement elements and the elements that implement or realize them.</p>	<a href="#">Requirements</a> <a href="#">Internal requirements</a>
<b>Requirements in the Model</b>	<p>Requirement elements can be grouped and organized within <b>Requirements diagrams</b>.</p> <p>The Requirement elements are connected to each other by <i>Aggregate</i> relationships to form a hierarchy:</p>  <p>It is quite usual to develop a package of many hundreds of Requirement elements, arranged individually and in hierarchies of varying complexity. In the Project Browser you can use the <b>Advanced   Turn On Level Numbering</b> option to highlight the order and arrangement of the Requirements quickly and easily.</p> <p>The following illustration shows a number of Requirements in a package, where Level Numbering makes the order and arrangement clear:</p>	<a href="#">Requirements Diagram</a> <a href="#">Turn On Level Numbering</a>

Topic	Detail	See also
	 <p>If elements are added, moved or deleted from the package, the numbering automatically adjusts.</p> <p>This numbering can also be applied in the <b>document report generator</b> using the <b>LevelNumber</b> field in the <i>Element</i> section – { <i>Element.LevelNumber</i> }.</p>	<a href="#">Design Custom Document Templates</a> <sup>[2684]</sup>
Use Cases	<p>Requirements are implemented (realized) by model elements such as Use Cases, Classes, Interfaces and Components. There are many ways to <b>trace</b> either the Requirement for the feature or service modeled by the elements, or the elements that develop the requirement, most visibly in <b>Traceability</b> diagrams that depict the Requirements and the model elements <b>connected</b> by Realize relationships. The Realize connector enables members of the project team to keep design objectives and development in tandem, and the development path and purpose clear.</p>  <p>The more usual realization relationship is between a Requirement and a Use Case. A Requirement can be realized by one or more Use Cases, and a Use Case can realize one or more Requirements.</p> <p>Whilst a Requirement defines a condition that must be met, the <b>Use Case</b> is the key to defining and visualizing how that condition is met. A <b>Use Case diagram</b> depicts the logical grouping of actions, processes and components to achieve a required result, and through the use of <b>Actor</b> elements also defines the user and/or system roles</p>	<a href="#">Trace: Tracking Dependencies</a> <sup>[723]</sup> <a href="#">Example Traceability Diagram</a> <sup>[743]</sup> <a href="#">Connector Requirement</a> <sup>[1779]</sup> <a href="#">Use Case</a> <sup>[1352]</sup> <a href="#">Use Case Diagram</a> <sup>[1201]</sup> <a href="#">Actor</a> <sup>[1284]</sup> <a href="#">Composite Element</a> <sup>[936]</sup> <a href="#">Sequence Diagram</a> <sup>[1249]</sup> <a href="#">Communication Diagram</a> <sup>[1255]</sup> <a href="#">Activity Diagram</a> <sup>[1199]</sup> <a href="#">State Machine Diagram</a> <sup>[1203]</sup> <a href="#">Create a Rule Flow Activity</a> <sup>[1826]</sup>

Topic	Detail	See also
	<p>participating in the process.</p> <p>Each Use Case (as a <b>composite element</b>) can contain a combination of child diagrams that define in greater detail how a particular activity or facility might be implemented - such diagrams include <b>Sequence, Communication, Activity, State Machine</b> and <b>Business Rule Flow</b> diagrams. The actual implementation of each Use Case is realized by Class, Component and Interface elements organized in their own diagrams. These realizations can also be captured and viewed in Traceability diagrams, depicting the full development pathway from initial requirement through to testing and production.</p>	

### 8.3.1 Requirements Diagram









A **Requirements diagram** is a custom diagram used to describe a system's requirements or features as a visual model. Each Requirement is defined as a Requirement element (a Custom element of type Requirement). The actual Requirement, as a text explanation, is the element name (short) or description (long) in the element properties.

Requirement elements can have relationships with other elements, such as other Requirements, Use Cases and Components, to illustrate how a requirement is satisfied by modeling and development. You can track the development arising from a specification or requirement using the Traceability window.

**Example Diagram** [Example Requirements Diagram](#) <sup>[1768]</sup>

#### Tools

You can create Requirements diagram elements and connectors by dragging them onto the diagram from the Requirements pages of the Toolbox.

Requirements Diagram Elements	Requirements Diagram Connectors
 Package	 Aggregate
 Requirement	 Inheritance
 Feature	 Associate
 Object	 Implements

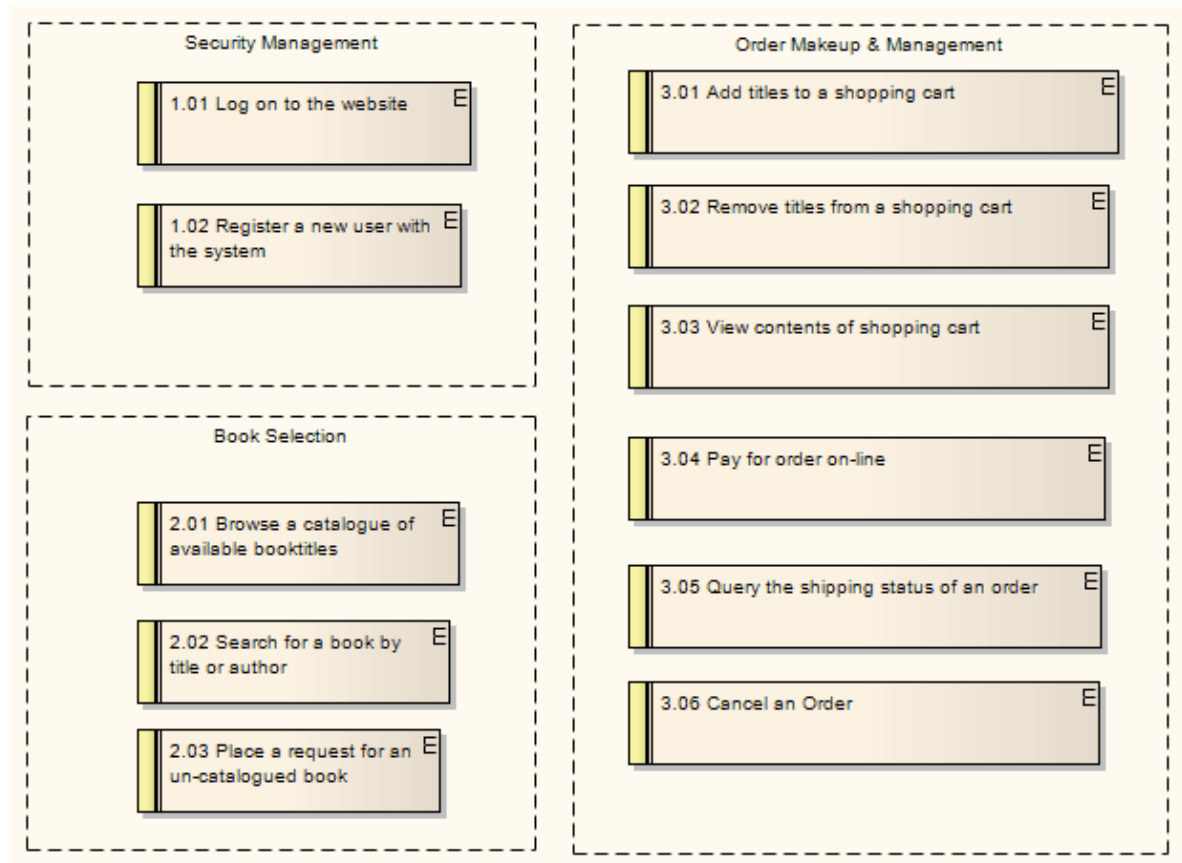
#### Learn more

- [Trace - Tracking Dependencies](#) <sup>[723]</sup>
- [The Traceability Window](#) <sup>[725]</sup>

### 8.3.1.1 Example Requirements Diagram

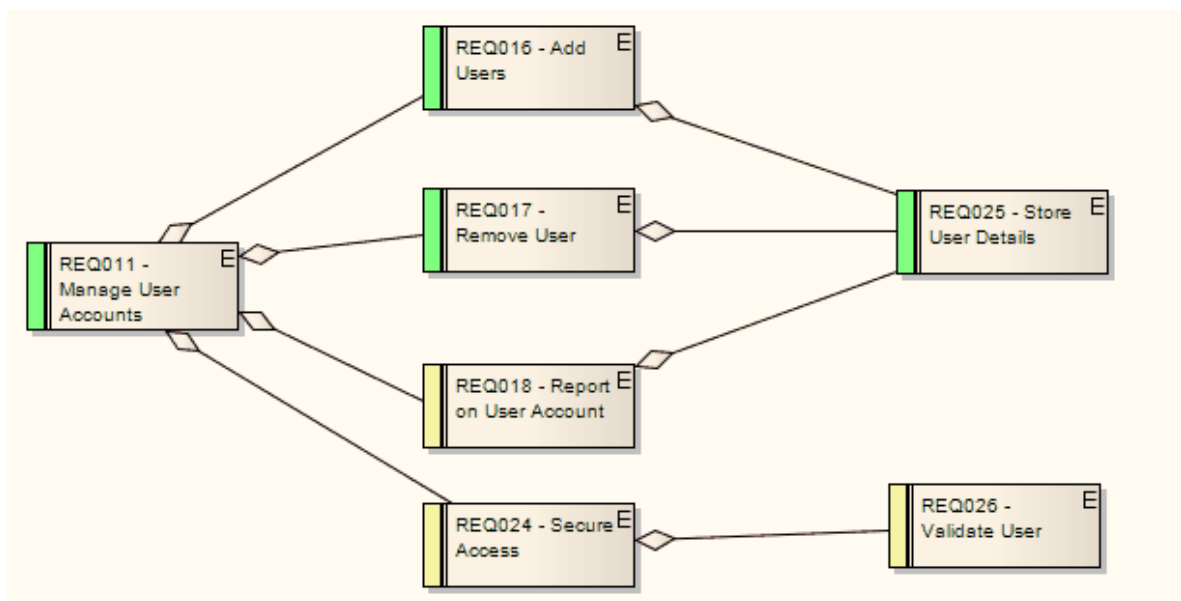
These examples illustrate possible structures of a **Requirements diagram**. Use Cases and Components in the system can be linked back to the Requirement elements to define how a particular system requirement is met.

#### Example 1



#### Example 2





### Notes

- Change and Defect (Issue) elements resemble Requirement elements and can be coded and managed in the same way

### Learn more

- [Requirements Diagram](#) <sup>[1767]</sup>
- [Requirements](#) <sup>[1763]</sup>
- [Use Case](#) <sup>[1352]</sup>
- [Component](#) <sup>[1370]</sup>
- [Changes](#) <sup>[2633]</sup>
- [Issues \(Defects\)](#) <sup>[2631]</sup>

## 8.3.2 Feature



### Description

A *Feature* is a small, granular function or characteristic expressed in client-valued terms as a satisfaction of a requirement; for example: 'context-sensitive Help', or 'ability to reverse-engineer VB.Net'.

Features are the primary requirements-gathering artifact of the Feature-Driven Design (FDD) methodology. They define the product feature that satisfies what a Requirement element has formalized as a contractual, testable, expected deliverable (for example: requirement - 'every element must provide context-sensitive Help'; feature - 'every element provides context-sensitive Help'). One Feature might realize one or more

Requirements, and one Requirement might be realized by more than one Feature.

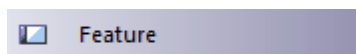
Features also have relationships with Use Cases. A Use Case defines the interaction a user has with the system in order to satisfy one or more Requirements. The Feature identifies the facility that provides the means for that interaction.

Feature elements are non-UML and are not related to the UML elements of the same name which are either BehavioralFeatures (operations, or methods) or StructuralFeatures (Ports, Parts and attributes).

Feature elements are available from the Requirements page of the Toolbox.

Feature elements can be displayed with or without an identifying **F** in the top right corner of the element. To toggle the display of this letter, select or deselect the **Show stereotype icon for requirements** checkbox on the Options dialog, Objects page.

#### Toolbox icon



#### Learn more

- [Feature-Driven Design \(FDD\) methodology](#)
- [Requirement Element](#)<sup>[1763]</sup>
- [Use Case Element](#)<sup>[1352]</sup>

### 8.3.3 Internal Requirements

#### Topics

Topic	Detail	See also
<b>Usage</b>	<p>Internal requirements in Enterprise Architect are element responsibilities. They are defined on the Requirements page of the element Properties dialog.</p> <p>Internal requirements form the functional requirements of the system to be built. The meaning of the requirement can vary depending on which element is the host; for example, a business process requirement might mean something different to a Use Case requirement, which again might mean something different to a Class requirement.</p> <p>For example, an internal responsibility to enable the user to login to the system has been defined for the Login Use Case. This is a responsibility of the Use Case - an action it is responsible for carrying out - and it applies only to this Use Case.</p> <p>The significant parameters (or, in Requirement Management terms, <i>attributes</i>) are the Type, Status, Difficulty and Priority. Whilst you can provide a detailed description of the responsibility in the <b>Notes</b> field, there is more scope in the name (<b>Requirement</b> field) to define the nature of the responsibility. An additional field, <b>Stability</b>, indicates the probability of the requirement changing; high stability means a low probability of change.</p>	<p><a href="#">Requirements</a><sup>[967]</sup></p> <p><a href="#">Make Internal Requirements External</a><sup>[1777]</sup></p> <p><a href="#">Scenarios &amp;</a></p>

Topic	Detail	See also
	<p>The example Use Case above also has connections to two external requirements, which are system functions that the Use Case implements either in full or in part. You can <b>convert</b> an internal responsibility into an external requirement.</p> <p>You can also create internal responsibilities for an element using the Scenarios &amp; Requirements window. A responsibility created in the window displays in the element Properties dialog, and vice versa.</p>	<a href="#">Requirements [992]</a>

### 8.3.3.1 Make Internal Requirement External

Elements in Enterprise Architect have internal requirements, or responsibilities (what they must do or accomplish). These can overlap or duplicate more formal requirements that the system in general must meet, so you might decide to make a single element's internal requirement into an external Requirement element (where the requirement can perhaps be implemented by multiple elements). You can make this conversion in one operation, using the **Move External** function.

#### Change an element's internal requirement into an external Requirement element

Step	Action	See also
1	Double-click on the element in a diagram or in the Project Browser. The element Properties dialog displays.	
2	Select the Requirements page.	
3	Locate and highlight the internal requirement.	
4	Click on the <b>Move External</b> button. The Browse Project dialog displays.	
5	Select the package to place the new requirement in.	
6	Click on the <b>OK</b> button.  A new Requirement element is created in the target package, with a Realization connector from the element to the Requirement.  In the Properties dialog, the requirement is now marked <i>external</i> and the dialog fields are grayed out. To edit its details, double-click on the requirement.	

#### Notes

- When an internal requirement is made into an external Requirement element, the **Stability** field and its value for the internal requirement are translated into the **Stability** Tagged Value in the external

Requirement

[Learn more](#)

- [Requirement Properties](#) 

## 8.4 Create Requirements

### Topics

Topic	Detail	See also
<b>Usage</b>	<p>Within Enterprise Architect you can create external Requirement elements in a number of ways, such as:</p> <ul style="list-style-type: none"> <li>• Typing or copy/pasting a text description into the body of the <b>Specification Manager</b></li> <li>• Dragging a <b>Requirement</b> icon from the Diagram Toolbox into a specific diagram</li> <li>• Generating an element within a specific package in the Project Browser</li> <li>• Dragging text from a text file onto a <b>diagram</b>, to generate a requirement based on that text - see below</li> <li>• Importing requirements from a spreadsheet application such as Excel, via CSV</li> <li>• Creating Requirement elements on the Package Browser or Diagram List for the selected package or diagram</li> <li>• Converting an internal element responsibility into an external element, in a selected target package</li> <li>• Importing requirements from another requirements management tool, such as <i>Telelogic DOORS</i> (in this case via the Sparx Systems <i>MDG Link For DOORS</i> integration tool)</li> </ul> <p>All methods that add a Requirement to a diagram or window also add the Requirement to the diagram's parent package in the Project Browser.</p>	<p><a href="#">Specification Manager</a> <sup>[1728]</sup></p> <p><a href="#">Requirements Toolbox</a> <sup>[814]</sup></p> <p><a href="#">Add Elements Directly To Packages</a> <sup>[903]</sup></p> <p><a href="#">Import Requirements using CSV</a> <sup>[1780]</sup></p> <p><a href="#">Package Browser</a> <sup>[673]</sup></p> <p><a href="#">Diagram List</a> <sup>[684]</sup></p> <p><a href="#">Converting an Internal Responsibility</a> <sup>[1771]</sup></p> <p><a href="#">Sparx Systems MDG Link For DOORS</a></p>

### Create Requirement elements from text

This procedure converts a text section heading into an element name and the section text into the element's Notes text. You can use this procedure to generate elements of a *range* of types; however, it is particularly useful for generating Requirements from a requirements specification document.

Step	Action	See also
1	Open a Requirements diagram in the Diagram View.	<a href="#">Requirements Diagram</a> <sup>[1767]</sup>
2	Open the document file containing the text you want to generate Requirement elements from (this can be opened in any common text editing tool).	
3	Highlight the required heading and associated text and drag them from the text file into the diagram.	<a href="#">Toolbox Shortcut</a>

Step	Action	See also
	The Toolbox Shortcut menu displays.	<a href="#">Menu</a> <sup>[799]</sup>
4	Navigate through the menus and select the required element type (in this case, click on <b>Common</b> and <b>Requirement</b> ).	
5	Enterprise Architect creates a (Requirement) element in the diagram, and displays the Properties dialog with the section heading in the <b>Name</b> (or equivalent) field and the text in the <b>Notes</b> field; the element is also added to the diagram's parent package.	

### Notes

- The Requirement element name can be simply descriptive text, with or without a manually-typed reference number; however, as requirements often have to have a unique reference for external checking, you can use the Enterprise Architect *auto-numbering* facility to automatically apply a numbering system with or without prefixes and suffixes - set the element type to **Requirement**
- External Requirement elements can be displayed on a diagram with or without an identifying **E** in the top right corner; to toggle display of this letter, select or deselect the **Show stereotype icon for requirements** checkbox on the Options dialog, Objects page
- Requirement elements can be color coded on a diagram to indicate their status

### Learn more

- [Object Display Options](#)<sup>[634]</sup>
- [Set Auto Naming and Auto Counters](#)<sup>[904]</sup>
- [Color Code External Requirements](#)<sup>[1776]</sup>

## 8.4.1 Requirement Properties

Requirement properties differ slightly from the properties of other elements; they include information related to the Type, Status, Difficulty and Priority of the Requirement. The **Notes** field is also important, as it describes precisely what requirement the element represents. Requirement naming can also require careful consideration and could reflect either a categorical naming convention, or simply a loose English description of the Requirement.

**Access**    **Requirement element context menu | Properties**

### Use to

- Document requirements
- Set Requirement features such as Type, Status and Priority
- Set other element properties common to both Requirements and other model elements

### Reference

Properties Window

	<p>the Properties window, and might not be shown in the drop-down list (if it is a stereotype and not a General Type)</p> <ul style="list-style-type: none"> <li>If you select a different value from the <b>Type</b> drop-down list, you change only the first of the multiple values (the one displayed in the field); you do not change any of the other multiple values, which remain set</li> </ul>	
<b>Phase</b>	The phase of this Requirement.	
<b>Version</b>	The version of this Requirement.	
<b>Last Update</b>	Read-only field specifying when this Requirement was last changed.	
<b>Created</b>	Read-only field specifying when this Requirement was first created.	
<b>Notes</b>	The description of this Requirement.	<a href="#">Note Editors</a> <sup>[1142]</sup> <a href="#">Linked Documents</a> <sup>[1044]</sup>

### Notes

- In Requirement Management tools and texts, the characteristics of a requirement are commonly called attributes; however, in UML the term *attribute* refers to a different type of feature, and the Requirement characteristics are defined as *properties* - in this Enterprise Architect documentation, the term *properties* is used
- In a project, it might be necessary to define more information in a Requirement than is provided by the standard properties; for more information on extending the Requirement properties, see the *Extend Requirement Properties* topic

### Learn more

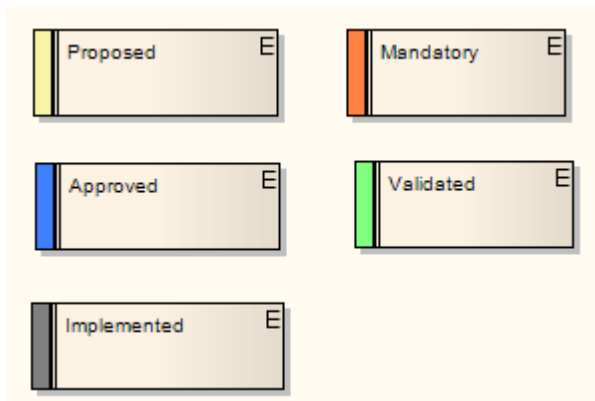
- [Extend Requirement Properties](#) <sup>[1777]</sup>

## 8.4.2 Color Code External Requirements

External requirements can be color coded on a diagram, to provide quick visual cues indicating the status of a requirement. The color code requirements use the following default conventions:

- Yellow for **Proposed**
- Blue for **Approved**
- Green for **Validated**
- Orange for **Mandatory**
- Black for **Implemented**





You can change these colors, and add or remove status types, using the Status Types dialog.

#### Enable color coded external requirements

Step	Action
1	Select the <b>Tools   Options</b> menu option. The Options dialog displays.
2	From the hierarchical tree select <b>Objects</b> , and select the <b>Show status colors on diagrams</b> checkbox to enable the status of external requirements to be represented by color coding.

#### Learn more

- [Status Types Dialog](#)<sup>[1159]</sup>

### 8.4.3 Extend Requirement Properties

#### Topics

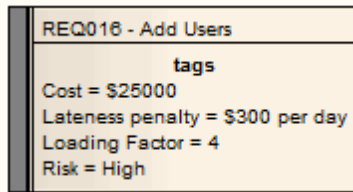
Topic	Detail	See also
<b>Introduction</b>	<p>A project might apply further properties to a requirement, such as cost, lateness penalty or risk to the business if not met. You can add these properties to specific Requirement elements, or configure them to be automatically available in all Requirement elements on creation, using <b>Tagged Values</b>. (These are sometimes referred to as User-defined attributes.)</p> <p>Extended element properties are not visible unless you open the Tagged Values window for the element. Alternatively, you can display</p>	<p><a href="#">Tagged Values</a><sup>[1134]</sup></p> <p><a href="#">Display Tagged Values on Diagrams</a><sup>[1778]</sup></p>

Topic	Detail	See also
	the additional properties on the element image on its diagrams.	
<b>Add Tagged Values to Existing Requirements</b>	<p>To add a property to a single Requirement as a Tagged Value, simply click on the Requirement, display the Tagged Values window ( <b>Ctrl+Shift+6</b> ), and enter the name of the property as the tag name and the value of the property as the tag value.</p> <p>It is likely that any property you add to one Requirement would also apply to others. You might therefore use a predefined Tagged Value Type to identify your Requirement property, so that you can select it whenever required. The predefined Tagged Value Type also enables you to define specific values for the Tagged Value.</p> <p>If the appropriate predefined Tagged Value Type does not exist, a Technology Developer can create it to add to the <b>structured tags</b>, <b>reference tags</b>, or <b>customized tags</b> collections.</p>	<a href="#">Assign a Tagged Value to an Item</a> <sup>[1137]</sup> <a href="#">Create Structured Tagged Values</a> <sup>[1622]</sup> <a href="#">Create Reference Date Tagged Values</a> <sup>[1628]</sup> <a href="#">Create a Custom Tagged Value Type</a> <sup>[1628]</sup>
<b>Configure Requirements to be Created With Extended Properties</b>	<p>If it is necessary to create all Requirements with the same extended set of properties, you can create a Requirement Template diagram and either create a special Requirement that defines those properties (as Tagged Values), or drag an existing Requirement with those properties onto the diagram. You then set the Requirement Template diagram as the template for all new Requirement elements, so that those new Requirements automatically have all of the properties you want.</p> <p>However, this then excludes other Requirement element formats, including the standard Requirement format. If you want to use another Requirement format, you have to replace or cancel the current Template. Alternatively, you can create a Profile.</p> <p>A Profile also defines exactly what a new Requirement element should contain, and how it should display in diagrams. However, a Profile is a collection of alternative element definitions, so it does not override the default Requirement format, nor does it prevent you from defining several different types of Requirement element. You can therefore have separate and parallel definitions of elements for business requirements, system requirements, project requirements, or any other category of requirement you decide to work with.</p> <p>For information on importing and using existing Profile files, see the <i>UML Profiles</i> topic. For information on creating new Profiles, see the <i>Developing Profiles</i> topic.</p>	<a href="#">Set Element Template Package</a> <sup>[929]</sup> <a href="#">Using UML Profiles</a> <sup>[1472]</sup> <a href="#">Developing Profiles</a> <sup>[1485]</sup>

#### 8.4.4 Display Tagged Values On Diagrams

##### Topics

Topic	Detail	See also
<b>Usage</b>	If you have extended the properties of a Requirement, you might want to make those properties visible in the Requirement elements in your diagrams, by switching on display of the element tags compartment.	

Topic	Detail	See also
	<p>You can do this in one of two ways:</p> <ul style="list-style-type: none"> <li>To display the <i>tags</i> compartment on all elements on a diagram, double-click on the diagram background and select the Elements tab of the Diagram Properties dialog; select the <b>Tags</b> checkbox and click on the <b>OK</b> button</li> <li>To display the <i>tags</i> compartment on a specific element on a diagram, right-click on the element and select the <b>Feature Visibility</b> context menu option; select the <b>Tags</b> checkbox in the Show Element Compartments panel of the Feature Visibility dialog, and click on the <b>OK</b> button</li> </ul> <p>The Tagged Values are then displayed in the Requirement element on the diagram.</p> 	

#### Learn more

- [Elements](#)<sup>[828]</sup>
- [Feature Visibility](#)<sup>[845]</sup>

## 8.4.5 Connect Requirements

#### Topics

Topic	Detail	See also
<b>Abstract</b>	<p>A Requirement element can be connected to other Requirements, most commonly using <b>Aggregate relationships</b> to form a hierarchy of requirements.</p> <p>Requirements are also connected to other types of element, most commonly Use Cases, by <b>Realize or Implements</b> relationships.</p> <p>These relationships are very important, both in identifying how the Requirements are organized and used in the model, and in <b>tracing</b> the development from the Requirements throughout the model. Both of these tasks are very simple in Enterprise Architect, because once a connector on a Requirement exists, Enterprise Architect automatically lists the Requirement in the:</p> <ul style="list-style-type: none"> <li>Requirements Traceability window (an important tool in examining the role of Requirements in the model)</li> <li>Specification Manager</li> </ul>	<a href="#">Aggregate relationships</a> <sup>[1392]</sup> <a href="#">Realization</a> <sup>[1440]</sup> <a href="#">Trace Requirements</a> <sup>[1784]</sup> <a href="#">Traceability</a> <sup>[725]</sup> <a href="#">Specification Manager</a> <sup>[1728]</sup> <a href="#">Requirements</a> <sup>[961]</sup> <a href="#">Requirement Properties</a> <sup>[1774]</sup> <a href="#">Scenarios &amp;</a>

Topic	Detail	See also
	<ul style="list-style-type: none"> <li>Requirements tab of the target element Properties dialog</li> <li>Links tab of the Requirement element Properties dialog</li> <li>Scenarios &amp; Requirements window</li> <li>Relationships Window</li> <li>Dependency and Implementation reports</li> <li>Standard document output</li> </ul> <p>The connector itself is also listed in the Links tab of the target element Properties dialog, and in the Relationship Matrix. There are, therefore, many ways to locate, view and track Requirement relationships.</p>	<a href="#">Requirements</a> <sup>[992]</sup> <a href="#">Relationships Window</a> <sup>[742]</sup> <a href="#">Dependency</a> <sup>[2739]</sup> <a href="#">Implementation reports</a> <sup>[2737]</sup> <a href="#">Document Reports</a> <sup>[2640]</sup> <a href="#">Links</a> <sup>[964]</sup> <a href="#">Relationship Matrix</a> <sup>[727]</sup>
<b>Connect On Diagram</b>	<p>Relationships can be created on a diagram by clicking on the appropriate connector icon from the Requirement and Common pages of the Toolbox, clicking on the source (originating) element, and dragging to the target element.</p> <p>If you are connecting elements in different packages, you can drag elements from the Project Browser onto a common diagram and set up the relationships there.</p>	<a href="#">Requirement Toolbox</a> <sup>[814]</sup> <a href="#">Common Page</a> <sup>[800]</sup>
<b>Quick Generation Of Realize Connector</b>	<p>You can quickly generate a Realize connector by dragging an existing Requirement element from the Project Browser into a diagram, over the element that implements the Requirement (usually a Use Case).</p> <p>Enterprise Architect interprets this as a request to create the Realize connector and does so automatically. The Requirement element is not added to the diagram. However, if you subsequently drag the Requirement onto the diagram the connector is already in place.</p>	
<b>Connect Off Diagram</b>	<p>You can also connect a Requirement element to other elements without necessarily having the elements on the same diagram, or having a diagram open.</p> <p>Use the Relationship Matrix to <b>create relationships</b> for requirements; this is a convenient way of quickly building up complex relationships and hierarchies.</p>	<a href="#">Creating Relationships</a> <sup>[737]</sup>

### 8.4.6 Import Requirements and Hierarchies in CSV

#### Topics

Topic	Detail	See also
<b>Usage</b>	You can import Requirements from a spreadsheet application in CSV format. Before doing this you must create a CSV import file specification that:	

Topic	Detail	See also
	<ul style="list-style-type: none"><li>• In the <b>Default Types</b> field has the value <b>requirement,package</b> to import requirements and a package structure to contain them</li><li>• Has the <b>Preserve Hierarchy</b> checkbox selected</li><li>• Identifies the data fields on the spreadsheet that are to be translated into Enterprise Architect, in the order in which they are plotted across the spreadsheet</li><li>• Is to operate on a spreadsheet containing the <b>CSV_KEY</b> and <b>CSV_PARENT_KEY</b> fields (which, if not generated by a CSV export from Enterprise Architect, you have added and populated yourself)</li></ul> <p>This enables you to import the individual and grouped requirements from the spreadsheet into Enterprise Architect, and to reconstruct the hierarchies of Requirements in the target package in the Project Browser.</p>	

**Learn more**

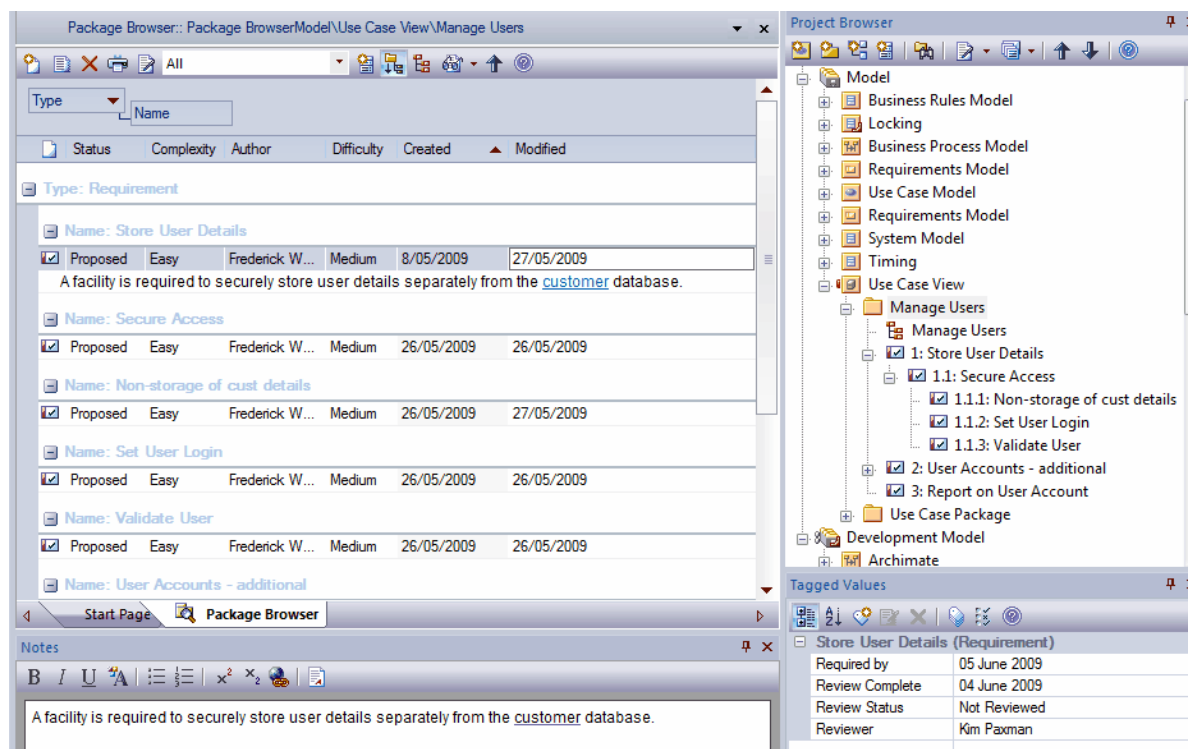
- [CSV Import](#) <sup>[503]</sup>
- [CSV Specifications](#) <sup>[498]</sup>
- [Using Preserve Hierarchy](#) <sup>[500]</sup>

## 8.5 Manage Requirements

One of the main advantages of managing Requirements in Enterprise Architect is that you can display many different aspects of information on the Requirements, from overall organization and location through lists, current status, general properties, detailed individual notes and specific properties, to relationships and dependencies with other elements in the model.

As most of these aspects are displayed in dockable windows, you can review the Requirements from several different perspectives simultaneously in the Enterprise Architect work area, as shown below:

### Example



This display shows the position of the *Store User Details* Requirement element in the model, and how it relates to other Requirements (Project Browser); the default characteristics of the Requirement (Package Browser) and the extended characteristics (Tagged Values window), and a detailed description of the Requirement (Notes window). You can configure some of these windows to display more information, and/or use other windows and facilities.

### Learn more

- [View Requirements](#) <sup>[1783]</sup>
- [Specification Manager](#) <sup>[1728]</sup>
- [Trace Use of Requirements](#) <sup>[1784]</sup>
- [Manage Requirement Changes](#) <sup>[1785]</sup>
- [Report on Requirements](#) <sup>[1787]</sup>

### 8.5.1 View Requirements

Use the following windows and facilities to: locate and list Requirement elements in the model; add, move and delete the elements; display and edit the properties and characteristics of individual elements; and generate reports on packages or specific elements.

#### Topics

Topic	Detail	See also
<b>Usage</b>	<p>Use the following windows and facilities to: locate and list Requirement elements in the model; add, move and delete the elements; display and edit the properties and characteristics of individual elements; and generate reports on packages or specific elements.</p> <ul style="list-style-type: none"> <li>• <b>Project Browser</b> - shows the content and structure of your model</li> <li>• <b>Specification Manager</b> - shows Requirements (and other element types) in a simple text format, and helps you to create and manage these elements</li> <li>• <b>Diagram List</b> - lists the elements in a diagram, filtered and sorted according to the settings you define; shows all or selected default properties of each element</li> <li>• <b>Package Browser</b> - lists the elements in a package, filtered and sorted according to the settings you define; shows all or selected default properties of each element</li> <li>• <b>(Requirements) Diagram</b> - shows the arrangement of a group of Requirements, and can show whether the elements are in the same package or different packages</li> <li>• <b>Model Search</b> - enables you to locate Requirements in general in the model, or specific Requirement elements, according to the search criteria you use</li> <li>• <b>Model Views</b> - enables you to maintain links to commonly-used elements, and to rapidly show developments and changes in (Requirement) package contents through either reports or slide shows of selected diagrams</li> <li>• <b>Properties</b> - shows every standard property of a selected element, whether updated by the user or maintained automatically by the system</li> <li>• <b>Tagged Values</b> - shows extended properties of a selected Requirement element</li> <li>• <b>Element Browser</b> - shows every added-on property, such as attributes, operations, Tagged Values and constraints</li> <li>• <b>Notes</b> - displays the detailed description of a requirement, and any other additional information recorded on the requirement</li> </ul>	<p><a href="#">Project Browser</a> <sup>[646]</sup></p> <p><a href="#">Specification Manager</a> <sup>[1728]</sup></p> <p><a href="#">Diagram List</a> <sup>[684]</sup></p> <p><a href="#">Package Browser</a> <sup>[673]</sup></p> <p><a href="#">Requirements Diagram</a> <sup>[1767]</sup> <sup>[784]</sup></p> <p><a href="#">Model Search</a> <sup>[700]</sup></p> <p><a href="#">Model Views</a> <sup>[686]</sup></p> <p><a href="#">Properties</a> <sup>[992]</sup></p> <p><a href="#">Tagged Values</a> <sup>[1134]</sup></p> <p><a href="#">Element Browser</a> <sup>[989]</sup></p> <p><a href="#">Notes</a> <sup>[1142]</sup></p>

## 8.5.2 Trace Use of Requirements

Having investigated the representation of requirements in your model, you could review either how they have been used to direct development through the model, or how a particular development was initiated. The windows and facilities you might use to follow development from Requirements are briefly described here; detailed information is also available in the *Traceability* topics.

The significant feature in tracing Requirements and development is the **connectors** between the elements.

### Facilities

Facility	Detail	See also
<b>Relationships Window</b>	Using the Relationships window, you can quickly identify every relationship of which a selected Requirement element is a member and the partner element in that relationship, whether or not the relationship is visible in the current diagram. If the partner element is not in the diagram, you have the option of adding it.	<a href="#">Relationships Window</a> <sup>[742]</sup>
<b>Traceability Window</b>	<p>The Traceability window is a very useful tool in showing chains of relationships that include the selected element. The window can show, for example, that:</p> <ul style="list-style-type: none"> <li>Requirement A is realized by a Use Case X, and</li> <li>Use Case X also realizes Requirement B, and</li> <li>Requirement B in turn is also realized by Use Case Y</li> </ul> <p>You can control the type and extent of these relationship chains. As the system checks the connectors and partner elements of every relationship within the limits you impose, if you specify broader limits the system can take some time to produce the final results.</p>	<a href="#">Traceability window</a> <sup>[725]</sup>
<b>Relationship Matrix</b>	The Relationship Matrix is a significant tool in mapping the relationships between the Requirements elements in a Package and other elements in either that Package or a different Package. Where a relationship is missing, you can add it; if an existing relationship is misplaced, you can delete it.	<a href="#">Relationship Matrix</a> <sup>[727]</sup>
<b>Requirements tab, Properties dialog</b>	On the Properties dialog for elements other than Requirements - particularly Use Cases - the Requirements tab shows all internal responsibilities and external Requirement elements attached to the element.	<a href="#">Properties Dialog</a> <sup>[956]</sup> <a href="#">Requirements</a> <sup>[967]</sup>
<b>Scenarios &amp; Requirements Window</b>	The Scenarios & Requirements window - as for the Properties dialog - shows the Requirements and responsibilities of the selected element, and the scenarios and constraints under which the Requirements are being realized.	<a href="#">Scenarios &amp; Requirements</a> <sup>[992]</sup>
<b>Validation</b>	It is useful to review the way that you have modeled your requirements, to check that they are correctly set up and connected to other elements. The Validation facility has a number of configuration	<a href="#">Model Validation</a> <sup>[2594]</sup>



Facility	Detail	See also
	options for validating various aspects of model development, as well as an option for specifically validating Requirements Management. This can reveal, for example, which of your Requirement elements do not yet have a Realization connector.	

#### Learn more

- [Traceability](#)<sup>[723]</sup>
- [Connect Requirements](#)<sup>[1779]</sup>

### 8.5.3 Manage Requirement Changes

Because requirements are statements of what a system or process must do or provide, they have a great impact on the modeling and development of the system. A new requirement might initiate an extensive program of work, and changes to or removal of that requirement can therefore have a major effect on the model. Issues concerning requirements, and changes to Requirement elements, must both be carefully managed.

The first steps in managing changes to requirements would be to raise **specific Issue** and **Change** request items against the Requirement element. You could monitor the appearance of these items using the filtered searches of the **Model Views**. You might then **review** the Requirement properties and/or its relationship hierarchies. During model development, you might capture periodic **Baselines** and use these to review the changes and, if necessary, roll them back to a previous point. You might also use the **Auditing** facility to monitor changes as they are made, and to ensure that no unauthorized or potentially risky changes are being made in the model.

#### Facilities

Facility	Detail	See also
<b>Changes and Issues</b>	<p>A change is, very broadly, an item defining an addition or alteration to a requirement. An issue identifies either a failure to meet a requirement, or a risk in meeting the requirement.</p> <p>Changes and issues can arise in development at a number of levels, being raised for problems that apply system-wide down to within a specific element. There are two mechanisms that can be used to identify a change or issue, and the work required to resolve it:</p> <ul style="list-style-type: none"> <li>• <b>Change and Issue</b> (or Defect) elements - structured comments that identify a problem at system-level, although they can also be attached to a specific element from which a problem arises. Both types of element resemble the Requirement element, and can be linked to one or more other elements that have to be reviewed, with relationships such as Association, Dependency and Realize. The two types of element can also form hierarchies or groups, where complex problems arise</li> <li>• <b>Maintenance items</b> raised against a specific element, and recorded for that element in the <b>Maintenance</b> window. Maintenance items enable the distinction between Defects (a failure to meet a requirement) and Issues (a risk factor that might affect satisfying the requirement). They also include Tasks, which record work items associated with the element</li> </ul>	<p><a href="#">Changes</a><sup>[2633]</sup></p> <p><a href="#">Issues (Defects)</a><sup>[2631]</sup></p> <p><a href="#">Maintenance</a><sup>[2619]</sup></p> <p><a href="#">Working on Maintenance Items</a><sup>[2623]</sup></p> <p><a href="#">Create an Element From a Maintenance Item</a><sup>[2629]</sup></p>

Facility	Detail	See also
	Maintenance items are very specific, but if there is a possibility of an item having a wider impact on other elements or the system in general, you can <b>translate the item</b> into a Change or Issue element, or any other type of element that best identifies the problem and its solution.	
<b>Model Views</b>	<b>Model Views</b> are very useful for trapping changes and issues in the model, especially on Requirements. You can set up searches to identify the appearance of new Change or Issue elements, or to detect changes in the properties of the Requirement elements themselves.	<a href="#">Model Views</a> <sup>[686]</sup>
<b>Baselines</b>	<p>A <b>Baseline</b> is a snapshot of a package or a model branch at a particular point in time, which you determine. You can use the Baseline as a distribution mechanism for changes to the model, but the main use is to enable you to compare the current model with a previous stage, and detect what changes have been made since the Baseline was captured.</p> <p>If you do not want a change to remain in the model, you can roll the affected elements back to the state they had in the Baseline. Therefore, if you maintain your requirements in a specific package or branch, you can capture Baselines of the package and ensure that changes conform to your change management process or, if not, can be reversed.</p>	<a href="#">Baselines</a> <sup>[459]</sup>
<b>Auditing</b>	<p>The <b>Auditing</b> facility enables you to capture any changes made to your model within the selection criteria that you define. You can, for example, configure the Auditing facility to specifically record <b>changes to Requirement</b> elements.</p> <p>As auditing is continuously monitoring, you can detect changes as they are made, and verify that they are acceptable. You can also store the log of changes, and review it later on.</p> <p>Note that you cannot reverse the changes automatically, as you can with Baselines. You might therefore use Auditing to identify changes to investigate more fully and - if necessary - reverse in a Baseline comparison.</p>	<a href="#">Auditing</a> <sup>[446]</sup> <a href="#">Auditing Settings</a> <sup>[448]</sup>

#### Learn more

- [View Requirements](#) <sup>[1783]</sup>

## 8.5.4 Report on Requirements

### Topics

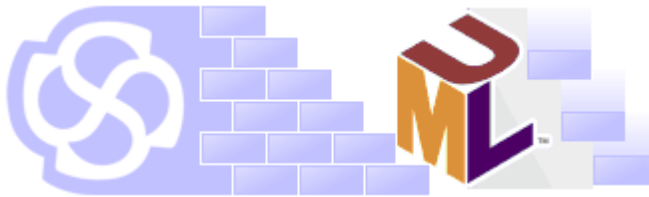
Topic	Detail	See also
<b>Usage</b>	<p>Enterprise Architect provides two report generation facilities that enable you to output document reports and web reports on your model structure and components.</p> <p>The <b>document reporting facility</b> is especially comprehensive, and contains a number of features that provide particular support to Requirements Management:</p> <ul style="list-style-type: none"> <li>• A <b>requirements report template</b> that extracts details of external requirements in the model; you can copy and tailor this template for your particular requirements</li> <li>• Options in the Specification Manager, Diagram List, Package Browser and Model Search to generate reports on selected (Requirement) items from the collected information</li> <li>• The <b>Implementation Report</b>, which lists for a selected package the elements that require implementers, together with any source elements in Realize (Implements) relationships with those elements</li> <li>• The <b>Dependency Report</b>, which lists for a selected package any elements that are dependent on another element for their specification; for example, a Use Case derives its specification from the Requirement that it realizes</li> </ul>	<p><a href="#">Document Reports</a> <sup>[2640]</sup></p> <p><a href="#">Generate Documentation</a> <sup>[2644]</sup></p> <p><a href="#">Specification Manager</a> <sup>[1728]</sup></p> <p><a href="#">Diagram List</a> <sup>[684]</sup></p> <p><a href="#">Package Browser</a> <sup>[673]</sup></p> <p><a href="#">Model Search</a> <sup>[700]</sup></p> <p><a href="#">Implementation Report</a> <sup>[2737]</sup></p> <p><a href="#">Realization</a> <sup>[1440]</sup></p> <p><a href="#">Dependency Report</a> <sup>[2739]</sup></p>

**Part**

---

**IX**

## 9 Domain Based Models












Enterprise Architect provides specific modeling tools for a range of specialized model types, as outlined below.

Enterprise Architect also provides an additional set of extended diagrams and a set of inbuilt and extension stereotype elements for more general use, and supports you in developing your own modeling languages.

Topic	Detail	See also
<b>ODM</b>	Enterprise Architect enables you to develop large-scale ontologies within the fully-integrated modeling environment, for your project domain.  ODM helps you to develop a formalized representation of business semantics and taxonomies, and a knowledge representation based on those formalizations.	<a href="#">MDG Technology for ODM</a> <sup>[1971]</sup>
<b>Requirements</b>	Enterprise Architect is one of the few UML tools that integrate Requirements Management with other software development disciplines in the core product, by defining requirements within the model.	<a href="#">Requirement Models</a> <sup>[1726]</sup>
<b>Business Modeling</b>	Modeling the business process is an essential part of any software development process, enabling the analyst to capture the broad outline and procedures that govern what it is a business does.	<a href="#">Business modeling</a> <sup>[1807]</sup>
<b>Business Rules</b>	Business Rule modeling captures the rules that govern a business, and their relationships with the entities and specific tasks within the organization or system.	<a href="#">Modeling Business Rules</a> <sup>[1814]</sup>
<b>BPMN</b>	The Business Process Modeling Notation is specifically targeted at the business modeling community and has a direct mapping to UML through BPMN Profiles; these profiles enable you to develop BPMN diagrams quickly and simply.	<a href="#">BPMN modeling</a> <sup>[1845]</sup>
<b>BPEL</b>	Business Process Execution Language is an executable language for specifying interactions with Web Services.  Enterprise Architect uses the BPMN profile as a graphical front-end to capture BPEL Process descriptions.	<a href="#">BPEL modeling</a> <sup>[1870]</sup>

<b>SysML</b>	SysML is a general-purpose graphical modeling language for specifying, analyzing, designing, and verifying complex systems that might include hardware, software, information, personnel, procedures and facilities.	<a href="#">Systems Engineering (SysML)</a> <sup>[2288]</sup>
<b>Data Modeling</b>	Enterprise Architect provides easy-to-use tools for building and maintaining all of the fundamental data models - Conceptual, Logical and Physical; because Enterprise Architect lets you visualize each type of data model in the same repository, you can easily manage dependencies between each level of abstraction.	<a href="#">Data Models</a> <sup>[1937]</sup>
<b>XSD</b>	Enterprise Architect supports rapid modeling, forward engineering and reverse engineering of W3C XML schemas (XSD), critical for the development of a complete Service Oriented Architecture (SOA).	<a href="#">XML Schema modeling (XSD)</a> <sup>[2387]</sup>
<b>WSDL</b>	Enterprise Architect enables rapid modeling, forward engineering and reverse engineering of W3C XML Web Service Definition Language (WSDL), critical for the development of a complete Service Oriented Architecture (SOA).	<a href="#">Web Service modeling (WSDL)</a> <sup>[2423]</sup>
<b>SPEM</b>	The Software and Systems Process Engineering Meta-model (SPEM) is a conceptual framework for modeling, documenting, presenting, managing, interchanging, and enacting development methods and processes.  SPEM 2.0 focuses on providing the additional information structures that you require for processes modeled with UML 2 Activities or BPMN/BPDM.	<a href="#">Software Process modeling (SPEM)</a> <sup>[1919]</sup>
<b>ArchiMate</b>	ArchiMate is an open-standard enterprise architecture language based on the IEEE 1471 standard, providing a common language for describing the construction and operation of business processes, organizational structures, information flows, IT systems and technical infrastructure.  It enables Enterprise Architects to clearly describe, analyse and visualize the relationships among business domains.	<a href="#">ArchiMate</a> <sup>[1926]</sup>
<b>ArcGIS</b>	ArcGIS is a suite of Geographic Information Systems (GIS) software products developed by Esri.	<a href="#">Geodatabase Design for ArcGIS</a> <sup>[1944]</sup>
<b>Data Flow Diagrams</b>	A data flow diagram (DFD) is a graphical representation of the flow of data through an information system, and can also be used to visualize data processing (structured design).  Developing a DFD helps in identifying the transaction data	<a href="#">Data Flow Diagrams</a> <sup>[1797]</sup>









	in the data model.	
<b>Entity Relationship Diagrams</b>	<p>Entity-relationship modeling is an abstract and conceptual database modeling method, used to produce a schema or semantic data model of, for example, a relational database and its requirements, visualized in Entity-Relationship Diagrams (ERDs).</p> <p>ERDs in Enterprise Architect assist you in building conceptual data models through to generating Data Definition Language (DDL) for the target DBMS.</p>	<a href="#">Entity Relationship Diagrams</a> <sup>[1939]</sup>
<b>Eriksson-Penker Extensions</b>	<p>Eriksson-Penker extensions provide a framework for UML business processing model extensions, to which an Enterprise Architect can add stereotypes and properties appropriate to their business..</p> <p>In Enterprise Architect, the Eriksson-Penker profile provides, through a set of stereotypes, a unique and powerful means of visualizing and communicating business processes and the necessary flow of information within an organization.</p>	<a href="#">Eriksson-Penker Extensions</a> <sup>[1929]</sup>
<b>Group of Four Patterns</b>	<p>Gang of Four (GoF) Patterns are 23 classic software design patterns providing recurring solutions to common problems in software design.</p> <p>Enterprise Architect provides each pattern through an icon in the Diagram Toolbox.</p>	<a href="#">GoF Patterns</a> <sup>[2285]</sup>
<b>ICONIX</b>	<p>The ICONIX Process is a streamlined approach to Use Case driven UML modeling that uses a core subset of UML diagrams and techniques to provide thorough coverage of object-oriented analysis and design.</p> <p>Its main activity is robustness analysis, a method for bridging the gap between analysis and design.</p>	<a href="#">ICONIX</a> <sup>[2282]</sup>
<b>Mind Mapping</b>	<p>A Mind Map is an image-centered diagram used to represent semantic or other connections between words, ideas, tasks or other items arranged radially around a central key word or idea.</p> <p>A Mind Map is used to generate, visualize, structure and classify ideas, and as an aid in study, organization, problem solving, decision making, and writing.</p>	<a href="#">Mind Mapping</a> <sup>[1794]</sup>
<b>SoaML</b>	Service Oriented Architecture (SOA) is an architectural paradigm for defining how people, organizations and systems provide and use services to achieve results.	<a href="#">SoaML</a> <sup>[2449]</sup>
<b>SOMF</b>	The service-oriented modeling framework (SOMF) is a	<a href="#">SOMF 2.1</a> <sup>[2454]</sup>

	service-oriented development life cycle methodology, offering a number of modeling practices and disciplines that contribute to a successful service-oriented life cycle management and modeling.	
<b>Extended Diagrams</b>	<p>Enterprise Architect provides an additional set of diagram types that extend the core UML diagrams for domain-specific models.</p> <p>Also, the specialized modeling tools listed in the first part of this table each have their own specialized diagrams.</p>	<a href="#">Analysis Diagram</a>  1801 <a href="#">Custom Diagram</a>  1796 <a href="#">Requirements Diagram</a>  1767 <a href="#">Maintenance Diagram</a>  2634 <a href="#">User Interface Diagram</a>  1990 <a href="#">Database Diagram</a>  1943 <a href="#">Business Modeling and Business Interaction Diagrams</a>  1805
<b>Inbuilt and Extension Stereotypes</b>	Behavioral and Structural elements can be extended through the use of stereotypes; Enterprise Architect provides a number of inbuilt extensions.	<a href="#">Inbuilt and Extension Stereotypes</a>  1997
<b>Build Your Own Modeling Language</b>	Enterprise Architect enables you to extend the scope both of your modeling and of the UML components you use, through the use of stereotypes, profiles and patterns to develop your own modeling applications.	<a href="#">Defining a Modeling Language</a>  1483



## 9.1 Domain Based Diagrams

In addition to diagrams defined by the UML, Enterprise Architect provides some extended diagram platforms to model business processes or develop custom diagrams.

Diagram Type	Detail	See also
<b>Analysis Diagram</b>	An Analysis diagram is a simplified Activity diagram, which is used to capture high level business processes and early models of system behavior and elements.	<a href="#">Analysis Diagram</a> 
<b>Custom Diagram</b>	A Custom diagram is an extended Class diagram that is used to capture requirements, user interfaces or custom-design models.	<a href="#">Custom Diagram</a> 
<b>Requirements Diagram</b>	A Requirements diagram is a custom diagram used to describe a system's requirements or features as a visual model.	<a href="#">Requirements Diagram</a> 
<b>Maintenance Diagram</b>	A Maintenance diagram is a custom diagram used to describe change requests and issue items within a system model.	<a href="#">Maintenance Diagram</a> 
<b>User Interface Diagram</b>	User Interface diagrams are custom diagrams used to visually mock-up a system's user interface using forms, controls and labels.	<a href="#">User Interface Diagram</a> 
<b>Data Modeling Diagram</b>	A Data Modeling diagram is a Class diagram used for representing database schemas.	<a href="#">Data Modeling Diagram</a> 
<b>Documentation</b>	Virtual documents enable you to structure and filter your document and web reports by selecting, grouping and ordering individual packages independent of the organization of the Project Browser.	<a href="#">Virtual Documents</a> 
<b>Business Modeling and Business Interaction</b>	Business Modeling diagrams and Business Interaction diagrams enable you to model both the structure and behavior of a business system.  Business Modeling diagrams are based on a Class (UML Structural) diagram, whilst Business Interaction diagrams are based on a Sequence (UML Behavioral) diagram.	<a href="#">Business Modeling and Business Interaction</a> 

### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Modeling Languages**

## 9.2 Analysis Models

Enterprise Architect provides specific modeling tools for analysis modeling.

Analysis Tool	Detail	See also
<b>Data Flow Diagrams</b>	<p>A Data Flow Diagram (DFD) is a graphical representation of the flow of data through an information system; it can also be used to visualize data processing (structured design).</p> <p>Developing a DFD helps in identifying the transaction data in the data model.</p>	<a href="#">Data Flow Diagrams</a> <sup>[1797]</sup>
<b>Custom Models</b>	<p>Custom models provide a number of extensions to the UML model and help you to perform exploratory and non-rigorous experimentation with model elements and diagrams. For example, using a Custom diagram you can model requirements, user interfaces or custom designs.</p>	<a href="#">Custom Diagram</a> <sup>[1796]</sup>
<b>Mind Mapping</b>	<p>A Mind Map is an image-centered diagram used to represent semantic or other connections between words, ideas, tasks or other items arranged radially around a central key word or idea.</p> <p>A Mind Map is used to generate, visualize, structure and classify ideas, and as an aid in study, organization, problem solving, decision making, and writing.</p>	<a href="#">Mind Mapping</a> <sup>[1794]</sup>

### Learn more

- [Analysis Diagram](#) <sup>[1801]</sup>
- [Analysis Stereotypes](#) <sup>[1800]</sup>
- [Process](#) <sup>[2008]</sup>

### 9.2.1 Mind Mapping

The following text is derived from the **Mind Map** entry in the online Wikipedia.

*A Mind Map is a diagram used to represent words, ideas, tasks or other items linked to and arranged radially around a central key word or idea. It is used to generate, visualize, structure and classify ideas, and as an aid in study, organization, problem solving, decision making, and writing.*

*A Mind Map is an image-centered diagram that represents semantic or other connections between portions of information. By presenting these connections in a radial, non-linear graphical manner, it encourages a brainstorming approach to any given organizational task, eliminating the hurdle of initially establishing an intrinsically appropriate or relevant conceptual framework to work within.*

*The elements are arranged intuitively according to the importance of the concepts and are organized into groupings, branches, or areas. The uniform graphic formulation of the semantic structure of information on the method of gathering knowledge, may aid recall of existing memories.*

The use of the term Mind Maps is trademarked in the UK and the USA by The Buzan Organization, Ltd.

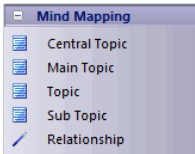
### Mind Mapping in Enterprise Architect

Within Enterprise Architect you can develop Mind Maps quickly and simply, through use of an MDG Technology integrated with the Enterprise Architect installer. The Mind Mapping facilities are provided in the form of:

- A Mind Mapping **diagram type**, accessed through the New Diagram dialog
- A Mind Mapping **page in the Toolbox**
- Mind Mapping **element** and **relationship** entries in the **Toolbox Shortcut Menu** and **Quick Linker**

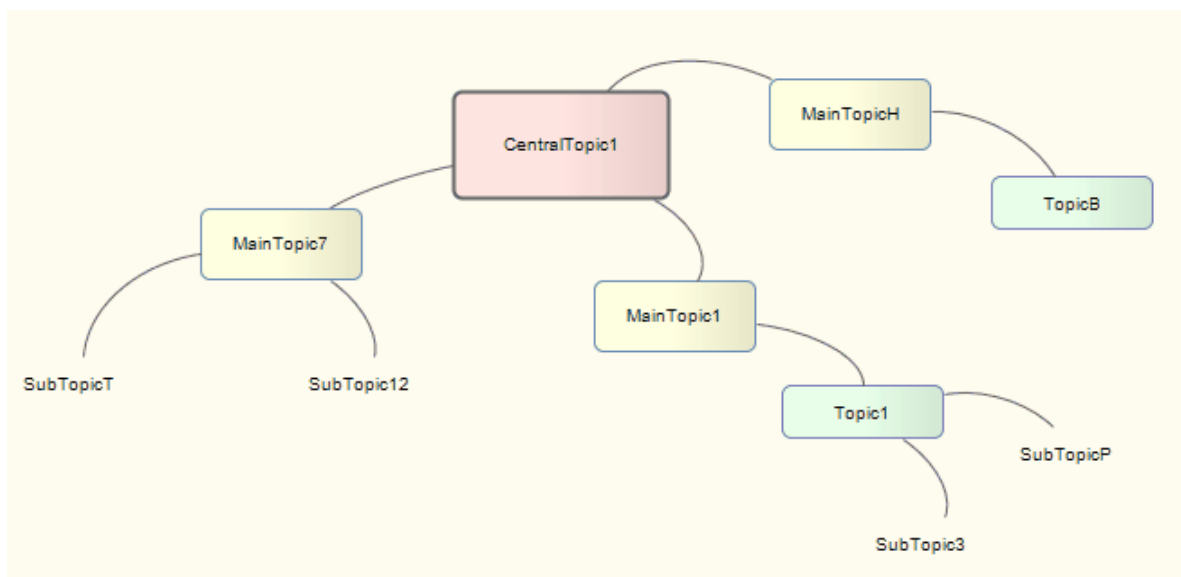
Access    **Diagram | Diagram Toolbox: More Tools | Mind Mapping**

#### Mind Mapping Toolbox Page

Page Appearance	Description	See also
	<ul style="list-style-type: none"><li>• <b>Central Topic</b> is the main theme of the Mind Map; you would normally have one or two of these on the diagram, but can add as many as are necessary</li><li>• <b>Main Topic</b> represents the immediate concepts generated by the Central Topic</li><li>• <b>Topic</b> represents the larger divisions of a Main Topic</li><li>• <b>Sub Topic</b> represents the finer divisions of a Topic or Main Topic; you could also have Subtopics of Subtopics to represent increasingly finer distinctions</li><li>• <b>Relationship</b> represents the connection between any two elements; you can have several Relationships per element</li></ul> <p>Each relationship has three anchor points, so you can curve the lines to develop the flow of concepts more easily</p>	

#### Mind Map Diagram

This is the general appearance of a Mind Mapping diagram.



As the elements can represent any concept, object or relationship, you can use the full range of element properties and features to expand on what the element represents, including adding Note elements. However, to preserve the simplicity and readability of the diagram itself, you cannot display the element compartments on the diagram.

### Disable Mind Mapping

If you prefer not to use Mind Mapping in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog (**Settings | MDG Technologies**).

### Learn more

- [Mind Map](#) (Online Resource)
- [New Diagram](#) <sup>[822]</sup>
- [Toolbox Shortcut](#) <sup>[799]</sup>
- [Quick Linker](#) <sup>[896]</sup>
- [MDG Technologies](#) <sup>[1477]</sup>

### Learning Center topics

- **(Alt+F1) | Enterprise Architect | Modeling Languages | Standard Profiles | Mind Mapping**














## 9.2.2 Custom Diagram

A **Custom diagram** is an extended Class diagram that is used to capture requirements, user interfaces or custom-design models.

Custom models provide a few extensions to the UML model and enable some exploratory and non-rigorous experimentation with model elements and diagrams.

### Tools

Select Custom diagram elements and connectors from the Custom pages of the Toolbox.

Custom Diagram Elements	Custom Diagram Connectors
 Package	 Associate
 Requirement	 Aggregate
 Issue	 Generalize
 Change	 Realize
 Screen	 Nesting
 UI Control	
 Test Case	
 Entity	

### 9.2.3 Data Flow Diagrams

The following text is derived from the **Data Flow Diagram** entry in the online Wikipedia.

*A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. A data flow diagram can also be used for the visualization of data processing (structured design). It is common practice for a designer to draw a context-level DFD first which shows the interaction between the system and outside entities. This context-level DFD is then "exploded" to show more detail of the system being modeled.*

*Data flow diagrams were invented by Larry Constantine ... based on Martin and Estrin's "data flow graph" model of computation. ( They ) are one of the three essential perspectives of Structured Systems Analysis and Design Method SSADM. The sponsor of a project and the end users will need to be briefed and consulted throughout all stages of a system's evolution. With a dataflow diagram, users are able to visualize how the system will operate, what the system will accomplish, and how the system will be implemented. The old system's dataflow diagrams can be drawn up and compared with the new system's dataflow diagrams to draw comparisons to implement a more efficient system.*

*Developing a DFD helps in identifying the transaction data in the data model.*

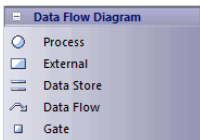
**[Access](#) [Diagram](#) | [Diagram Toolbox: More Tools](#) | [Data Flow Diagrams](#)**

**[Data Flow Diagrams in Enterprise Architect](#)**

Within Enterprise Architect, you can develop Data Flow diagrams quickly and simply through use of an MDG Technology integrated with the Enterprise Architect installer. The Data Flow diagram facilities are provided in the form of:

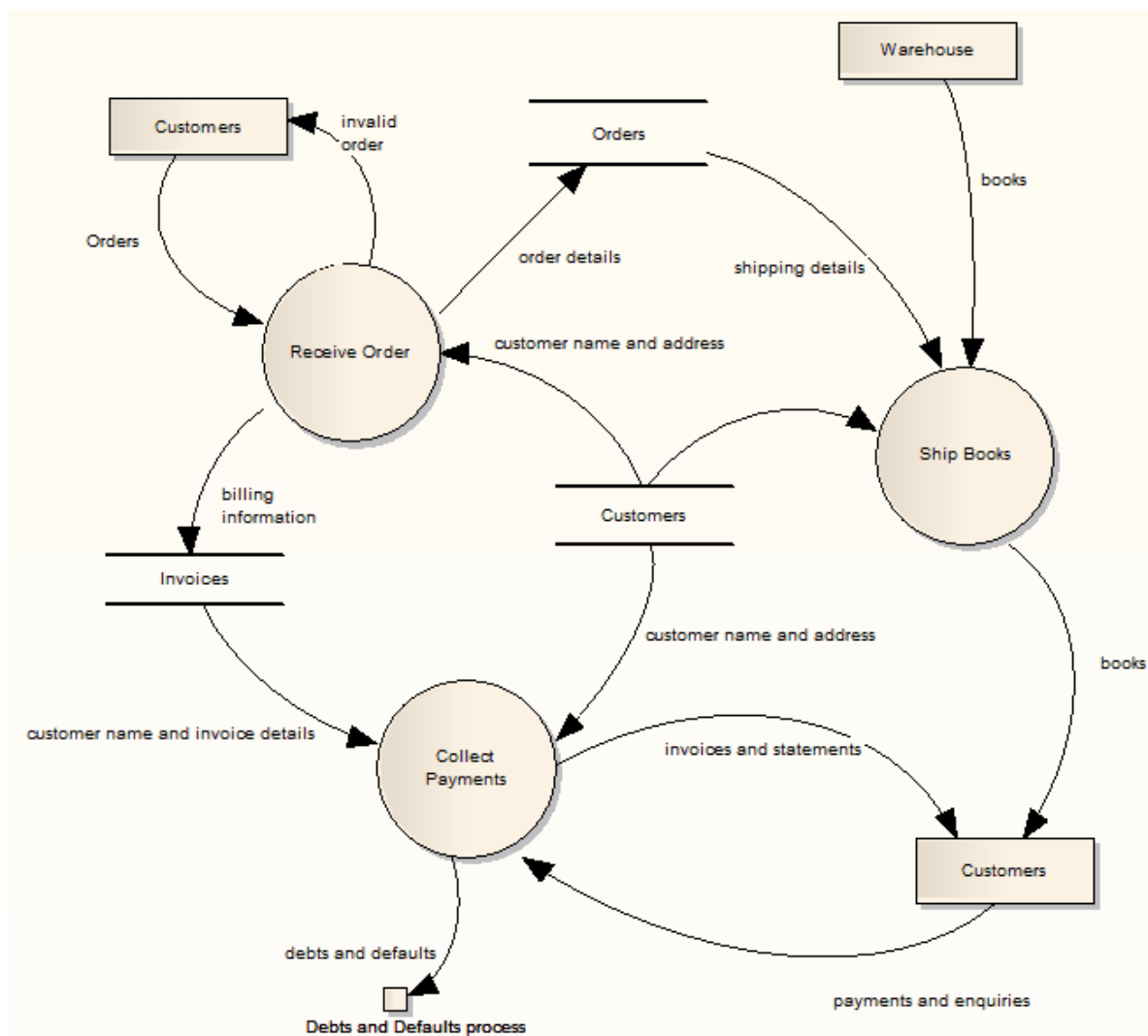
- A Data Flow diagram type, accessed through the New Diagram dialog
- A Data Flow Diagram page in the Toolbox
- Data Flow element and relationship entries in the Toolbox Shortcut Menu and Quick Linker

#### Data Flow Diagram Toolbox Page

Page Appearance	Description	See also
	<ul style="list-style-type: none"><li>• <b>Process</b> is a process or activity in which data is used or generated</li><li>• <b>External</b> represents an external source, user or repository of the data</li><li>• <b>Data Store</b> represents an internal physical or electronic repository of data, into and out of which data is stored and retrieved</li><li>• <b>Data Flow</b> (connector) represents how data flows through the system, in physical or electronic form</li><li>• <b>Gate</b> represents the termination point of incoming and outgoing messages on a lower level diagram (that is, messages to and from processes depicted elsewhere)</li></ul>	

#### Data Flow Diagram Appearance

When dragged onto a Data Flow diagram, the elements and relationships have the following appearances:



To preserve the simplicity and readability of the diagram, you cannot display the element compartments on the diagram.

### Context Diagram

A *Context* diagram is a top-level Data Flow diagram that has just one Process element representing the system being modeled, showing its relationship to external systems.

### Disable Data Flow Diagrams

If you prefer to not use Data Flow Diagramming in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog (**Settings | MDG Technologies**).

### Learn more

- [Data Flow Diagram](#) (Online Resource)
- [New Diagram](#) <sup>822</sup>

- [Toolbox Shortcut](#)<sup>[799]</sup>
- [Quick Linker](#)<sup>[896]</sup>
- [MDG Technologies](#)<sup>[1477]</sup>

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect** | **Modeling Languages** | **Standard Profiles** | **Data Flow Diagrams**

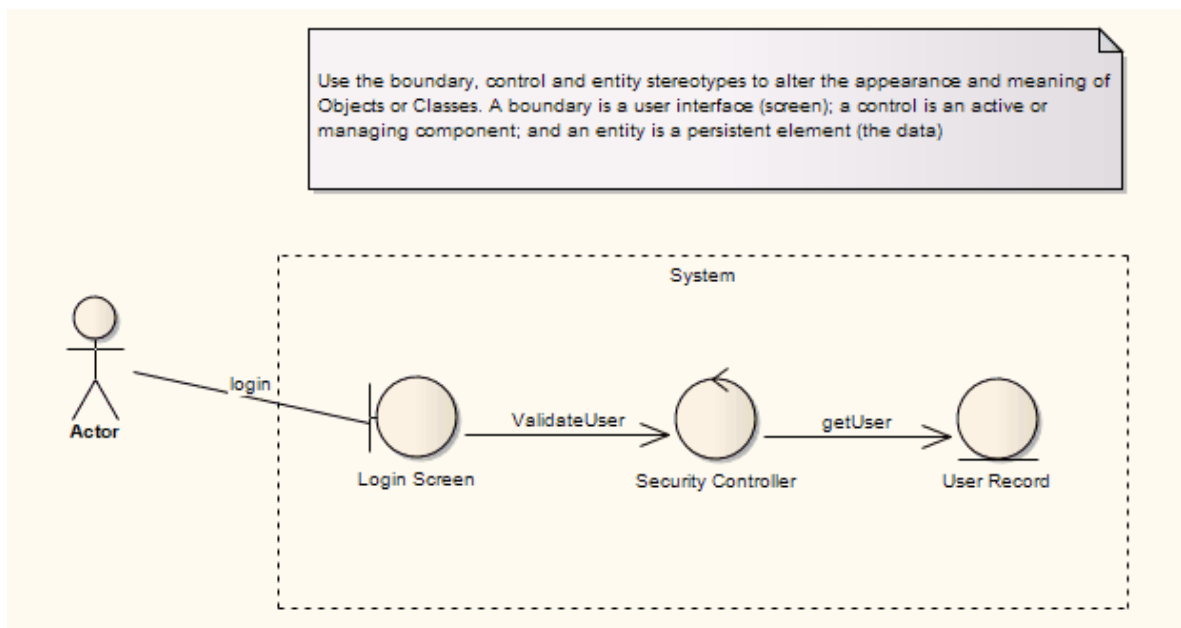
## 9.2.4 Analysis Stereotypes

Enterprise Architect provides some built in **stereotypes** that you can assign to an element during **analysis**. The effect of each of these stereotypes is to display a different **icon** from the normal **element** icon, giving a visual key to the element's purpose. The main types of inbuilt icons for elements created using the stereotypes include:

- **Boundary** - for a system boundary (for example, a Login screen)
- **Control** - to specify an element is a controller of some process (as in the **Model-View-Controller** pattern)
- **Entity** - the element is a persistent or data element

The elements created using these stereotypes are illustrated in the *Robustness* diagram below. Also see the **Business Modeling** elements, used in Business Modeling and Business Interaction diagrams.

#### Example of Analysis Stereotypes in use



#### Learn more

- [Boundary](#)<sup>[1997]</sup>
- [Control](#)<sup>[1999]</sup>
- [Entity](#)<sup>[2000]</sup>
- [Business Modeling](#)<sup>[1805]</sup>



Learning Center topics

- (Alt+F1) | [Enterprise Architect](#) | [Modeling Languages](#) | [Business](#) | [Business Modeling and Interactions](#)















## 9.2.5 Analysis Diagram






An **Analysis** diagram is a simplified **Activity** diagram, used to capture high level business processes and early models of system behavior and elements. It is less formal than some other diagrams, but provides a useful means of capturing the essential business characteristics and requirements. Robustness diagrams, used extensively in ICONIX, can be created as Analysis diagrams.

Example Diagram [Example Analysis Diagram](#) <sup>1802</sup>

Tools

Select Analysis diagram elements and connectors from the Analysis pages of the Toolbox.

Analysis Diagram Elements	Analysis Diagram Connectors
 Actor	 Information Flow
 Object	 Object Flow
 Process	 Associate
 Collaboration	 Realize
 Collaboration Use	 Representation
 Send	
 Receive	
 Information	
 Information Item	

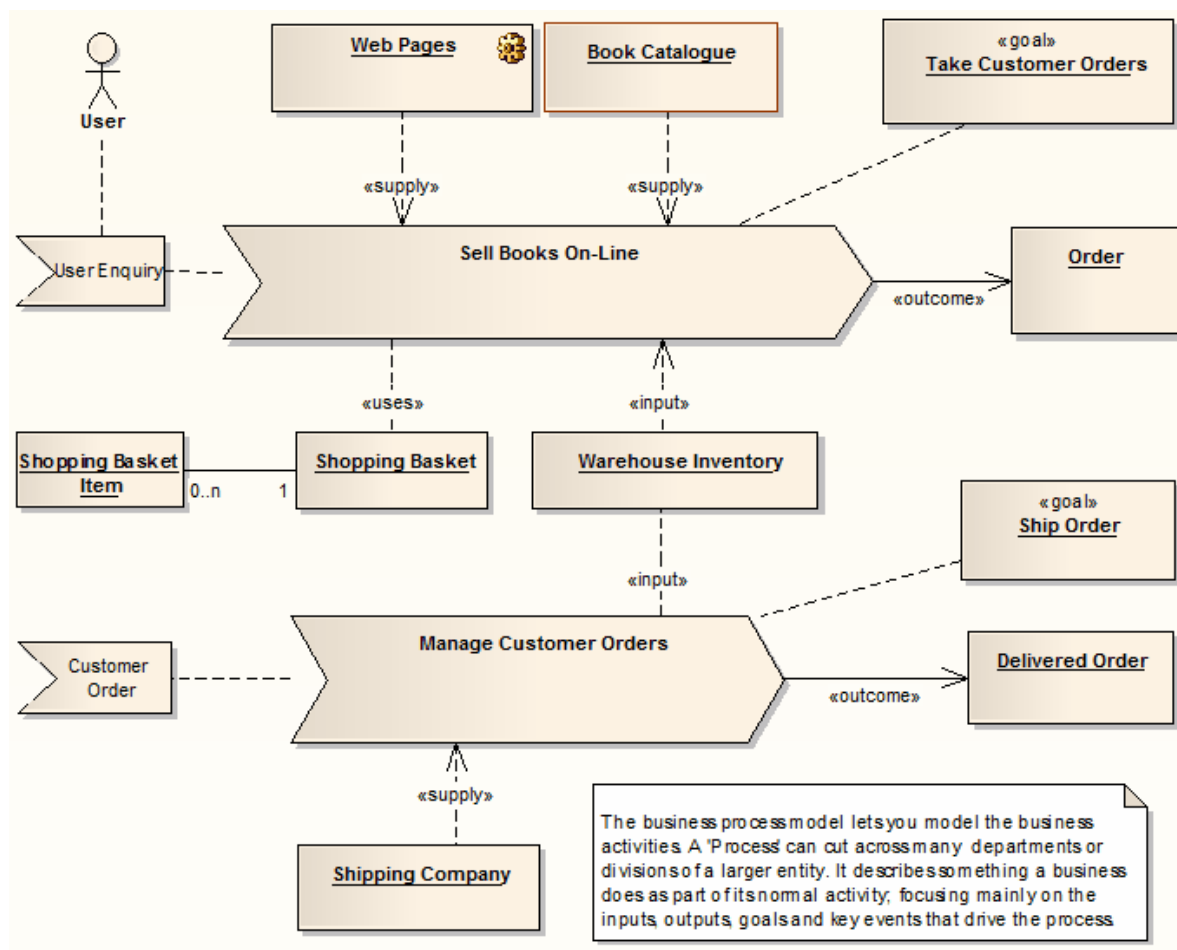
Analysis Diagram Elements	Analysis Diagram Connectors
 Decision	
 Merge	
 Boundary	
 Control	
 Entity	

#### Learn more

- [Activity Diagram](#) <sup>[1199]</sup>
- [Business Process Modeling](#) <sup>[1807]</sup>
- [ICONIX](#) <sup>[2282]</sup>

#### 9.2.5.1 Example Analysis Diagram

Enterprise Architect supports some of the **Eriksson-Penker Business Extensions** that facilitate business process modeling. You can also load the complete Eriksson-Penker Business Extensions **UML Profile** into Enterprise Architect to create detailed process models.



### Learn more

- [Eriksson-Penker Business Extensions](#) <sup>1929</sup>

## 9.3 Business Models

A **Business Process model** describes both the behavior and the information flows within an organization or system. As a model of business activity, it captures the significant events, inputs, resources, processing and outputs associated with relevant business processes.

Enterprise Architect provides specific modeling tools for a range of Business Modeling types.

Topic	Detail	See also
<b>Requirements</b>	Enterprise Architect is one of the few UML tools that integrate Requirements Management with other software development disciplines in the core product, by defining requirements within the model.	<a href="#">Requirements modeling</a> <sup>[1726]</sup>
<b>Business Modeling</b>	Modeling the business process is an essential part of any software development process, enabling the analyst to capture the broad outline and procedures that govern what it is a business does.	<a href="#">Business modeling</a> <sup>[1807]</sup> <a href="#">Business Modeling/Interaction</a> <sup>[1805]</sup>
<b>Business Rules</b>	Business Rule modeling captures the rules that govern a business, and their relationships with the entities and specific tasks within the organization or system.	<a href="#">Modeling Business Rules</a> <sup>[1814]</sup>
<b>BPMN</b>	The Business Process Modeling Notation is specifically targeted at the business modeling community and has a direct mapping to UML through BPMN Profiles; these profiles enable you to develop BPMN diagrams quickly and simply.	<a href="#">BPMN modeling</a> <sup>[1845]</sup>
<b>BPEL</b>	Business Process Execution Language is an executable language for specifying interactions with Web Services.  Enterprise Architect uses the BPMN profile as a graphical front-end to capture BPEL Process descriptions.	<a href="#">BPEL modeling</a> <sup>[1870]</sup>
<b>SPEM</b>	The Software and Systems Process Engineering Meta-model (SPEM) is a conceptual framework for modeling, documenting, presenting, managing, interchanging, and enacting development methods and processes.  SPEM 2.0 focuses on providing the additional information structures that you require for processes modeled with UML 2 Activities or BPMN/BPDM.	<a href="#">Software Process modeling (SPEM)</a> <sup>[1919]</sup>
<b>ArchiMate</b>	ArchiMate is an open-standard enterprise architecture language based on the IEEE 1471 standard, providing a common language for describing the construction and operation of business processes, organizational structures, information flows, IT systems and technical infrastructure.  It enables Enterprise Architects to clearly describe, analyse	<a href="#">ArchiMate</a> <sup>[1926]</sup>

Topic	Detail	See also
	and visualize the relationships among business domains.	
<b>Eriksson-Penker Extensions</b>	<p>Eriksson-Penker extensions provide a framework for UML business processing model extensions, to which an Enterprise Architect can add stereotypes and properties appropriate to their business.</p> <p>In Enterprise Architect, the Eriksson-Penker profile provides, through a set of stereotypes, a unique and powerful means of visualizing and communicating business processes and the necessary flow of information within an organization.</p>	<a href="#">Eriksson-Penker Extensions</a> <small>[1929]</small>

### 9.3.1 Business Modeling/Interaction

Business Modeling diagrams and Business Interaction diagrams enable you to model both the structure and behavior of a business system. Business Modeling diagrams are based on a Class (UML Structural) diagram, whilst Business Interaction diagrams are based on a Sequence (UML Behavioral) diagram. Both diagram types have the same default Toolbox, which consists of a Business Modeling element page. The available elements include stereotyped Objects, and a stereotyped Actor (Business Actor), Use Case (Business Use Case) and Collaboration (Business Use Case Realization).

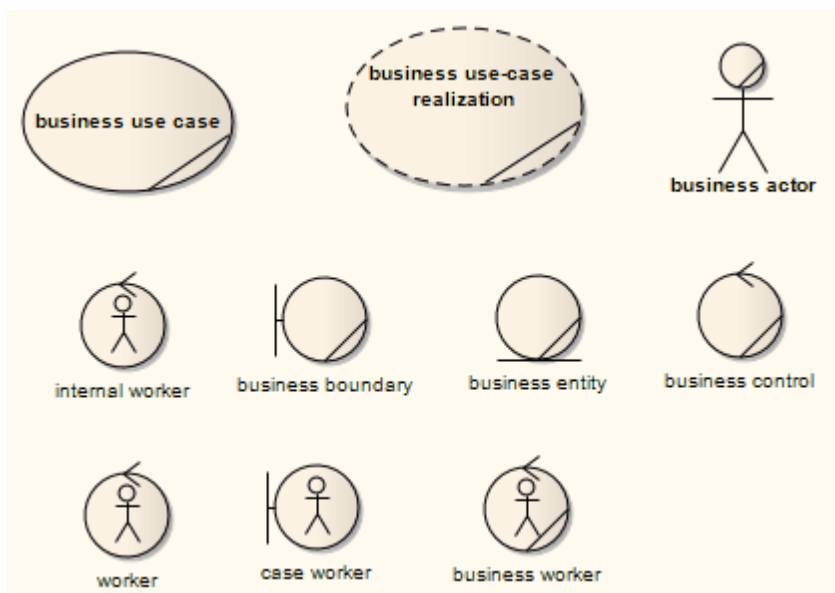
**Example Diagram** [Example Business Modeling Diagram](#) [1805]

#### Learning Center topics

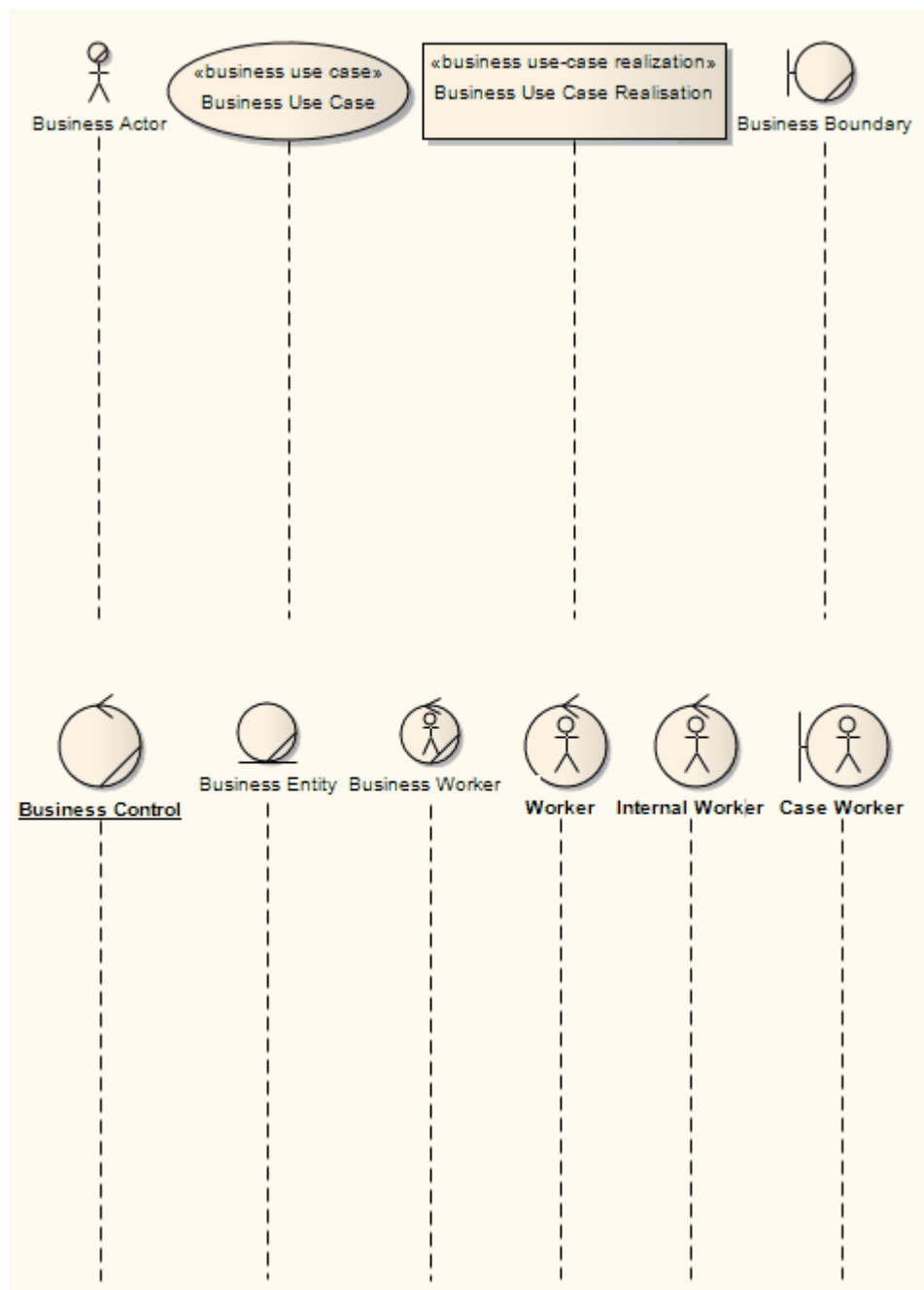
- **(Alt+F1) | Enterprise Architect | Modeling Languages | Business | Business Modeling and Interactions**

#### 9.3.1.1 Example Business Modeling Diagram

The following diagram shows the appearance of the elements when dragged and dropped onto a Business Modeling diagram:



The following diagram shows the appearance of the elements when dragged and dropped onto a Business Interaction diagram:



### 9.3.2 Business Models

#### Topics

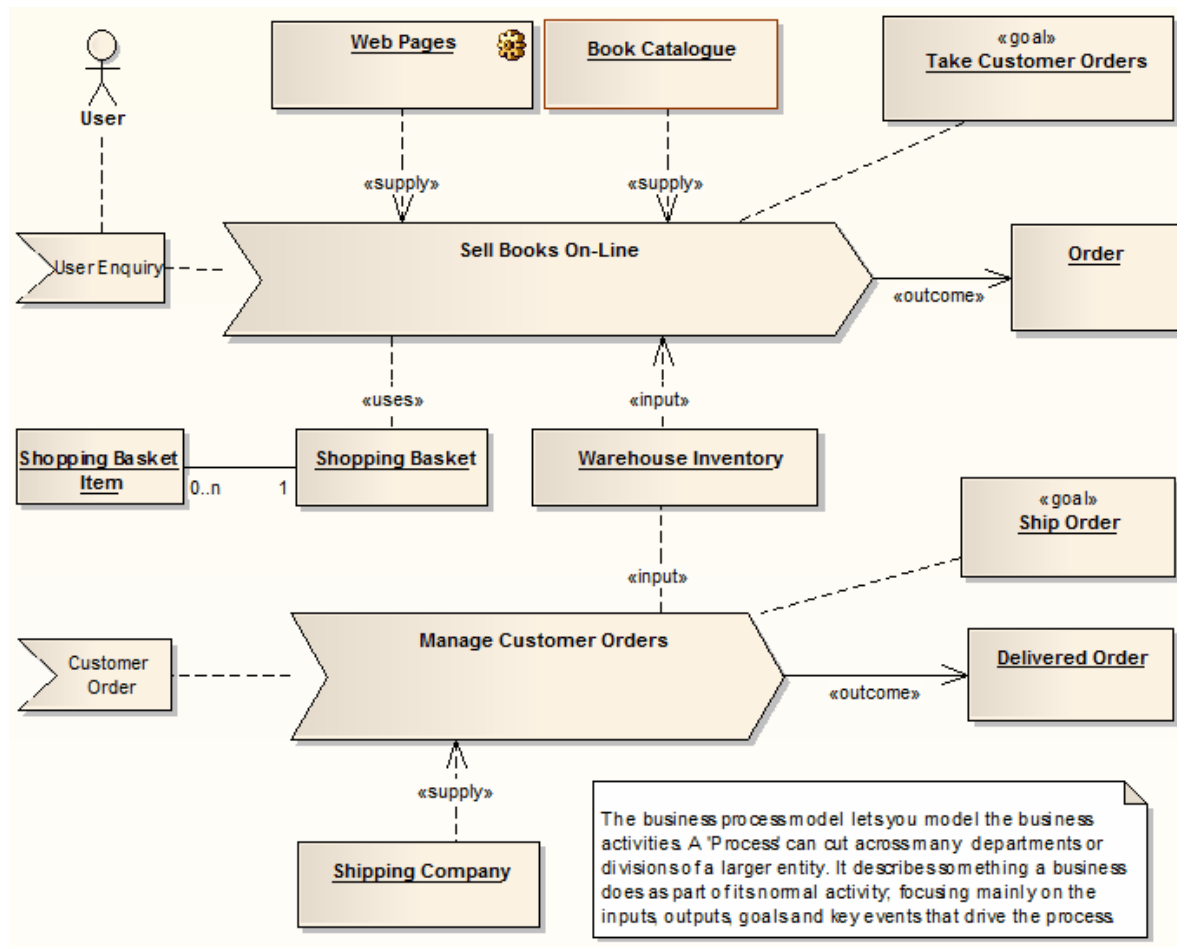
Topic	Detail	See also
<b>Modeling the Business Process</b>	Modeling the business process is an essential part of any software development process. It enables the analyst to capture the broad outline and procedures that govern what it is a business does. This <b>analysis model</b> provides an overview of where the proposed software system being considered fits into the organizational structure and daily activities. It can also provide the justification for building the	<a href="#">Analysis Model</a> 1809

Topic	Detail	See also
	<p>system by capturing the current manual and automated procedures that are to be rolled up into a new system, and the associated cost benefit.</p> <p>As an early model of business activity, it enables the analyst to capture the significant events, inputs, resources and outputs associated with business process. By connecting later design elements (such as Use Cases) back to the business process model through Implementation connectors, it is possible to build up a fully traceable model from the broad process outlines to the functional requirements and eventually to the software artifacts actually being constructed.</p> <p>As the Business Process Model typically has a broader and more inclusive range than just the software system being considered, it also enables the analyst to clearly map what is in the scope of the proposed system and what is to be implemented in other ways (such as a manual process).</p>	

#### **Example**

The example below demonstrates the kind of model that can be built up to represent a business process. In this model, the goal of the business process is to take customer orders and to ship those orders out. A user starts the process with an inquiry, which leads to the involvement of the Book Catalogue, Shopping Cart, on-line pages and warehouse inventory. The output of significance to the business is a customer order.





The second half of the process model is to respond to a customer order and ship the required items. The second process involves the warehouse inventory and shipping company, and completes when an order is delivered to the customer.

#### Learn more

- [Business Modeling and Business Interaction Diagrams](#) <sup>[1805]</sup>
- [Web Stereotypes](#) <sup>[1995]</sup>

### 9.3.2.1 Analysis Models

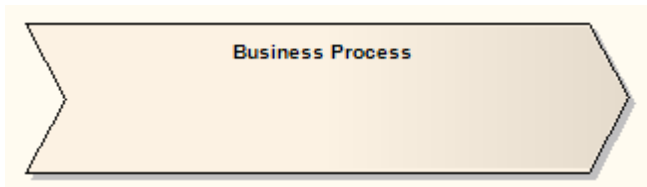
This section discusses the development of analysis models to construct business processes. It describes:

Topic	Link
Process Modeling Notation	<a href="#">Process Modeling Notation</a> <sup>[1810]</sup>
Inputs, Resources and Information	<a href="#">Inputs, Resources and Information</a> <sup>[1811]</sup>

Topic	Link
Events	<a href="#">Events</a> <sup>[1812]</sup>
Outputs	<a href="#">Outputs</a> <sup>[1812]</sup>
Goals	<a href="#">Goals</a> <sup>[1813]</sup>
A Complete Business Process	<a href="#">A Complete Business Process</a> <sup>[1813]</sup>

### 9.3.2.1.1 Process Modeling Notation

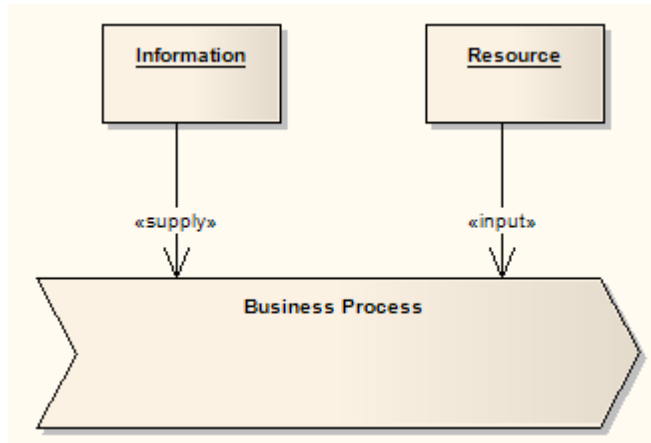
#### Topics

Topic	Detail	See also
<b>Abstract</b>	<p>A business process model typically defines the following elements:</p> <ul style="list-style-type: none"> <li>• The goal or reason for the process</li> <li>• Specific inputs</li> <li>• Specific outputs</li> <li>• Resources consumed</li> <li>• Activities that are performed in some order, and</li> <li>• Events that drive the process</li> </ul> <p>The business process:</p> <ul style="list-style-type: none"> <li>• Can affect more than one organizational unit</li> <li>• Can have a horizontal organizational impact</li> <li>• Creates value of some kind for the customer; customers can be internal or external</li> </ul> <p>A business process is a collection of activities designed to produce a specific output for a particular customer or market. It implies a strong emphasis on how the work is done within an organization, in contrast to a product's focus on what. A process is thus a specific ordering of work activities across time and place, with a beginning, an end, and clearly defined inputs and outputs: a structure for action. The notation used to depict a business process is illustrated below.</p> 	<p><a href="#">Process</a> <sup>[2008]</sup></p> <p><a href="#">Events</a> <sup>[1812]</sup></p> <p><a href="#">Activity</a> <sup>[1279]</sup></p>

Topic	Detail	See also
	<p>The <b>process notation</b> implies a flow of activities from left to right. Typically an <b>Event</b> element is placed to the left of the process and the output to the right. To specifically notate the internal activities, <b>Activity</b> elements can be placed inside the process element.</p>	
<b>The BPMN File</b>	<p>One popular notation and approach to business modeling is the Business Process Modeling Notation (BPMN). This notation is specifically targeted at the business modeling community and has a relatively direct mapping to UML through a BPMN Profile. Sparx Systems provides a built-in <b>profile for BPMN</b> modeling in Enterprise Architect.</p>	<a href="#">BPMN Models</a> [1845]

### 9.3.2.1.2 Inputs, Resources and Information

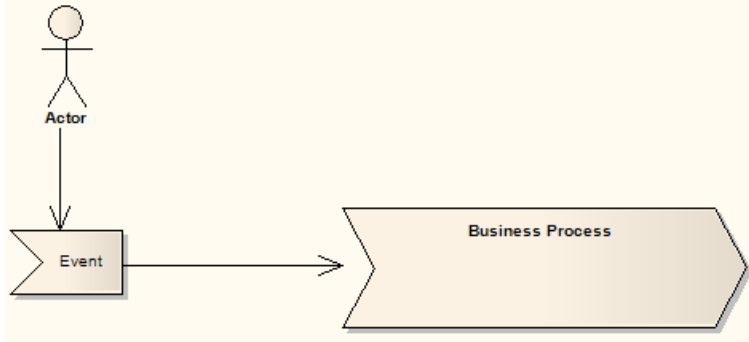
#### Topics

Topic	Detail	See also
<b>Abstract</b>	<p>Business processes use information to tailor or complete their activities. Information, unlike resources, is not consumed in the process; rather it is used as part of the transformation process. Information can come from external sources, from customers, from internal organizational units and could even be the product of other processes. A resource is an input to a business process and, unlike information, is typically consumed during the processing. For example, as each daily train service is run and actuals recorded, the service resource is 'used up' as far as the process of recording actual train times is concerned.</p> <p>The notation to illustrate information and resources is shown below.</p>  <p>A <i>Supply</i> connector indicates that the information or object linked to the process is not used up in the processing phase. For example,</p>	

Topic	Detail	See also
	<p>order templates can be used over and over to provide new orders of a certain style; the templates are not altered or exhausted as part of this activity.</p> <p>An <i>Input</i> connector indicates that the attached object or resource is consumed in the processing procedure. As an example, as customer orders are processed they are completed and signed off, and typically are used only once per unique resource (order).</p>	

### 9.3.2.1.3 Events

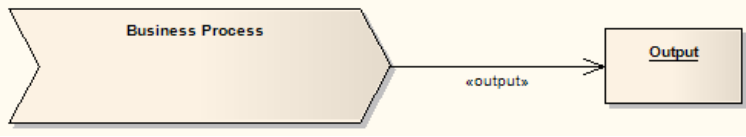
#### Topics

Topic	Detail	See also
<b>Abstract</b>	<p>An <b>event</b> is the receipt of some object, a time or date reached, a notification or some other trigger that initiates the business process. The event might be consumed and transformed (for example a customer order) or simply act as a catalyst (for example, nightly batch job).</p>  <pre> graph TD     Actor((Actor)) --&gt; Event{Event}     Event --&gt; BusinessProcess[Business Process]   </pre>	<a href="#">Event</a> <small>[2007]</small>

### 9.3.2.1.4 Outputs

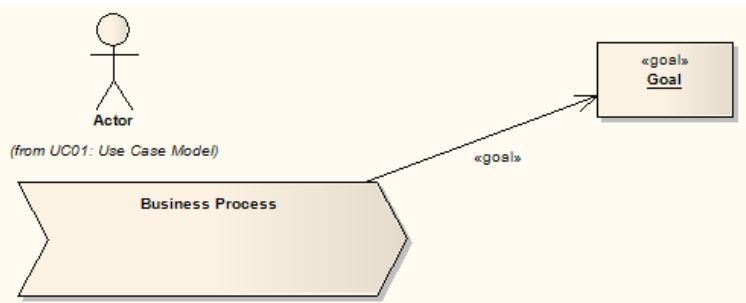
#### Topics

Topic	Detail	See also
<b>Abstract</b>	<p>A business process typically produces one or more outputs of value to the business, either for internal use or to satisfy external requirements. An output might be a physical object (such as a report or invoice), a transformation of raw resources into a new arrangement (a daily schedule or roster) or an overall business result such as completing a customer order.</p> <p>An output of one business process might feed into another process, either as a requested item or a trigger to initiate new activities.</p>	

Topic	Detail	See also
	 <p>An Output connector indicates that the business process produces some object (either physical or logical) that is of value to the organization, either as an externally visible item or as an internal product (possibly feeding another process).</p>	

### 9.3.2.1.5 Goals

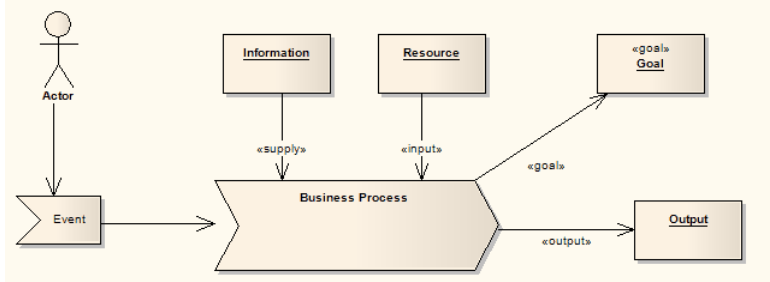
#### Topics

Topic	Detail	See also
<b>Abstract</b>	<p>A business process has some well defined goal. This is the reason the organization does this work, and should be defined in terms of the benefits this process has for the organization as a whole and in satisfying the business requirements.</p>  <p>A Goal connector indicates that the attached object to the business process describes the goal of the process. A goal is the business justification for performing the activity.</p>	

### 9.3.2.1.6 A Complete Business Process

#### Topics

Topic	Detail	See also
<b>Abstract</b>	The diagram below illustrates how the various model elements can be grouped together to produce a coherent picture of a named	

Topic	Detail	See also
	<p>business process. Included are the inputs, outputs, events, goals and other resources that are of significance.</p> 	

### 9.3.3 Modeling Business Rules

In any business action or process, the start, progress and end result are usually determined by reference to a set of **rules**. These rules can be very simple, such as 'the client must present documentary evidence of being at least 18 years old', or very complex, such as the actuarial rules that determine what a tailored insurance policy will and will not cover.

Whether the rules of your business are simple or complex, there are two sets of considerations that you need to take into account:

- **How to manage the rules** - How are they initially identified? Where are they held? Are the rules easily maintained and updated? How are they refined and tested?
- **How to use the rules** - How easy is it to identify which rules apply in a specific context? How easily can any specific rule be recognized and applied? How are the rules **executed** in the process - can they be integrated with the process? Can execution of the rules be **automated** in the process?

Both sets of considerations can be easily managed by modeling your business processes in Enterprise Architect, and using the **Business Rule Model** facility.

#### Managing Rules

Broadly, modeling your business processes can clarify:

- Your business **requirements** (from which many business rules are ultimately derived)
- The **use cases** - and the scenarios in each use case - to satisfy those requirements, and
- The exact processes, stages, objects, actions and data structures that support those use cases, represented by **Classes**

This process will also clarify which of your current business **rules** are applicable to which points in each process, and what refinements or new business rules are required. You can then **map** your business rules to existing or new Classes, using two specific Business Rules models; the:

- **Business Domain** model, in which you group the business objects (represented by Classes) involved in a process or application, and develop a **Rule Flow** that defines the tasks (as **Rule Task** elements) associated with the process as a whole or specific objects in the process
- **Business Rules** model, in which you create a specific **Business Rule element** for each business rule and **associate** it with the **Rule Task** to which the rule applies

When you have defined all the tasks, their sequences, and the rules that apply to each one, you can compose the rules per task to define the values and conditions of the rules and how they take effect in the task. You can then validate the rules for the task to ensure that they are logical.

A valuable resource that you have created in this process is a **database of business rules** associated directly with the tasks and procedures they apply to, easily explored (according to the naming and/or numbering convention you have used) with the Model Search and other navigation and display facilities, and documented through the document or web reporting facilities. You can also record further information on each rule using internal or external notes, Tagged Values and Linked Documents.

### Using Business Rules

Having set up the business rules database, your users can access the models or their documentation as a reference. As explained above, the context of any given rule, or the rules applicable to a context, can be quickly established using the search, navigation or Traceability facilities.

However, you can use Enterprise Architect to model and create applications and user interfaces that can **apply** the business rules you have defined, and a further facility of Business Rule modeling is to **generate the behavioral code** for the rules in a specific task. You can merge this into your code to prompt for or even automate the correct use of the business rules in performing a task.

### Advantages of modeling Business Rules

Whether you create a database of rules, or applications that apply the rules, you have a modular solution to a business process requirement. This provides an advantage in localization. Business Rules can vary between locations; for example, car hire operates in roughly the same way in most countries, but the legal driving age differs between the countries, as do the models of car available for hire. You can easily create different localized rule modules and switch the appropriate one for the current location into the common model.

### Notes

- Business Rule Modeling is available in the Business and Software Engineering edition and the Ultimate edition of Enterprise Architect

### Learn more

- [Develop a Business Rules Model](#) <sup>[1816]</sup>
- [Compose Business Rules](#) <sup>[1833]</sup>
- [Enterprise Architect Product Demonstration - Business Rules](#) (Online Resource)

### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Modeling Languages | Business | Business Rule Model Introduction**
- (Alt+F1) | **Enterprise Architect | Modeling Languages | Business | Business Rule Model**
- (Alt+F1) | **Enterprise Architect | Modeling Languages | Business | Business Rule Flow**
- (Alt+F1) | **Enterprise Architect | Business Modeling | Business Rules | Getting Started**
- (Alt+F1) | **Enterprise Architect | Business Modeling | Business Rules | Example Model**

### 9.3.3.1 *Develop a Business Rules Model*

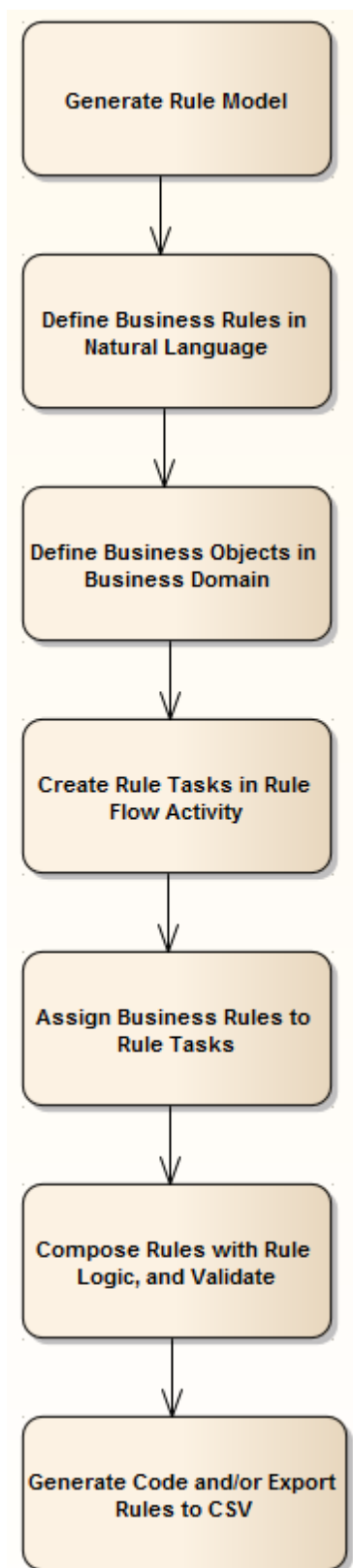
In modeling Business Rules, you first develop the **model structure** to represent the rules in the context of their use, and then effectively compile (or **compose**) the rules to make them operational within that context. From the compiled rules you can either create a spreadsheet for reference or generate behavioral code for applications that apply the rules, or both.

#### Process Summary

1. Use the Model Wizard to generate a **Business Rules Model** in which to define the business rules.
2. In the generated Business Rules **diagram**, begin to identify the business rules as Business Rule **elements**, each element representing a specific business rule.
3. In the generated Business Domain Model diagram, edit the existing Classes - and create others - to represent the Business Objects in the business domain or process; these provide the business vocabulary as the context for the business rules.
4. One of the Classes represents the actual **application** of the rules; under this Class is a Rule Flow Activity and Activity diagram, in which you create the Rule Tasks under which the business rules are grouped, in the sequence in which the business rules are executed. If you require a higher level of grouping to define different areas of rule application, you can create other rule-processing Classes with their own Activities.
5. Return to the Business Rules diagram and drag in the Rule Task elements from the Project Browser, assigning to each Task the corresponding Business Rule elements.
6. Compose and validate the business rules, using the Rule Composer.
7. If you want code that applies and executes the business rules, generate it from the Class elements that contain the Business Rules Activities.

The steps are represented graphically in the following flow:



**Notes**

- Business Rule Modeling is available in the Business and Software Engineering edition and the

Ultimate edition of Enterprise Architect

### Learn more

- [Modeling Business Rules](#) <sup>[1814]</sup>
- [Generate a Business Rules Model](#) <sup>[1815]</sup>
- [Create a Business Domain Model](#) <sup>[1824]</sup>
- [Create a Rule Flow Activity](#) <sup>[1826]</sup>
- [Model Business Rules For RuleTasks](#) <sup>[1821]</sup>
- [Compose Business Rules](#) <sup>[1833]</sup>

### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Business Modeling | Business Rules | Getting Started**
- (Alt+F1) | **Enterprise Architect | Business Modeling | Business Rules | Example Model**
- (Alt+F1) | **Enterprise Architect | Modeling Languages | Business | Business Rule Model**

#### **9.3.3.1.1 Generate a Business Rules Model**

The Business Rules Model captures:

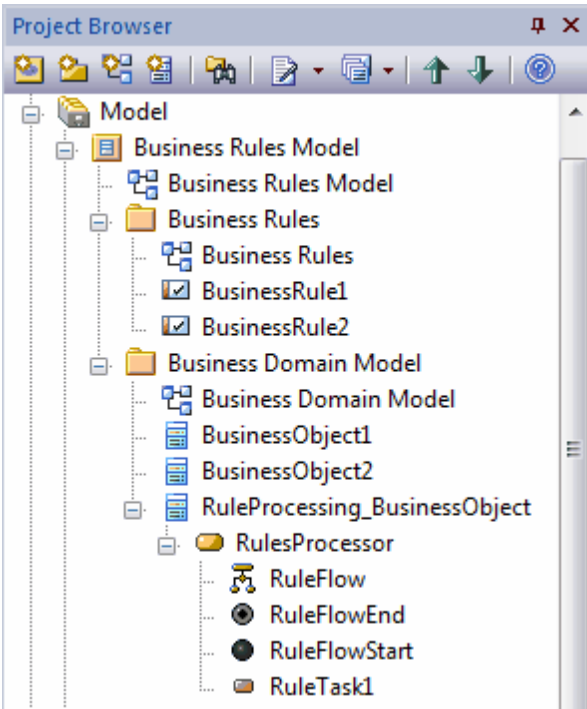
- The rules that apply to a business process
- The business tasks or objects that the rules take effect on, and
- The actual processing that takes place to apply the rules and produce a decision or result

As a very useful starting point in setting up your Business Rules model, you can generate the model structure and initial components using the Enterprise Architect Model Wizard.

**Access** **Project Browser | New Model From Pattern toolbar icon > Model Patterns, or**  
**Project Browser | Right-click model root node | Add a New Model using Wizard > Model**  
**Patterns, or**  
**Project Browser | Right-click package | Add | Add a New Model using Wizard > Model Patterns**

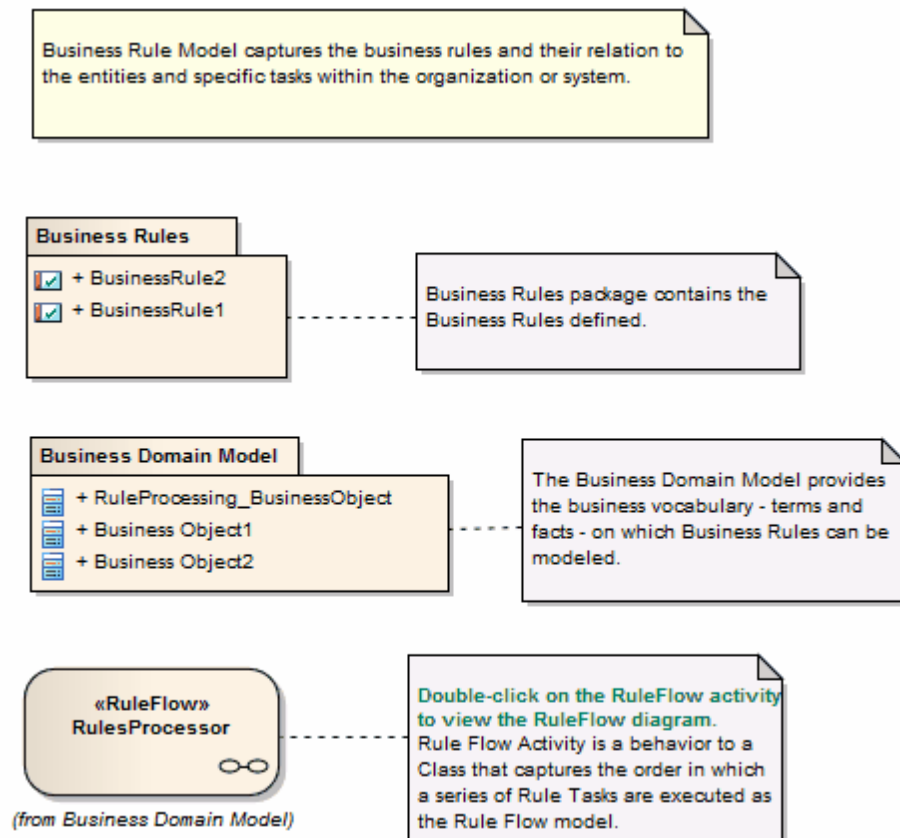
### Generate Business Rules Model from Model Wizard

Step	Action	See also
1	In the Technology panel, click on <b>Business Rule Model</b> .	
2	In the Name panel, select the checkbox next to the <b>Business Rule Model</b> icon.	
3	Click on the <b>OK</b> button.  A Business Rules Model structure is generated in the Project Browser, as shown.	

Step	Action	See also
	 <p>The screenshot shows a 'Project Browser' window with a toolbar at the top. The tree structure is as follows:</p> <ul style="list-style-type: none"><li>Model<ul style="list-style-type: none"><li>Business Rules Model<ul style="list-style-type: none"><li>Business Rules Model<ul style="list-style-type: none"><li>Business Rules<ul style="list-style-type: none"><li>BusinessRule1</li><li>BusinessRule2</li></ul></li><li>Business Domain Model<ul style="list-style-type: none"><li>Business Domain Model<ul style="list-style-type: none"><li>BusinessObject1</li><li>BusinessObject2</li></ul></li><li>RuleProcessing_BusinessObject</li></ul></li><li>RulesProcessor<ul style="list-style-type: none"><li>RuleFlow</li><li>RuleFlowEnd</li><li>RuleFlowStart</li><li>RuleTask1</li></ul></li></ul></li></ul></li></ul></li></ul>	

### Example Diagram

The *Business Rules Model* diagram generated at the top level of the model and shown below, encapsulates the components of the Business Rules model.



### Notes

- Business Rule Modeling is available in the Business and Software Engineering edition and the Ultimate edition of Enterprise Architect
- To see an example of a Business Rules model, open the *EAExample* model provided with the installer, and expand:

**Project Models > Analysis and Business Modeling > Business Domain Model > Business Rule Model > Business Domain Model and > Defining Business Rules**

### Learn more

- [Modeling Business Rules](#) <sup>1814</sup>
- [Create a Business Domain Model](#) <sup>1824</sup>
- [Create a Rule Flow Activity](#) <sup>1826</sup>
- [Model Business Rules For RuleTasks](#) <sup>1827</sup>
- [Compose Business Rules](#) <sup>1833</sup>

### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Business Modeling | Business Rules | Getting Started**
- (Alt+F1) | **Enterprise Architect | Business Modeling | Business Rules | Example Model**
- (Alt+F1) | **Enterprise Architect | Business Modeling | Business Rules | Rule Model Example**

- (Alt+F1) | **Enterprise Architect** | **Modeling Languages** | **Business** | **Business Rule Model**

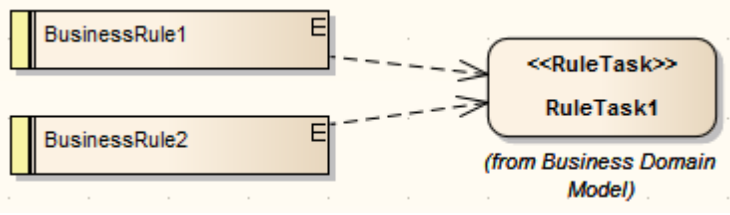
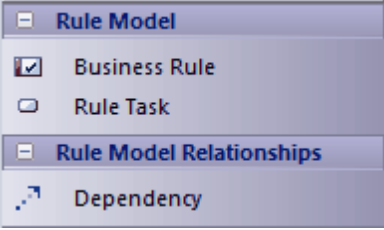
### 9.3.3.1.2 Model Business Rules

In the Business Rules Model, you initially define each business rule as a **Business Rule element** and later group these rules by linking them (using Dependency connectors) with **Rule Task** elements. In the:

- First stage you assemble a collection of rules
- Second stage you organize the rules into groups and sequences through the Rule Task elements created in the Business Domain model, and refine them by adding further Business Rule elements (and, if appropriate, further Rule Task elements)

In the Car Rental example in the *EAExample* model, rules have been defined and grouped to perform an eligibility check on a customer, to determine if the customer is eligible to rent a car.

#### Define Business Rule elements

Step	Action	See also
1	<p>In the generated Business Rules Model, expand the <b>Business Rules package</b> and double-click on the <b>Business Rules diagram</b> to open it. The diagram shows two example Business Rule elements connected to an example Rule Task.</p>  <p>In the Diagram Toolbox, the Rule Model pages display.</p> 	
2	<p>If you should need to create another (empty) Business Rules diagram, right-click on the Business Rules package and select <b>Add   Add Diagram</b>.</p> <p>On the New Diagram dialog, type an appropriate name in the <b>Name</b> field, select <b>Business Rule Model</b> in the Select From panel, and select <b>Rule Model</b> in the Diagram Types panel.</p> <p>Click on the <b>OK</b> button.</p>	
3	<p>For each business rule you want to identify, drag the Business Rule icon from the Toolbox.</p>	


Step	Action	See also
	Type the rule (or a shortened version of it) in the <b>Short Description</b> field of the element Properties dialog. This displays as the name of the element in the diagram.  You will later define the parameters of the rules using the Rule Composer.	<a href="#">Compose Business Rules</a> <sup>[1833]</sup>
4	When you have created all the required Business Rule elements, create the Business Domain model, the Rule Flow Activity diagram and the Business Task elements.	<a href="#">Create a Business Domain Model</a> <sup>[1824]</sup>

### Associate Business Rules with Business Tasks

After you have created the Business Domain Model and Rule Flow Activity (which contains the Rule Task elements in the context of the business process), you can link the Business Rule elements to the Rule Task for the action in which the rules take effect. You can apply a Business Rule to more than one Rule Task, if it has an effect in different contexts.

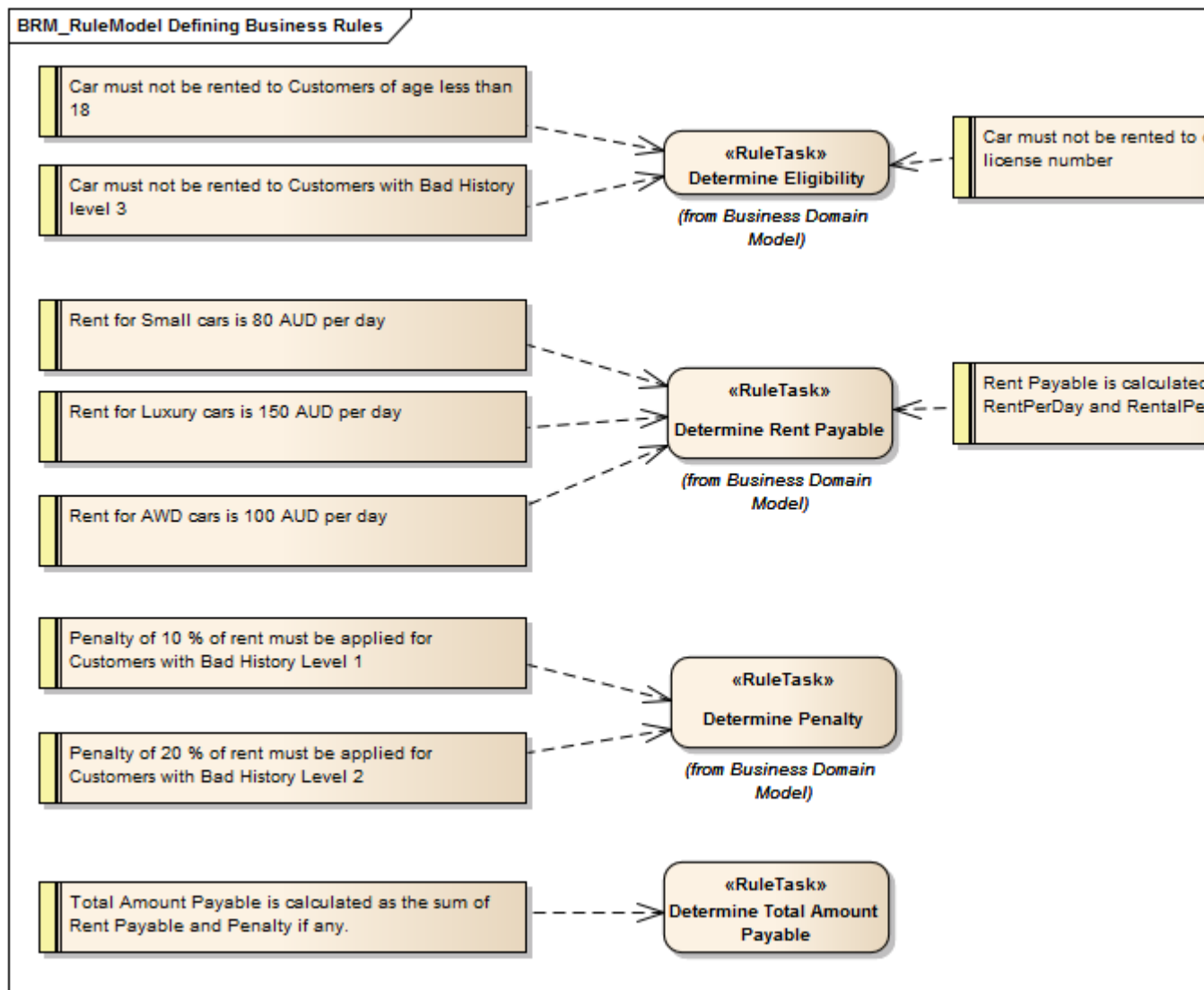
There are several ways to establish the relationship between Business Rule and Rule Task.

Method	Action	See also
<b>Use the Business Rules diagram</b>	Open the Business Rules diagram and drag onto it a Rule Task element from the RuleFlow Activity in the Project Browser.  Create a Dependency relationship to the Rule Task from each Business Rule element that applies to it (by dragging the connector from the Toolbox, or by using the Quicklinker arrow on the Business Rule element).  Repeat the process for the next Rule Task element from the Rule Flow Activity.  Create any additional Business Rule and Rule Task elements that are necessary (this should not be a common event); you must add any new Rule Tasks to the Rule Flow Activity diagram.  Save the diagram and open the Rule Composer for the first Rule Task element.	<a href="#">Create a Rule Flow Activity</a> <sup>[1826]</sup>  <a href="#">Compose Business Rules</a> <sup>[1833]</sup>
<b>Assemble existing elements through the Rule Composer</b>	In the Project Browser, open the Rule Composer on a Rule Task element, and drag each applicable Business Rule element from the Project Browser into an empty row of the Rule Statements table.  This establishes a Dependency relationship between the Rule Task element and each Business Rule element.  Continue to compose the rule conditions.	<a href="#">Compose Business Rules</a> <sup>[1833]</sup>
<b>Use the Relationship Matrix</b>	You can quickly create all the Dependency relationships between a number of Rule Task elements in the Business Domain Model package and the Business Rule elements that apply to each one.  Set the source element <b>Type</b> to <b>BusinessRule</b> and the target	<a href="#">Relationship Matrix</a> <sup>[727]</sup>

Method	Action	See also
	<p>element <b>Type</b> to <b>RuleTask</b>.</p> <p>Set the <b>Link Type</b> to <b>Dependency</b>, and the <b>Direction</b> to <b>Source-&gt;Target</b>.</p> <p>Set the <b>Source</b> package to <b>Business Rules</b> and the <b>Target</b> to <b>Business Domain Model</b>.</p> <p>In the cell at each intersection of a Business Task column and the appropriate Business Rule row, right click and select the <b>Create new relationship   UML:: Dependency</b> option.</p> <p>When you have finished creating the relationships, close the Relationship Matrix.</p> <p>Open the Rule Composer for the first Rule Task element in the Project Browser.</p>	<p><a href="#">Compose Business Rules</a> </p>

### Example

However you connect the Rule Task and Business Rule elements, if they are added to a diagram it might display as shown.



### Notes

- Business Rule Modeling is available in the Business and Software Engineering edition and the Ultimate edition of Enterprise Architect

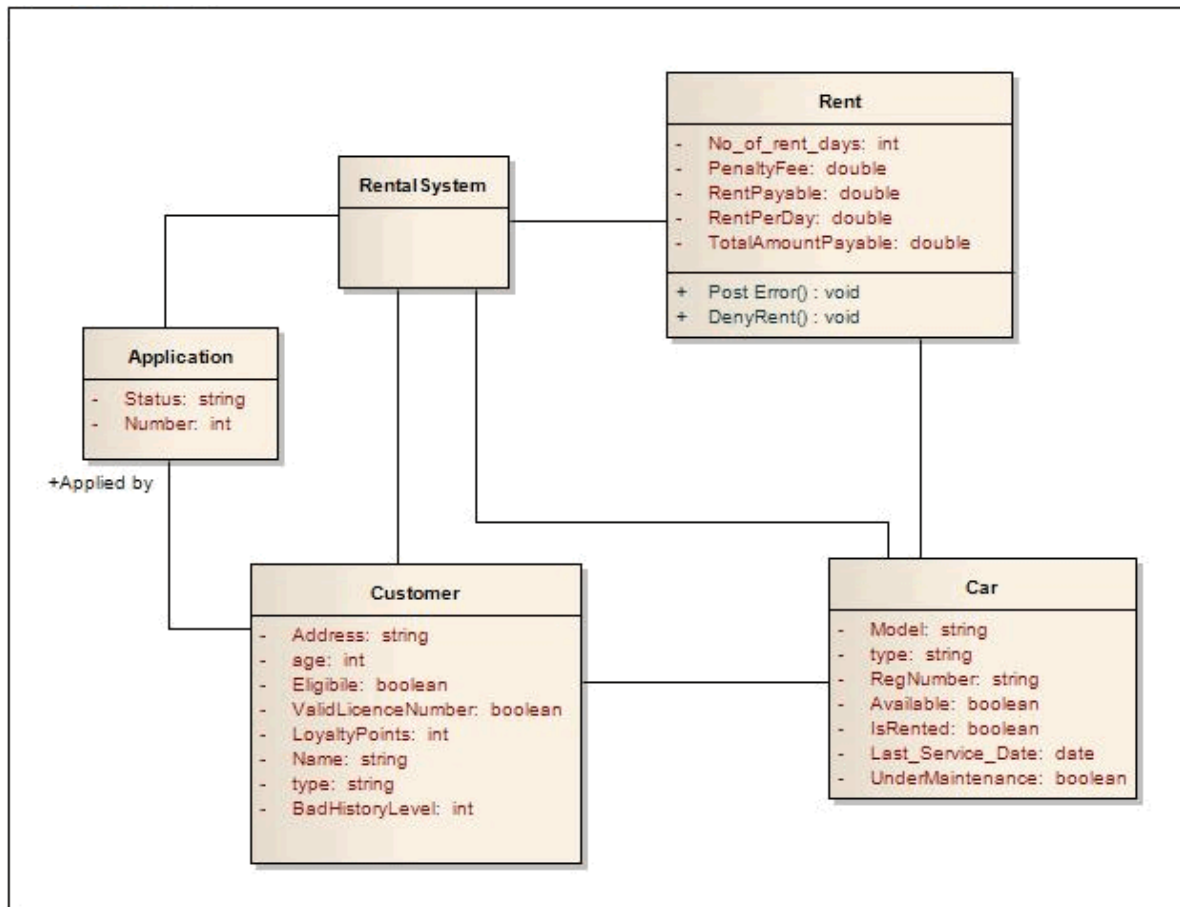
### Learning Center topics

- (Alt+F1) | [Enterprise Architect | Modeling Languages | Business | Business Rule Model](#)
- (Alt+F1) | [Enterprise Architect | Business Modeling | Business Rules | Define Business Rules](#)
- (Alt+F1) | [Enterprise Architect | Business Modeling | Business Rules | Rule Model Example](#)

#### 9.3.3.1.3 Create a Business Domain Model

The **Business Domain Model** provides the business **vocabulary** - the terms and facts - on which Business Rules can be modeled. In Enterprise Architect a Business Domain model is represented as a conceptual Class diagram, as illustrated by this diagram from the Car Rental System model from the *EAExample* model.





In the Business Domain model shown in this diagram, the Classes *Rent*, *Customer*, *Car* and *Application*, together with their attributes and operations, provide the **terms for the business vocabulary** for the car rental system. The operations and attributes identify the conditions that must be met, the actions that must be taken, and the calculations that must be made to filter and apply the rules to provide a specific value or outcome.

The Class *Rental System* **processes the rules**; to make this possible, you add a **Rule Flow Activity** as a **behavior** for this Class.

When you create a Rule Flow Activity under a Class, you model the events and sequence as a structure of **Rule Tasks** (Actions). When you generate code for the Class (in the example, *Rental System*) the rule flow behavior is rendered as a method inside the Class.

Alternatively, if you have existing **operations** in the Class that already suit the purpose, you can model business tasks in those operations. When code is generated for the Class, the rules logic is generated as the method body for the corresponding operation.

### Notes

- Business Rule Modeling is available in the Business and Software Engineering edition and the Ultimate edition of Enterprise Architect
- When you create Classes in the Business Domain model, select the correct language for code generation to ensure that the correct data type is set for attributes and operation parameters
- Business Rules code generation is supported for the following languages:
  - C++

- C#
- Java
- VB.Net

#### Learn more

- [Create a Rule Flow Activity](#)<sup>[1826]</sup>
- [Model Rules in an Operation](#)<sup>[1830]</sup>

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Business Modeling | Business Rules | Business Domain Model**

#### 9.3.3.1.4 Create a Rule Flow Activity

When you set up a Business Domain model within a Business Rules model, you create a **Rule Flow Activity** as a behavior for one of the domain Classes, to enable that Class to process a set of rules. In the Rule Flow Activity you create a number of **Rule Task** elements, which are stereotyped Actions that group Business Rules for a specific task. The Rule Flow Activity automatically generates a Rule Flow diagram, in which you create the Rule Task elements and model the **sequence** in which they are executed.


#### Add a Rule Flow Activity to a Class

Step	Action	See also
1	On the Business Domain model diagram, right-click on the Class that processes the rules (in the Car Rental example in the <i>EAExample</i> model, this would be <i>Rental System</i> ).	<a href="#">Create a Business Domain Model</a> <sup>[1824]</sup>
2	From the context menu select the <b>New Diagram   RuleFlow Activity</b> option.  A new Rule Flow Activity is created with a <b>Rule Flow diagram</b> , which immediately opens. Go on to create the Rule Task elements (below).  Code generation for a Rule Flow model renders each RuleFlow Activity as a set of operations or methods. Depending on what you want these methods to do, you might want to pass in parameters to be used within the Rule Flow Activity.	<a href="#">Pass Parameters to Rule Flow Activity</a> <sup>[1829]</sup>

#### Add Rule Task elements to the Rule Flow Activity diagram

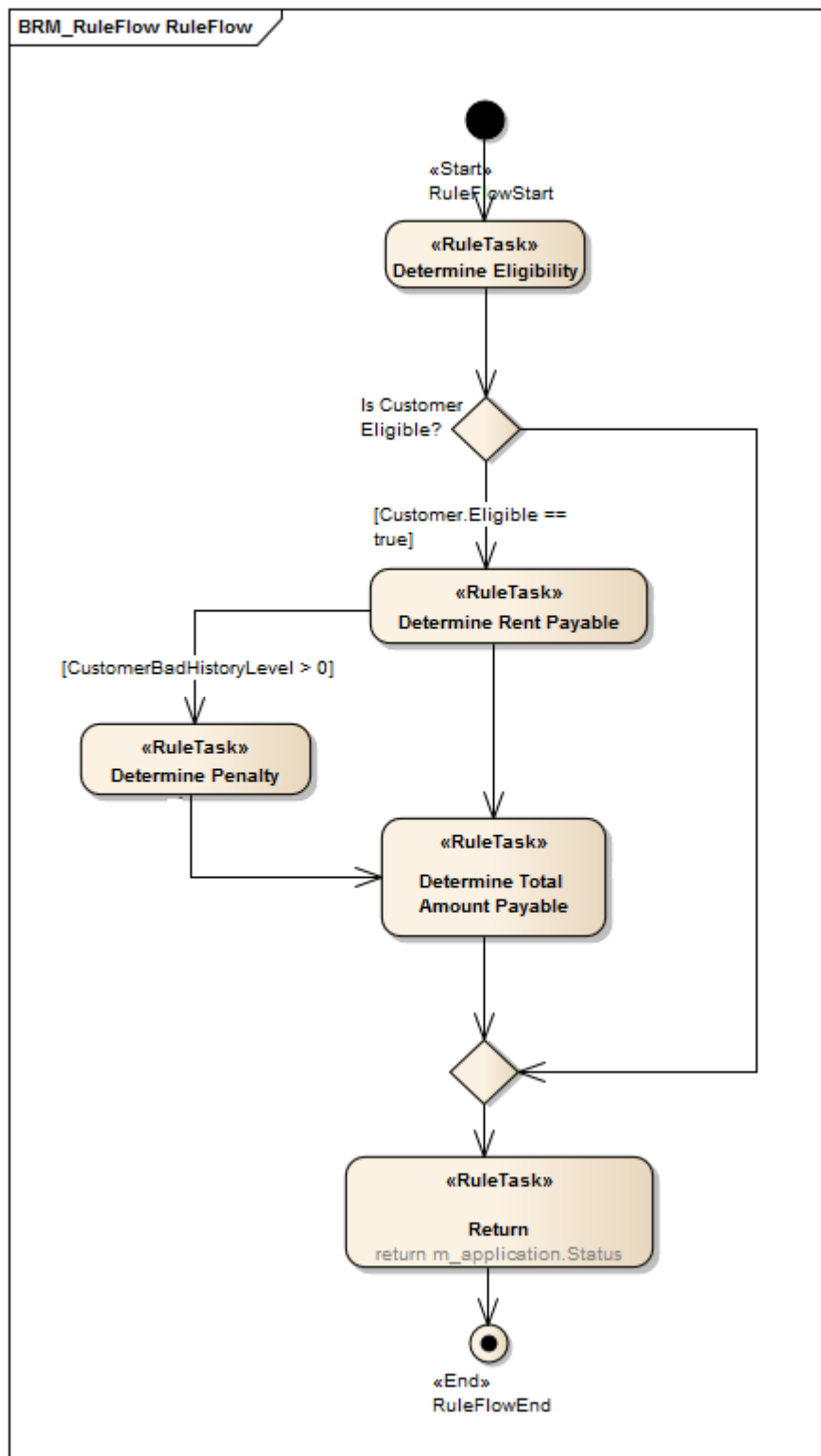
You can create Rule Task elements directly under the Rule Flow Activity in the **Project Browser**, by clicking on the **New Element** icon in the Toolbar and selecting the **UML:: Activity** toolset, **Action** element type and **RuleTask** stereotype. However, it is much simpler to create the elements on the Rule Flow diagram, and at the same time to organize them into their processing sequence.

Step	Detail	See also
1	(If necessary) Click on the <b>More tools</b> button in the Diagram Toolbox and select	

Step	Detail	See also
	<p>the <b>Business Rule Model   Rule Flow</b> option.</p> 	
2	<p>Drag the <b>Rule Task</b> icon from the Toolbox onto the diagram and give the element, as a name, the title of the task it represents, such as <i>Calculate Debit Charge</i> or <i>Determine Eligibility</i>.</p> <p>Create a Rule Task element for each task or action in the process.</p> <p>You can also use the Quicklinker arrow to create the new elements and Control Flow connectors.</p>	
3	<p>Organize the Rule Tasks into a sequence of events, initiated and terminated by the <b>Start</b> and <b>End</b> elements, and representing any branching and rejoining with <b>Decision</b> and <b>Merge</b> elements. All elements are connected by <b>Control Flow</b> connectors.</p> <p>See the example diagram below.</p>	
4	Go to the Business Rules model diagram and group the Business Rule elements on their appropriate Rule Task element.	<a href="#">Model Business Rules</a> <small>1827</small>

### Example

This Rule Flow diagram is from the *EAExample* model Car Rental example.



The Rule Task elements *Determine Eligibility*, *Determine Rent Payable*, *Determine Penalty* and *Determine Total Amount Payable* group the business rules for the specific task indicated by the element name

#### Notes

- Business Rule Modeling is available in the Business and Software Engineering edition and the Ultimate edition of Enterprise Architect
- In a Rule Flow diagram, every **Decision** node has a matching **Merge** node to ensure proper code generation
- For code generation, the **Rule Task** elements must be grouped inside the appropriate Rule Flow Activity in the Project Browser (**Business Rule** elements can be defined anywhere in the model, as they can be used in more than one Rule Task)

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Modeling Languages | Business | Business Rule Flow**
- (Alt+F1) | **Enterprise Architect | Business Modeling | Business Rules | Rule Flow Model**
- (Alt+F1) | **Enterprise Architect | Business Modeling | Business Rules | Rule Flow Example**

#### **9.3.3.1.4.1 Pass Parameters to Rule Flow Activity**

When you generate code for a Rule Flow model, each RuleFlow Activity is rendered as a set of operations or methods. You can pass in Activity Parameters to be used by the Rule Tasks within the Rule Flow Activity, to define what you want the methods to do. You can use the parameters as condition variables or action variables in the Business Rule Decision Table, or as rule variables in the Computation Table for any of the Rule Tasks.

If the Activity Parameter is not accessible to a Rule Task, the system displays an error message.

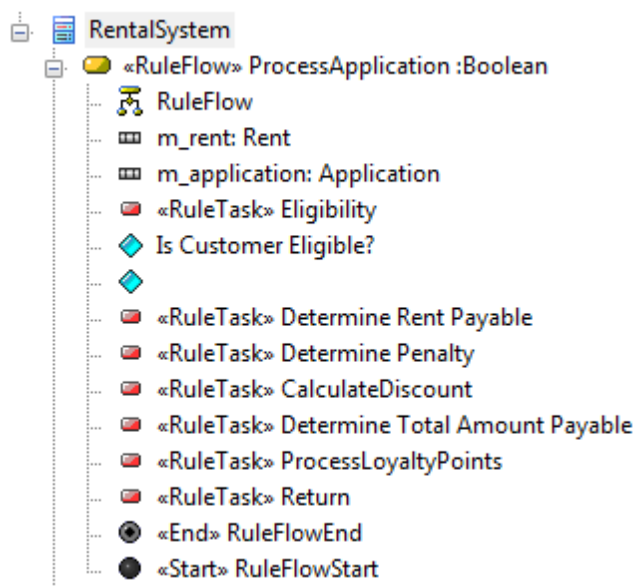
**Access**    **Project Browser | Double-click Rule Flow Activity element > Parameters**

#### Define parameters to be used within a Rule Flow Activity

Step	Action	See also
1	On the Parameters page, create and define each parameter, in particular the <b>Type</b> and <b>Default</b> values.	<a href="#">Define Parameters</a> [1028]
2	Save each parameter and, when you have finished setting all parameters, close the Properties dialog.	

#### Example

In this hierarchy, the parameters *m\_rent* and *m\_application* can be used by any of the Rule Tasks under the *ProcessApplication* Rule Flow Activity.



#### Learn more

- [Compose Business Rules](#) <sup>[1833]</sup>

#### 9.3.3.1.5 Model Rules In an Operation

You can model business rules as Business Rule elements in the Business Rules package, and attach them to the **Rule Task elements** in a RuleFlow Activity diagram.

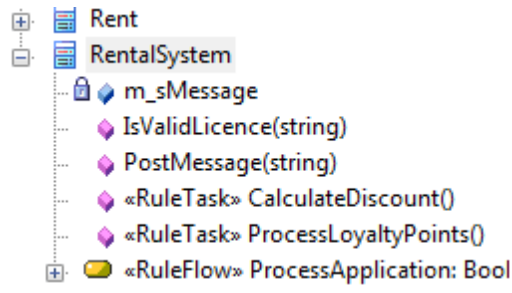
**Alternatively**, in the Business Domain model, if you have operations in the rules processing Class that represent business action, you can define each of those **operations** as a **Rule Task** and attach the Business Rules to these operations on the Business Rules diagram or within the Rule Composer.

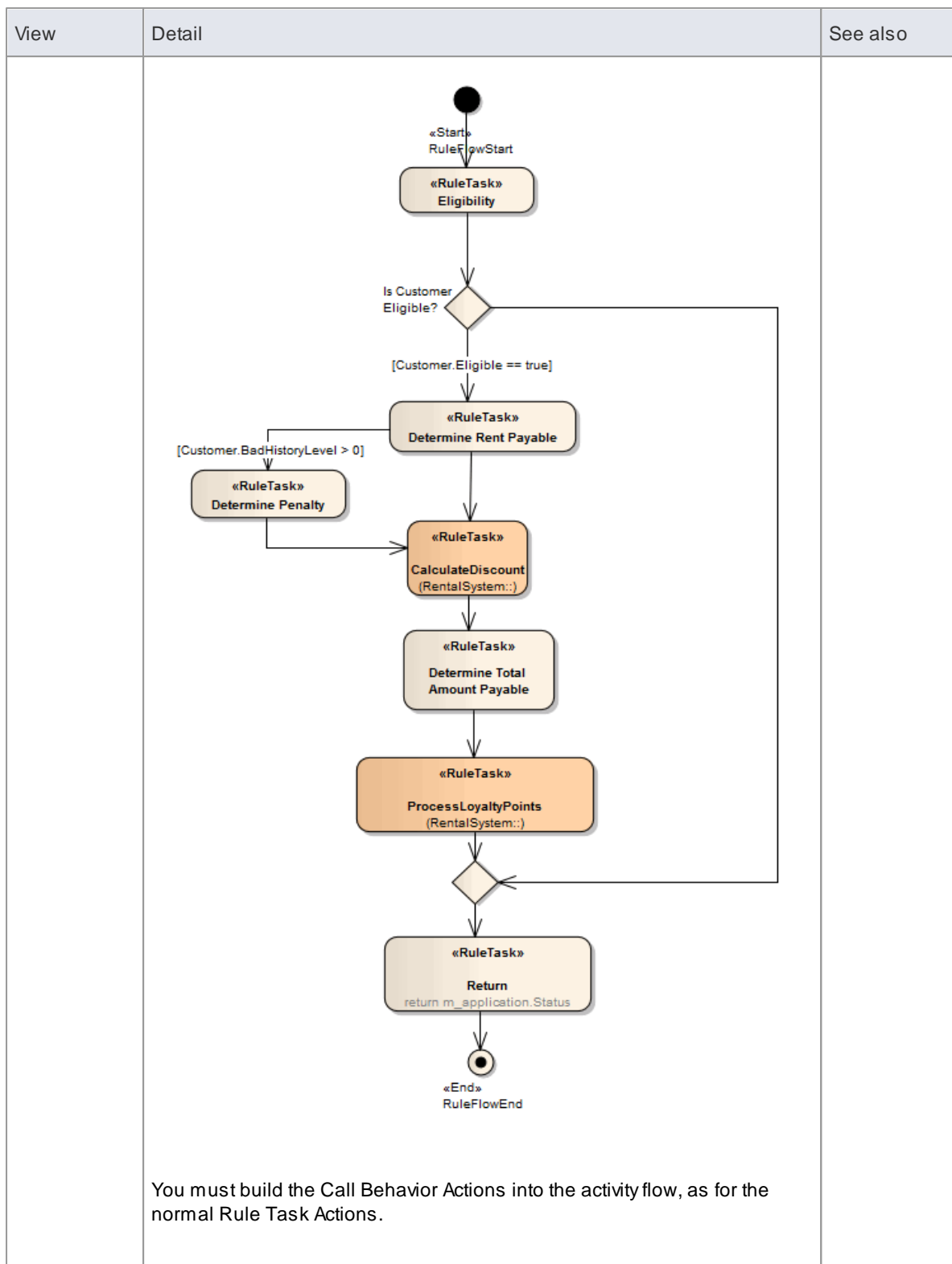
**Access** **Project Browser | Double-click on Operation > General**

#### Model Business Task on an operation

Step	Action	See also
1	In the <b>Stereotype</b> field type <b>RuleTask</b> .	
2	Click on the <b>Save</b> button and on the <b>Close</b> button.	
3	Drag the Operation from the Project Browser onto the RuleFlow Activity diagram. Also assign Business Rules to the operation, as for a Rule Task element.	<a href="#">Model Business Rules</a> <sup>[1827]</sup>

**Representation of Operation Rule Tasks**

View	Detail	See also
<b>Project Browser</b>	<p>The operations stereotyped as <b>RuleTask</b> display in the Project Browser as shown:</p>  <p>To pass the <b>parameters</b> for these operation calls, open the operation Properties dialog and select the Call page, then set the <b>Behavior</b> field to the operation to be called; under the <b>Arguments</b> field, click on the <b>Edit</b> button and set up or edit the argument values to be passed.</p> <p>On code generation, the code for rules logic is generated in the method body.</p>	<p><a href="#">Behaviour Calls</a> <sup>1026</sup></p> <p><a href="#">Behaviour Call Arguments</a> <sup>1027</sup></p>
<b>On the Rule Flow Diagram</b>	<p>When you drag and drop a RuleTask-stereotyped operation onto a Rule Flow diagram, an operation <b>Call Behavior Action</b> is created, as shown by the two darker elements.</p>	



#### Notes

- Business Rule Modeling is available in the Business and Software Engineering edition and the Ultimate edition of Enterprise Architect



Learn more

- [Model Business Rules](#) <sup>[1821]</sup>
- [Create a Rule Flow Activity](#) <sup>[1826]</sup>
- [Create a Business Domain Model](#) <sup>[1824]</sup>
- [Add and Remove Rules](#) <sup>[1834]</sup>

**9.3.3.2 Compose Business Rules**

In modeling your Business Rules, you first define each business rule as a **conceptual-level, plain text** string within a Business Rule element, and then group the rules by association with Business Task elements. Your next step is to define exactly how the rules operate **within the task**, setting up the values, conditions, actions and computations that define the actions of a **single** rule or a **combination** of rules. For this you use the **Rule Composer**, with which you transform each conceptual-level business rule statement into a **logical level, technology-specific tabulated statement** that you can either:

- Generate code from or
- Download to a spreadsheet application such as Microsoft Excel, via a CSV file

**Access**    **Right-click on a Rule Task element | Rule Composer**

Rule Composer Tables

The Rule Composer displays as a view in the central work area, divided into three tables.

Table	Detail	See also
<b>Rule Statements</b>	<p>The Rule Statements table lists the rules associated with the selected Rule Task; you add a rule to the table by dragging an existing Business Rule element from the Project Browser onto an empty row of the table.</p> <p>You do not create new rules within the table.</p>	<a href="#">Add and Remove Rules</a> <sup>[1834]</sup>
<b>Decision</b>	<p>The Decision Table is used to model <i>conditional</i> rules (for example: <i>Cars must not be rented to customers of age less than 18</i>).</p> <p>The table has three sections:</p> <ul style="list-style-type: none"> <li>• Rule Conditions – to model condition variables</li> <li>• Rule Actions – to model action variables</li> <li>• Rule Bindings – to link the rule in the rule table</li> </ul>	<a href="#">Define Rule Conditions</a> <sup>[1836]</sup> <a href="#">Define Rule Actions</a> <sup>[1837]</sup> <a href="#">Bind Rules to Conditions and Actions</a> <sup>[1839]</sup>
<b>Computation Rule</b>	<p>Using the Computation Rule table, you model rules that require a calculation to be performed on the source information, and/or the interaction of rules.</p> <p>The table has the following columns:</p> <ul style="list-style-type: none"> <li>• Computation Rule Actions</li> </ul>	<a href="#">Define Computation Rules</a> <sup>[1840]</sup>

Table	Detail	See also
	<ul style="list-style-type: none"> <li>• Expression</li> <li>• Rule Bindings</li> <li>• Rule Dependency</li> </ul>	






### Notes

- Business Rule Modeling is available in the Business and Software Engineering edition and the Ultimate edition of Enterprise Architect
- To assist with traceability as you complete the relationships across the Rule Composer, selecting an entry in one table automatically highlights the corresponding rows and columns of the other tables; for example, if a Rule Statement is selected, the related rule column in the Decision Table and row in the Computation Rule Table are highlighted

Similarly, if a Computational Rule is selected, the corresponding column in the Decision Table and row in the Rule Statements list are highlighted

- If the table columns are not wide enough to display their contents, you can drag the column header margins to increase the width; all tables on the Decision Table tab and the Computation Rule Tab are linked, so if you increase the column width on one table it changes the width on all tables
- The Rule Composer can be opened and the rules logic viewed in the **Lite edition** of Enterprise Architect

### Learn more

- [Validate Business Rules](#) 
- [Export Composed Rules to CSV](#) 
- [Code Generation for Business Rules](#) 
- [Modeling Business Rules](#) 
- [Develop a Business Rules Model](#) 

### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Business Modeling | Business Rules | Rule Composer**
- (Alt+F1) | **Enterprise Architect | Business Modeling | Business Rules | Rule Statement Table**
- (Alt+F1) | **Enterprise Architect | Business Modeling | Business Rules | Decision Table**
- (Alt+F1) | **Enterprise Architect | Business Modeling | Business Rules | Computation Rule Table**
- (Alt+F1) | **Enterprise Architect | Business Modeling | Business Rules | Rule Composer Example**

#### 9.3.3.2.1 Add and Remove Rules

When you open the Rule Composer for a Business Task, all Business Rules **currently linked** to that task are already listed in the Rule Statements table. You can add further rules that are available in the Project Browser, or remove a rule from the **table** that is no longer applicable to the task.

You do not create or delete the rules themselves in the Rule Composer.

**Access** **Diagram | Right-click Rule Task element | Rule Composer, or**  
**Project Browser | Right-click Rule Task element | Rule Composer**

#### Add a business rule to be associated with the selected Rule Task

When you open the Rules Composer	Action	See also
<b>There are no rules listed</b>	The Rule Statements table has a <b>single</b> empty row. Select the Business Rule in the Project Browser and drag it onto this empty row.  Another empty row is automatically created underneath the first row. Drag the next Business Rule from the Project Browser onto the Rule Statements table.	
<b>There are rules listed</b>	The Rule Statements table has <b>no</b> empty rows.  Click on the <b>No</b> column and select the <b>Add Row</b> context menu option; an empty row is added to the table.  Drag a Business Rule from the Project Browser onto the table, as above.	

When you add a Business Rule to the Rule Statements table, if the Business Rule element is not already on the Business Rules diagram, the element is added to the diagram. A Dependency relationship is created between the 'new' Business Rule and the selected Rule Task.

#### Remove a rule that is no longer required in the Rule Composer

Right-click on the appropriate **No** field and select the **Remove Rule** context menu option. The rule is removed from the Rule Composer and the Dependency relationship with the Rule Task element is deleted.

The Business Rule element is **not** deleted from either the Business Rule diagram or the Project Browser (where, in either case, it might be in use with other Rule Task elements).

#### Learn more

- [Model Business Rules](#)<sup>[1821]</sup>
- [Create a Rule Flow Activity](#)<sup>[1826]</sup>

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Business Modeling | Business Rules | Rule Composer**
- (Alt+F1) | **Enterprise Architect | Business Modeling | Business Rules | Rule Statement Table**
- (Alt+F1) | **Enterprise Architect | Business Modeling | Business Rules | Rule Composer Example**

### 9.3.3.2.2 Define Rule Conditions

When you create the Business Domain model, you set up a number of Classes that define the business terms and entities (such as **Customer**) and their associated attributes and operations. You create the attributes and operations as you set up the Class, with at least some values or parameters, and you tailor some of the features to the rules to define the **conditions** through which a rule takes effect, in the Rule Conditions table.

For example, in a Class that defines the properties of *Car*, the attribute **Type** might be used to set the condition 'Car is: **Small**, **Medium** or **Large**', through which the rules defining base rental charge would be filtered and applied.

**Access** [Diagram](#) | [Right-click Rule Task element](#) | [Rule Composer](#) > [Decision Table](#)

#### Model Rule Conditions

Step	Action	See also
1	<p>From the appropriate Class element in the Project Browser, drag and drop the condition attribute (such as <i>age</i>) or operation (such as <i>IsValidLicense()</i>) onto the first empty field in the <b>Rule Conditions</b> column.</p> <ul style="list-style-type: none"> <li>The <b>Rule Conditions</b> fields apply Intelli-sense to display possible entries for the field; press <b>(Ctrl+Spacebar)</b> in the field to display a list of possible Classes, double-click on the selected Class, type . (a period) and double-click on the attribute or operation in the automatically-displayed list</li> <li>If the Rule Condition is of type <i>enum</i>, the <b>Allowable Values</b> field is automatically set with the <i>enum</i> literals; this procedure then ends here.</li> </ul>	
2	Determine a range of accepted values for the Rule Condition.	
3	<p>Right-click on the <b>Allowable Values</b> column and select:</p> <ul style="list-style-type: none"> <li>For an attribute, the <b>Edit Allowable Values</b> context menu option; the Edit Allowable Values dialog displays</li> <li>For an operation, the <b>Edit Parameters</b> context menu option; the Edit Parameters dialog displays (see step 5)</li> </ul> <p>Type each required value or range of values in the <b>Value</b> field, and click on the <b>Save</b> button to display the value in the <b>Allowable Values</b> list box</p> <p><i>age</i> could have the values:</p> <p><b>&lt;18</b>  <b>&gt;18 and &lt;50</b>  <b>&gt;50</b></p> <p><i>IsValidLicense()</i> could return:</p> <p><b>True</b>  <b>False</b></p>	
4	Click on the <b>OK</b> button to save the values and close the dialog; for an attribute, a new constraint <i>AllowableValues</i> is created.	

Step	Action	See also
	<ul style="list-style-type: none"> <li>You can check this constraint by opening the Properties dialog for the attribute and selecting the Constraints tab</li> <li>If the Rule Condition references an enumeration, the enum literals are not editable in the Edit Allowable Values dialog</li> </ul>	
5	<p>If the Rule Condition is an operation, you can pass parameters to it.</p> <p>Right click on the <b>Allowable Values</b> field, and select the <b>Edit Parameters</b> context menu option; the Edit Parameters dialog displays, listing the parameters that already exist for the operation.</p> <p>Select the parameters and type their values into the <b>Value</b> text box; click on the <b>OK</b> button to cancel the dialog.</p> <ul style="list-style-type: none"> <li>You can add an operation as a Rule Condition more than once, so you can call the operation with different sets of parameters</li> </ul>	
6	<p>To add another Rule Condition, right-click on the <b>No</b> column and select the <b>Add Row</b> context menu option; an empty row is added to the table.</p> <p>To remove a Rule Condition from the table, right-click on the appropriate <b>No</b> field and select the <b>Delete Row</b> context menu option.</p> <ul style="list-style-type: none"> <li>If the condition is based on an attribute, this does not affect either the original attribute <b>or the new constraint</b> in the model; you can either re-use the attribute with its constraint, or use the attribute Properties dialog to remove the constraint</li> </ul>	
7	If any of the condition values invoke an action or a decision, you can define it as a <b>Rule Action</b> .	<a href="#">Define Rule Actions</a> <sup>[1837]</sup>

Learn more

- [Create a Business Domain Model](#)<sup>[1824]</sup>
- [Pass Parameters to Rule Flow Activity](#)<sup>[1829]</sup>

Learning Center topics

- (Alt+F1) | **Enterprise Architect | Business Modeling | Business Rules | Rule Composer**
- (Alt+F1) | **Enterprise Architect | Business Modeling | Business Rules | Decision Table**
- (Alt+F1) | **Enterprise Architect | Business Modeling | Business Rules | Rule Composer Example**

**9.3.3.2.3 Define Rule Actions**

In the Classes in the Business Domain model, the attributes and operations that you set up to define the **conditions** through which a rule takes effect might in their turn call a decision attribute or an operation, which you can assign as an **action** in the **Rule Actions (Outcome)** table. You create the attributes and operations as you set up the Class, with at least some values or parameters, and you tailor some of these features to define the actions that determine the **effect** of the rule.

**Access** [Diagram](#) | [Right-click Rule Task element](#) | [Rule Composer](#) > [Decision Table](#)

### Model Rule Actions

Step	Action	See also
1	<p>From the appropriate business term Class element in the Project Browser, drag and drop the decision attribute (such as <i>Eligible - Yes/No</i>) or operation (such as <i>post error</i>) onto the first empty field in the <b>Rule Actions (Outcome)</b> column.</p> <ul style="list-style-type: none"> <li>The <b>Rule Actions</b> fields apply Intelli-sense to display possible entries for the field; press (<b>Ctrl+Spacebar</b>) in the field to display a list of possible Classes, double-click on the selected Class, type . (a period) and double-click on the attribute or operation in the automatically-displayed list</li> </ul>	
2	<p>For an <b>attribute</b>, if the dropped action variable is of type <i>enum</i>, the <b>Allowable Values/Parameters</b> fields are automatically set with the enum literals.</p> <p>Otherwise, double-click on the <b>Allowable Values/Parameters</b> field (or right-click and select the <b>Edit Allowable Values</b> menu option); the Edit Allowable Values dialog displays.</p> <p>Click on the <b>New</b> button, type a possible value in the text box (such as <b>Yes</b> or <b>Accept</b>), click on the <b>Save</b> button, and repeat the process for the next possible value (such as <b>No</b> or <b>Reject</b>).</p>	
3	<p>When you have entered all possible values, close the dialog and select the appropriate response in the <b>Result&lt;n&gt;</b> column field underneath the appropriate <b>Value&lt;n&gt;</b> field for the calling condition.</p>	
4	<p>For an <b>operation</b>, a checkbox displays in each of the <b>Result&lt;n&gt;</b> column fields; to call the operation as the response to a value of the corresponding condition, select the checkbox in the <b>Result&lt;n&gt;</b> field underneath that condition <b>Value &lt;n&gt;</b> field.</p>	
5	<p>To pass <b>parameters</b> to the operation, double-click on the <b>Allowable Values/Parameters</b> field (or right-click and select the <b>Edit Parameters</b> menu option); the Edit Parameters dialog displays.</p> <p>Select the parameters and type the values into the <b>Value</b> text box; click on the <b>Save</b> button and close the dialog.</p> <p>You can add an operation as a Rule Action more than once, so you can call the operation with different sets of parameters.</p>	
6	<p>Click on the <b>Save</b> button in the Rule Composer toolbar to save the values.</p>	
7	<p>To <b>add</b> another Rule Action, right-click on the <b>No</b> column and select the <b>Add Row</b> context menu option; an empty row is added to the table.</p> <p>To <b>remove</b> a Rule Action from the table, right-click on the appropriate <b>No</b> field and select the <b>Delete Row</b> context menu option; this does not affect the original attribute or operation in the model.</p>	

Step	Action	See also

### Learn more

- [Create a Business Domain Model](#)<sup>[1824]</sup>
- [Pass Parameters to Rule Flow Activity](#)<sup>[1829]</sup>

### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Business Modeling | Business Rules | Rule Composer**
- (Alt+F1) | **Enterprise Architect | Business Modeling | Business Rules | Rule Composer Example**

#### 9.3.3.2.4 Bind Rules to Conditions and Actions

After defining which rules are applied, which conditions are evaluated and/or which actions are taken in the completion of a Rule Task, you bind the rules to the conditions and actions to define:

- Which rules are evaluated against a condition
- What values of that condition are applicable to each rule
- What decisions or actions are taken on each value output from evaluation of the rule, and what the result of that decision or action is

You bind the rules, conditions and actions using the Rule Bindings table, just above the Rule Conditions table.

**Access**   **Diagram | Right-click Rule Task element | Rule Composer > Decision Table**

### Bind rules

Step	Action	See also
1	Click on the <b>Rule Bindings</b> field immediately above the appropriate <b>Value&lt;n&gt;</b> and/or <b>Result&lt;n&gt;</b> field.  Click on the number of the rule that is evaluated against the condition for the specific value and to generate the action to provide the specific result identified in those two fields.	<a href="#">Define Rule Conditions</a> <sup>[1836]</sup>  <a href="#">Define Rule Actions</a> <sup>[1837]</sup>
2	Ensure that the values set in the <b>Value&lt;n&gt;</b> and/or <b>Result&lt;n&gt;</b> field underneath the rule number do all operate under the rule.	
3	Click on the <b>Save</b> icon in the Rule Composer toolbar.  Continue to bind business rules to conditions and actions having logical values for an instance of the rule.	

**Example**

- Rule 2 is *A car must not be rented to a customer of age less than 18*
- The first Rule Condition is *Customer.age*
- The **Value1** field contains the value **< 18** against that Rule Condition
- The first Rule Action (or decision) is *Customer.Eligible*
- The **Result1** field contains the value **No** against that Rule Action
- The second Rule Action is *Application.Status*
- The **Result1** field contains the value **Reject** against that Rule Action

You would select **2** in the **Rule Bindings** field over the **Value1/Result1** column, so that a when a customer's application to rent a car is tested against rule 2, if the customer is found to be under 18 they are automatically classified as ineligible and the application rejected.

**Learn more**

- [Model Business Rules](#) 

**Learning Center topics**

- (Alt+F1) | **Enterprise Architect | Business Modeling | Business Rules | Rule Composer**
- (Alt+F1) | **Enterprise Architect | Business Modeling | Business Rules | Decision Table**
- (Alt+F1) | **Enterprise Architect | Business Modeling | Business Rules | Rule Composer Example**


**9.3.3.2.5 Define Computation Rules**

In a business process, some actions and effects are dependent on the interaction of rules with either another rule or a calculation. For example:

- Rule 1) *Cars cannot be rented to customers aged under 18 years*, Rule 2) *Customers must have a valid driving licence* - Rule 1 must be satisfied before Rule 2 is tested
- Rule 4) *Penalty of 10% of total rental cost applies to customers with Bad History Level 1* - If the customer has a Bad history Level 1, multiply the rent payable value by 1.1

**Access** **Diagram | Right-click Rule Task element | Rule Composer > Computation Rule Table**

**Define a computation rule**

Step	Action	See also
1	<p>In the Project Browser, expand the appropriate business entity Class in the Business Domain model, and drag the attribute that represents the computed action into the <b>Computation Rule Actions</b> field.</p> <p>Alternatively, you can use Intelli-sense in both the <b>Computation Rule Actions</b> field and the <b>Expression</b> field to display a list of possible Classes and attributes:</p> <ol style="list-style-type: none"> <li>1. Press <b>(Ctrl+Spacebar)</b> in the field to display the list of Classes.</li> <li>2. Double-click on the required Class, and type . (period) after it; a list of attributes</li> </ol>	<a href="#">Create a Business Domain Model</a> 



Step	Action	See also
	for that Class displays automatically. 3. Double-click on the required attribute.	
2	In the <b>Expression</b> field, complete the expression to be evaluated. For example: <ul style="list-style-type: none"> <li>If computing Rule 4 above, and you have selected <i>Rent.RentPayable</i> in the <b>Expression</b> field, then type <b>*1.1</b> immediately after it</li> </ul>	
3	In the <b>Rule Bindings</b> field, click on the drop-down arrow and select the number of the rule being modeled (as listed in the Rule table) to link the computation data to the rule.	<a href="#">Add and Remove Rules</a> <sup>[1834]</sup>
4	If the rule depends on another rule being satisfied first (as for Rule 2 above), type the number of that rule in the <b>Rule Dependency</b> field.	
5	Click on the <b>Save</b> icon in the Rule Composer toolbar to save the computation rule.  If the computation rule is also a Rule Conditions rule, add the condition variable in the Decision table and bind the appropriate rule in the Rule Bind section.	<a href="#">Define Rule Conditions</a> <sup>[1836]</sup>  <a href="#">Bind Rules to Conditions and Actions</a> <sup>[1839]</sup>

Learn more

- [Model Business Rules](#) <sup>[1821]</sup>

Learning Center topics


- (Alt+F1) | [Enterprise Architect](#) | [Enterprise Architect](#) | [Business Modeling](#) | [Business Rules](#) | [Rule Composer](#)
- (Alt+F1) | [Enterprise Architect](#) | [Enterprise Architect](#) | [Business Modeling](#) | [Business Rules](#) | [Computation Rule Table](#)
- (Alt+F1) | [Enterprise Architect](#) | [Enterprise Architect](#) | [Business Modeling](#) | [Business Rules](#) | [Rule Composer Example](#)

**9.3.3.2.6 Validate Business Rules**

Before you generate code for the Rule Task elements, or export them to a CSV file, it is recommended that you validate the business rules in the Rule Composer. When you do this, the business rules on the Rule Composer are parsed and any errors or warnings that might indicate incomplete or unfavorable code generation are displayed on the RuleComposer Validation tab on the system Output window.

Access    [Diagram](#) | [Right-click Rule Task element](#) | [Rule Composer](#)

**Validate Rules**

Step	Detail	See also
1	<p>Click on the <b>Validation</b> icon (green tick) in the Rule Composer toolbar:</p>  <p>The Output window displays, showing the status of the validation - <i>Validation complete</i> plus any validation errors or warnings - on the RuleComposer Validation tab.</p>	
2	Double-click on each warning or error message in turn to highlight and investigate the faulty data that caused that message, in the Rule Composer.	

**Learn more**

- [Compose Business Rules](#) <sup>[1833]</sup>
- [Export Composed Rules to CSV](#) <sup>[1842]</sup>
- [Code Generation For Business Rules](#) <sup>[1843]</sup>
- [The System Output Window](#) <sup>[169]</sup>

**Learning Center topics**


- (Alt+F1) | **Enterprise Architect | Business Modeling | Business Rules | Validate Business Rules**

**9.3.3.2.7 Export Composed Rules to CSV**

It is possible to export composed Business Rules to an external spreadsheet - such as Microsoft Excel - as a .csv file. You can then maintain the spreadsheet in that tool for use as a reference for the organization, if you have not created an application to automatically apply those rules.

**Access**    **Diagram | Right-click Rule Task element | Rule Composer**

**Export the contents of the Rule Composer to a CSV file**

Step	Action	See also
1	<p>Click on the <b>Export to CSV</b> icon () in the Rule Composer toolbar.</p> <p>The Windows Browser dialog displays.</p>	
2	Browse to the required file location and type in a name for the .csv file to export the data into.	

Step	Action	See also
3	Click on the <b>Save</b> button to export the data.	

#### Learn more

- [CSV Import and Export](#)<sup>[497]</sup>
- [Compose Business Rules](#)<sup>[1833]</sup>

#### 9.3.3.2.8 Code Generation For Business Rules

After you have modeled rules for all Rule Task elements in the Rule Flow diagram, the Business Domain model is ready for **code transformation**. The internal code templates provided for generating technology-specific rule code work hand-in-hand with the **EASL code templates** to generate the code from the rules-processing Class and its Rule Flow structure.

#### Return a value from the Rule Flow behavior

Step	Action	See also
1	Double-click on the last <b>Rule Task</b> element before the end node of the Rule Flow diagram.  The element's Properties dialog displays.	<a href="#">Create a Business Domain Model</a> <sup>[1824]</sup>
2	Click on the Effect tab.	
3	In the <b>Effect</b> field, type the <i>return</i> statement; for example, <i>return true</i> .	
4	Click on the <b>Save</b> button, and on the <b>OK</b> button to close the dialog.  Generate code for the Class containing the rule flow behavior (in our initial example, <i>Rental System</i> ); the code for business rules logic is generated, with the rule statements expressed in natural language as comments.	<a href="#">Generate code for a single class</a> <sup>[2113]</sup>

#### Example

The following code snippet was generated from the *Rental System* Class element in the *EAExample* model:

```

////////////////////////////////////
// RentalSystem.cs
// Implementation of the Class RentalSystem
// Generated by Enterprise Architect
// Created on:      08-May-2012 2:39:23 PM
////////////////////////////////////

```

```

public class RentalSystem {

    public Customer m_Customer;
    public Car m_Car;
    public Rent m_Rent;

    public RentalSystem() {

    }

    ~RentalSystem() {

    }

    public virtual void Dispose() {

    }

    /* Begin - EA generated code for Activities and Interactions */

    public bool ProcessApplication(Rent m_rent, Application m_application)
    {
        // behavior is an Activity

        /* CAR MUST NOT BE RENTED TO CUSTOMERS WITHOUT A VALID
L I C E N C E N U M B E R */
        if ( m_Customer.ValidLicenceNumber == "FALSE" )
        {
            m_application.Status = "Reject";
            m_Customer.Eligible = false;
        }
        /* CAR MUST NOT BE RENTED TO CUSTOMERS OF AGE LESS THAN
18 */
        if ( m_Customer.age < 18 )
        {
            m_application.Status = "Reject";
            m_Customer.Eligible = false;
        }
        /* CAR MUST NOT BE RENTED TO CUSTOMERS WITH BAD HI STORY
L E V E L 3 */
        if ( m_Customer.BadHistoryLevel == 3 )
        {
            m_application.Status = "Reject";
            m_Customer.Eligible = false;
        }
        if ( Customer.Eligible == true)
        {

            /* RENT FOR SMALL CARS IS 80 AUD PER DAY */
            if ( m_Car.type == Small )
            {
                m_rent.Rent Per Day = 80;
            }
            /* RENT FOR AWD CARS IS 100 AUD PER DAY */
            if ( m_Car.type == AWD )
            {
                m_rent.Rent Per Day = 100;
            }
            /* RENT FOR LUXURY CARS IS 150 AUD PER DAY */
            if ( m_Car.type == Luxury )
            {
                m_rent.Rent Per Day = 150;
            }
            /* RENT PAYABLE IS CALCULATED AS THE PRODUCT OF
RENT PER DAY AND RENTAL PERIOD IN DAYS */
            m_rent.Rent Payable = m_rent.Rent Per Day * m_rent.
No_of_rent_days;
            if ( Customer.BadHistoryLevel > 0)
            {

```

```

/* PENALTY OF 20 % OF RENT MUST BE APPLIED FOR
CUSTOMERS WITH BAD HISTORY LEVEL 2*/
if( m_Customer.BadHistoryLevel == 2 )
{
    m_rent.PenaltyFee = m_rent.RentPayable
* 0.2;
}
/* PENALTY OF 10 % OF RENT MUST BE APPLIED FOR
CUSTOMERS WITH BAD HISTORY LEVEL 1*/
if( m_Customer.BadHistoryLevel == 1 )
{
    m_rent.PenaltyFee = m_rent.RentPayable
* 0.1;
}
}
else
{
}

/*TOTAL AMOUNT PAYABLE IS CALCULATED AS THE SUM OF
RENT PAYABLE AND PENALTY IF ANY.*/
m_rent.TotalAmountPayable = m_rent.RentPerDay +
m_rent.PenaltyFee;
}
else
{
}
return m_application.Status;
}

/* End - EA generated code for Activities and Interactions */
} // end Rental System

```

### Notes

- Business Rule Modeling is available in the Business and Software Engineering edition and the Ultimate edition of Enterprise Architect

### Learn more

- [EASL code templates](#) <sup>[1688]</sup>
- [Compose Business Rules](#) <sup>[1833]</sup>
- [Validate Business Rules](#) <sup>[1841]</sup>

### Learning Center topics

- (Alt+F1) | [Enterprise Architect](#) | [Business Modeling](#) | [Business Rules](#) | [Executable Business Rules](#)

## 9.3.4 BPMN Models

If you need to model the activity of a business, capturing the behavior and the information flows within the organization or system, you can do so using the Business Process Modeling Notation (BPMN). BPMN is specifically targeted at the business modeling community and has a direct mapping to UML through BPMN Profiles integrated with the Enterprise Architect installer. Through use of these profiles, you can develop BPMN diagrams quickly and simply, including:

- Maintain existing diagrams created in BPMN 1.0 format, and create new diagrams in BPMN 1.0

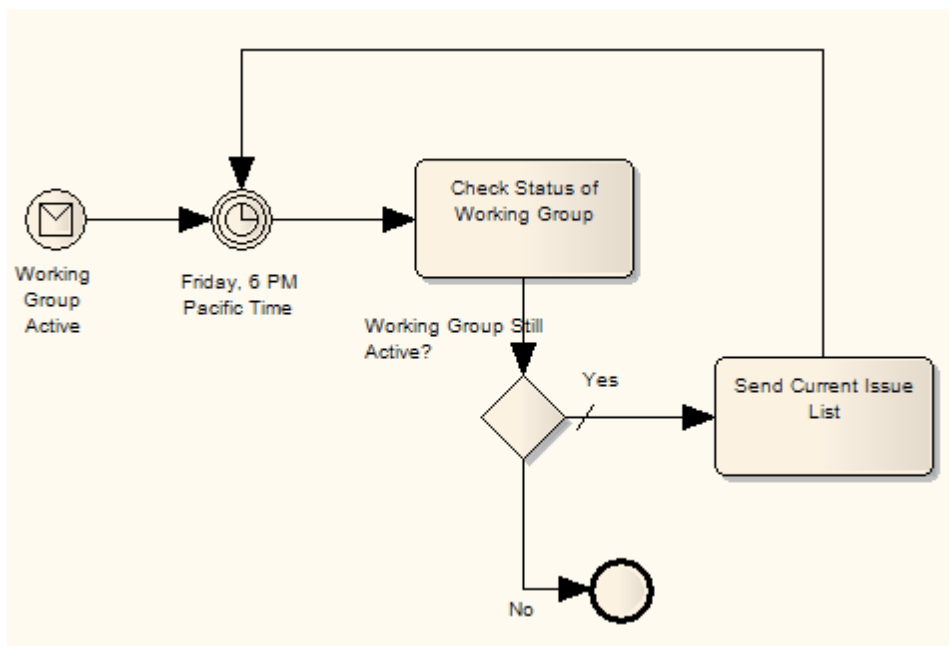
- Create and maintain diagrams in the BPMN 1.1 and BPEL formats
- Create and maintain diagrams in the BPMN 2.0 and BPEL formats
- Migrate a BPMN 1.0 model (or part of a model) to BPMN 1.1, and a BPMN 1.1 model (or part of a model) to BPMN 2.0

The Enterprise Architect installer for releases later than 8.0 provides you with separate versions of the MDG Technology for BPMN that support BPMN versions 1.0, 1.1 and 2.0, and BPEL.

**Access** BPMN facilities are provided in the form of:

- A BPMN diagram type, accessed through the New Diagram dialog
- BPMN pages in the Toolbox
- BPMN element and relationship entries in the Toolbox **Shortcut** menu and Quick Linker.

Specifications of BPMN elements and relationships are defined by Tagged Values; for example, to define the *Message*, *Timer* and *Default Path (/)* symbols in this diagram:



### BPMN 2.0 Diagram Orientation

On a **BPMN 2.0** diagram, you can set the flow orientation to either horizontal or vertical, or not apply an orientation.

To set or clear the orientation, right click on the diagram background and click on the **Set Diagram Flow Direction** context menu option. Then click on either:

- **None** (the default, no specific orientation set)
- **Horizontal** (diagram flow across the page, Pool and Lane elements occupy the full width of the diagram), or
- **Vertical** (diagram flow down the page, Pool and Lane elements occupy the full height of the diagram)

### **BPMN 2.0 Sequence Flow and Message Flow Rules**

Rules concerning the use of Sequence Flows and Message Flows, as defined in the *BPMN 2.0 Specification* and described below, are implemented by the MDG Technology for BPMN 2.0.

The following rules apply to **Sequence Flows** in relation to Events (Start, Intermediate and End), Activities (Task and Sub-Process, for Processes), Choreography Activities (Choreography Task and Sub-Choreography, for Choreographies) and Gateways:

- A Sequence Flow cannot cross a Pool boundary
- An End Event, **unless** edge-mounted, **cannot** be the source element for a Sequence Flow
- A Start Event, **unless** edge-mounted, **cannot** be the target element for a Sequence Flow
- An Intermediate Event, **if** edge mounted, **cannot** be the source element for a Sequence Flow; it cannot have incoming Sequence Flows
- An Intermediate Event - **if** edge mounted **and** having the Tagged Value *eventDefinition=Compensation*, cannot be either the source or target element for a Sequence Flow
- Objects **within** a Sub-Process cannot have a Sequence Flow relationship with objects **outside** the Sub-Process
- A Sequence Flow cannot connect directly to a Pool

The following rules apply to **Message Flows** in relation to Events (Start, Intermediate and End), Activities (Task and Sub-Process, for Processes), Choreography Activities (Choreography Task and Sub-Choreography, for Choreographies) and Pools:

- Message Flows **can be** created from objects in one Pool to objects in another Pool; Message Flows can connect directly to another Pool
- A Message Flow can connect to Events (Start, Intermediate and End) **only** if they have the Tagged Value *eventDefinition=Message* or *eventDefinition=Multiple*
- A Start Event **cannot be** the source element for a Message Flow
- An End Event **cannot be** the target element for a Message Flow

### **BPMN Toolbox Pages**





You can access the BPMN Toolbox pages through the **Diagram | Diagram Toolbox: More tools...** option; select the **BPMN 1.0**, **1.1** or **2.0** options as appropriate.

You can also set BPMN as the active default technology to access the Toolbox pages directly.

### **Disable BPMN**

If you prefer not to use BPMN in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog (**Settings | MDG Technologies**).

### **Learn more**

- [BPMN 1.0 and 1.1 Toolbox Pages](#) 
- [BPMN 2.0 Toolbox Pages](#) 
- [BPMN 2.0 XML](#) 
- [Migrate a BPMN 1.0 Model to BPMN 1.1](#) 

- [Migrate a BPMN 1.1 Model to BPMN 2.0](#)<sup>[1868]</sup>
- [Change BPMN Element Appearance](#)<sup>[1862]</sup>
- [BPEL Models](#)<sup>[1870]</sup>
- [New Diagram dialog](#)<sup>[822]</sup>
- [Set Technology as Active Default](#)<sup>[1479]</sup>

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Modeling Languages | Business | BPMN | Introduction**
- (Alt+F1) | **Enterprise Architect | Business Modeling | BPMN | Getting Started**

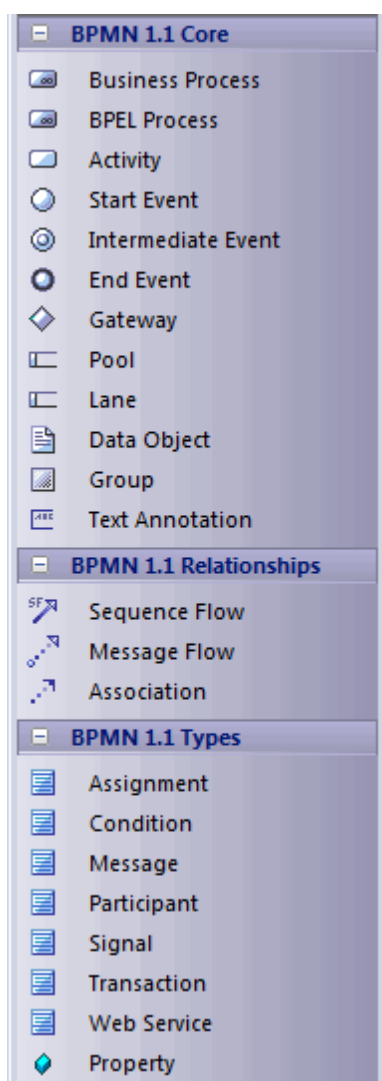
### **9.3.4.1 BPMN 1.0 and 1.1 Toolbox Pages**

You can create graphical (Core) and non-graphical (Types) BPMN elements and relationships on business process diagrams in **BPMN 1.0**, **BPMN 1.1** and **BPEL** formats, using the BPMN 1.0 and BPMN 1.1 pages of the Diagram Toolbox.

Access    **Diagram | Diagram Toolbox: More tools | BPMN 1.0**  
**Diagram | Diagram Toolbox: More tools | BPMN 1.1**

#### Toolbox





### Element and Connector Descriptions

Page	Item	Action	See also
<b>Core</b>	<b>Business Process</b>	Extend a <i>composite Activity</i> that defines a business process.	
	<b>BPEL Process</b>	Define the behavior of an executable or abstract business process.	
	<b>Activity</b>	<p>Represents work that is performed within a <i>Business Process</i>. An <i>Activity</i> can be modeled as a:</p> <ul style="list-style-type: none"> <li>• <b>Sub-Process</b> - a compound <i>Activity</i> that is defined as a flow of other <b>BPMN 1.1</b> elements or</li> <li>• <b>Task</b> - an atomic <i>Activity</i> that cannot be broken down into a smaller unit</li> </ul>	<a href="#">Composite Elements</a> <sup>936</sup>

Page	Item	Action	See also
		As a sub-process, the Activity can be made a composite element that links to a child diagram containing the flow of other BPMN elements.	
	<b>Start Event</b>	Define the initiating event in a process.	
	<b>Intermediate Event</b>	Define an intermediate event in a process.	
	<b>End Event</b>	Define the terminating event in a process.	
	<b>Gateway</b>	Define a decision point in a business process. If a condition is <b>true</b> , then processing continues one way; if not, then another.	
	<b>Pool</b>	Extend a <i>Partition</i> element to logically organize an Activity.	
	<b>Lane</b>	Extend a <i>Partition</i> element to subdivide a Pool.	
	<b>Data Object</b>	Extend an <i>Artifact</i> element to define a physical piece of information used or produced by a system.	
	<b>Group</b>	Extend a <i>Boundary</i> element to group other elements.	
	<b>Text Annotation</b>	Create a comment.	
<b>Relationships</b>	<b>Sequence Flow</b>	Extend a <i>Control Flow</i> relationship to define the flow of activity.	
	<b>Message Flow</b>	Extend a <i>Control Flow</i> relationship to define the flow of communications in the process.	
	<b>Association</b>	Associate information and artifacts with flow objects.	
<b>Types</b>	<b>Assignment</b>		

Page	Item	Action	See also
	<b>Condition</b>	Define the <i>properties</i> (Tagged Values) of the Core BPMN 1.1 elements such as Activities, Events and Gates.	<a href="#">Change BPMN Element Appearance</a> <sup>[1862]</sup>
	<b>Message</b>		
	<b>Participant</b>		
	<b>Signal</b>		
	<b>Rule</b>		
	<b>Transaction</b>		
	<b>Web Service</b>		
	<b>Property</b>		

### 9.3.4.2 BPMN 2.0 Toolbox Pages

You can create BPMN elements and relationships on diagrams in **BPMN 2.0** and **BPEL** formats using the **BPMN 2.0 pages** of the Diagram Toolbox.

**Access** [Diagram](#) | **Diagram Toolbox: More tools** | **BPMN 2.0**

Enterprise Architect provides a set of pages for each of the following BPMN 2.0 diagram types:

Topic	See also
<b>Business Process</b> - contains the constructs needed to build Business Process models.	<a href="#">BPMN 2.0 Business Process Toolbox Pages</a> <sup>[1852]</sup>
<b>Choreography</b> - contains the constructs needed to build Choreography models.	<a href="#">BPMN 2.0 Choreography Toolbox Pages</a> <sup>[1854]</sup>
<b>Collaboration</b> - contains the constructs required to build Collaboration models.	<a href="#">BPMN 2.0 Collaboration Toolbox Page</a> <sup>[1856]</sup>
<b>Conversation</b> - contains the constructs needed to build Conversation models.	<a href="#">BPMN 2.0 Conversation Toolbox Pages</a> <sup>[1859]</sup>

Topic	See also
<b>Types</b> - contains the constructs common to all BPMN 2.0 diagrams.	<a href="#">BPMN 2.0 Type Toolbox Page</a> <sup>1861</sup>

#### 9.3.4.2.1 BPMN 2.0 Business Process Toolbox Pages

To create BPMN elements and relationships on **business process** diagrams in either BPMN 2.0 or BPEL formats, you can use the BPMN 2.0 **Business Process** pages of the Diagram Toolbox.

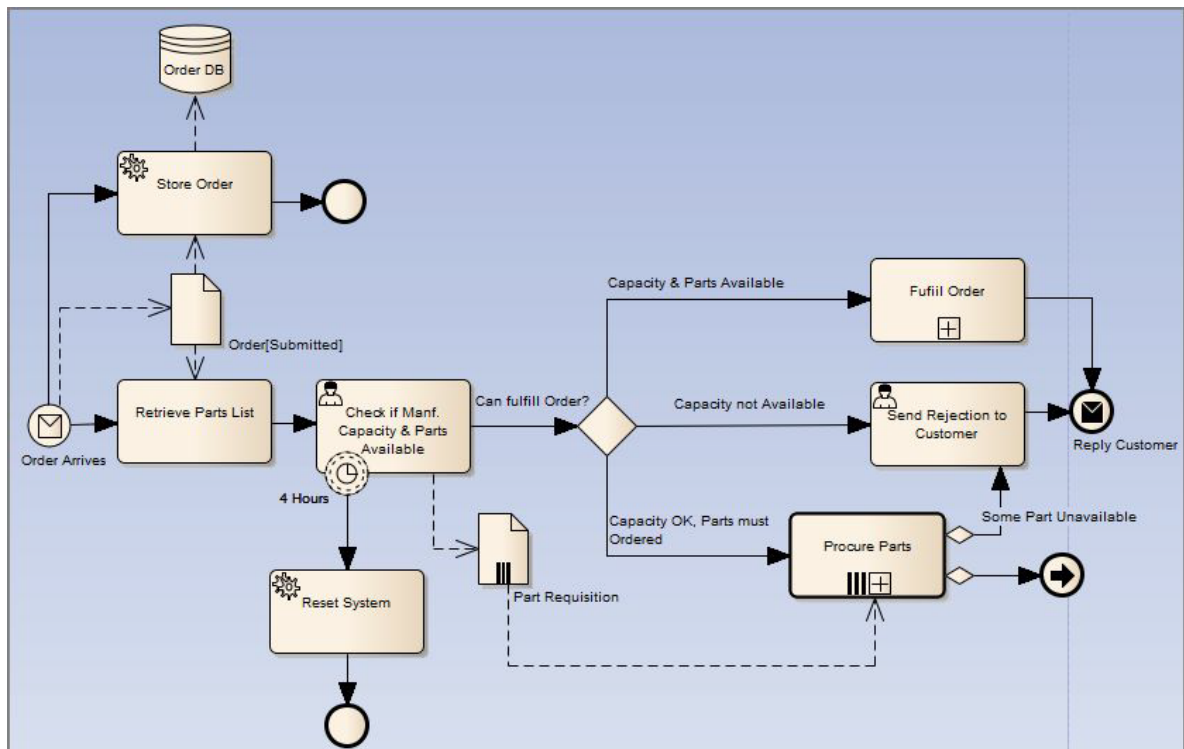
Access   **Diagram | Diagram Toolbox: More tools | BPMN 2.0 | BPMN 2.0 Business Process**

#### Element and Connector Descriptions

Page	Item	Use to
<b>BPMN 2.0 Business Process</b>	<b>Business Process</b>	Extend a composite Activity that defines a business process.
	<b>BPEL</b>	Define the behavior of an executable or abstract business process.
	<b>Activity</b>	Represents work that is performed within a <i>Business Process</i> . An <i>Activity</i> can be modeled as a: <ul style="list-style-type: none"> <li>• <b>Sub-Process</b> - a compound <i>Activity</i> that is defined as a flow of other <b>BPMN 2.0</b> elements or</li> <li>• <b>Task</b> - an atomic <i>Activity</i> that cannot be broken down into a smaller unit</li> </ul> <p>As a sub-process, the Activity can be made a composite element that links to a child diagram containing the flow of other BPMN elements.</p>
	<b>Choreography</b>	Extend an Activity element to represent a process unit of information exchange between Participants.  Enterprise Architect currently supports two to four participant bands within one Choreography Activity/Task.
	<b>Conversation</b>	Extend a Class element to group a set of Message Flows together based on a certain concept.
	<b>Data Object</b>	Provide or store the information for an Activity.
	<b>Data Store</b>	Represent a mechanism for an Activity to retrieve or update stored information.

Page	Item	Use to
	<b>Start Event</b>	Define the initiating event in a process.
	<b>Intermediate Event</b>	Define an intermediate event in a process.
	<b>End Event</b>	Define the terminating event in a process.
	<b>Gateway</b>	Define a decision point in a business process. If a condition is <b>true</b> then processing continues one way; if <b>false</b> , then another.
	<b>Pool</b>	Extend a Partition element to logically organize an Activity.
	<b>Lane</b>	Extend a Partition element to subdivide a Pool.
	<b>Message</b>	Represent the contents of a communication between two Participants.
	<b>Group</b>	Extend a Boundary element to group other elements.
	<b>Text Annotation</b>	Create a comment.
<b>BPMN 2.0 Business Process Connectors</b>	<b>Sequence Flow</b>	Extend a Control Flow relationship to define the flow of activity.
	<b>Message Flow</b>	Extend a Control Flow relationship to define the flow of communications in the process.
	<b>Data Association</b>	Move data between Data Objects, Data Store, Properties and Activities, Processes.
	<b>Association</b>	Link the information and Artifacts with BPMN graphic elements.
	<b>Conversation Link</b>	Connect a Conversation Node with a Participant, in either direction.

Example BPMN 2.0 Business Process Diagram



### Notes

- The appearance and specification of some elements and connectors are defined by Tagged Values

### Learn more

- [Composite Elements](#) <sup>[936]</sup>
- [Change BPMN Element Appearance](#) <sup>[1862]</sup>

### Learning Center topics

- (Alt+F1) | [Enterprise Architect](#) | [Modeling Languages](#) | [Business](#) | [BPMN](#) | [Business Process Toolbox](#)
- (Alt+F1) | [Enterprise Architect](#) | [Business Modeling](#) | [BPMN](#) | [Process Modeling](#)

#### 9.3.4.2.2 BPMN 2.0 Choreography Toolbox Pages

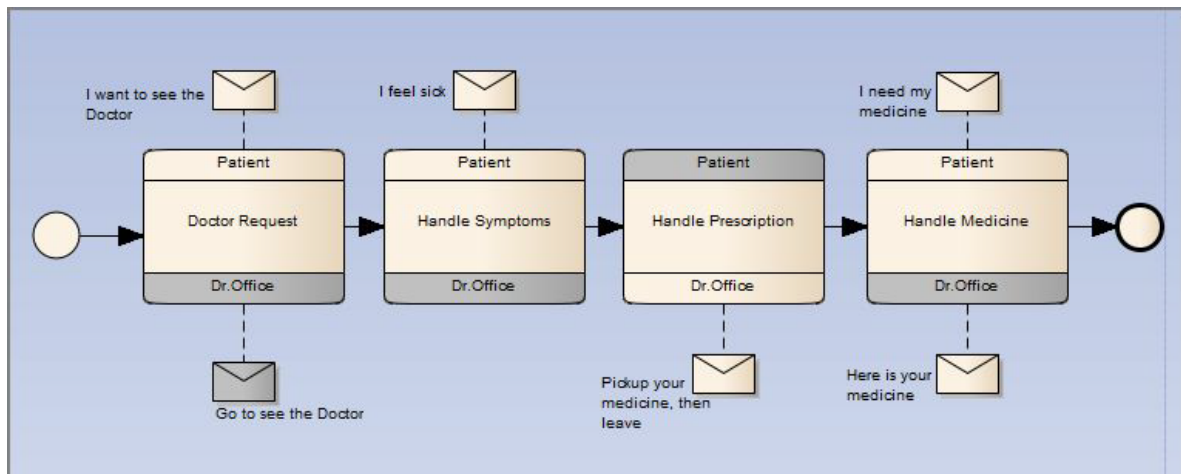
To create BPMN elements and relationships on **Choreography** diagrams in either BPMN 2.0 or BPEL formats, you can use the BPMN 2.0 **Choreography** pages of the Diagram Toolbox.

**Access** [Diagram](#) | [Diagram Toolbox: More tools](#) | [BPMN 2.0](#) | [BPMN 2.0 Choreography](#)

### Element and Connector Descriptions

Page	Item	Use to
<b>BPMN 2.0 Choreography</b>	<b>Choreography Model</b>	Extend a composite Activity that defines a Choreography process.
	<b>Choreography</b>	Extend an Activity element to represent a process unit of information exchange between Participants.  Enterprise Architect currently supports two to four participant bands within one Choreography Activity/Task.
	<b>Start Event</b>	Define the initiating event in a process.
	<b>Intermediate Event</b>	Define an intermediate event in a process.
	<b>End Event</b>	Define the terminating event in a process.
	<b>Gateway</b>	Define a decision point in a business process.  If a condition is <b>true</b> then processing continues one way; if <b>false</b> , then another.
	<b>Message</b>	Represent the contents of a communication between two Participants.
	<b>Pool</b>	Extend a Partition element to logically organize an Activity.
	<b>Text Annotation</b>	Create a comment.
<b>BPMN 2.0 Choreography Connectors</b>	<b>Sequence Flow</b>	Define the order of activity in a Choreography.
	<b>Association</b>	Link the information and Artifacts with BPMN graphic elements.
	<b>Message Flow</b>	Extend a Control Flow relationship to define the flow of communications in the process.

#### Example BPMN 2.0 Choreography Diagram



### Notes

- The appearance and specification of some elements and connectors are defined by Tagged Values

### Learn more

- [Change BPMN Element Appearance](#)<sup>[1862]</sup>

### Learning Center topics

- (Alt+F1) | [Enterprise Architect](#) | [Modeling Languages](#) | [Business](#) | [BPMN](#) | [Choreography](#)
- (Alt+F1) | [Enterprise Architect](#) | [Business Modeling](#) | [BPMN](#) | [Choreography Modeling](#)

#### 9.3.4.2.3 BPMN 2.0 Collaboration Toolbox Page

To create BPMN elements and relationships on **Collaboration** diagrams in either BPMN 2.0 or BPEL formats, you can use the BPMN 2.0 **Collaboration** pages of the Diagram Toolbox.

**Access** [Diagram](#) | [Diagram Toolbox: More tools](#) | [BPMN 2.0](#) | [BPMN 2.0 Collaboration](#)

### Element and Connector Descriptions

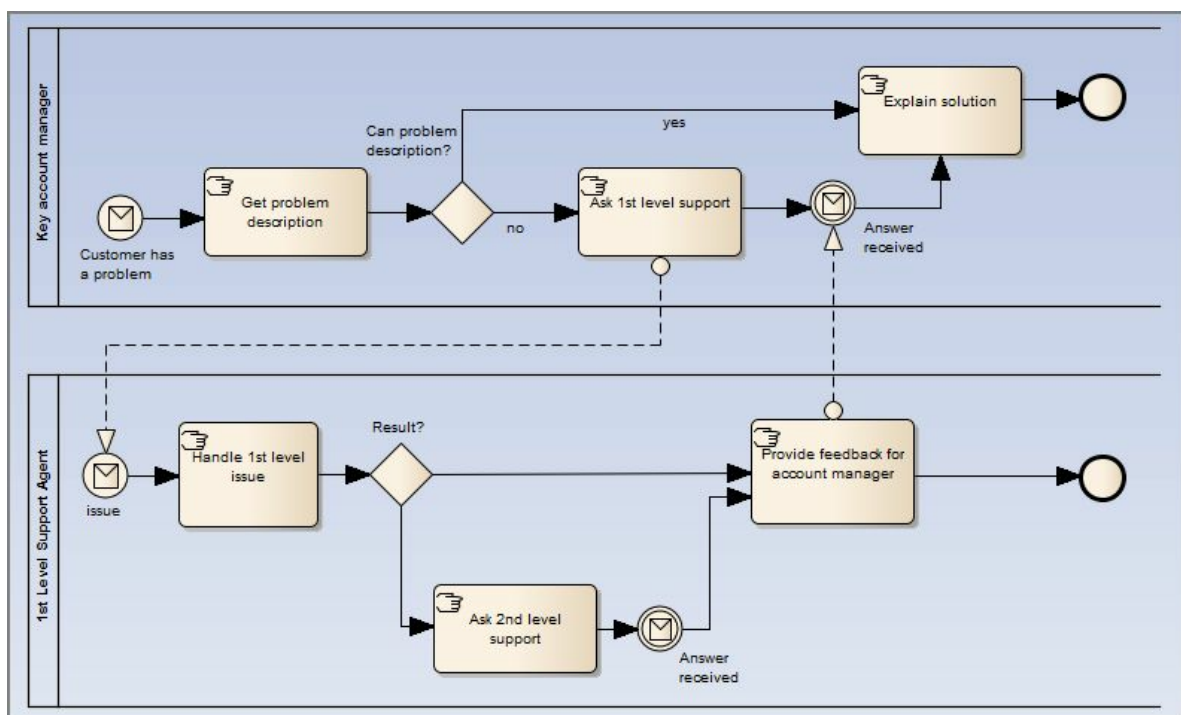
Page	Item	Use to
<b>BPMN 2.0 Collaboration</b>	<b>Pool</b>	Extend a Partition element to logically organize an Activity.
	<b>Lane</b>	Extend a Partition element to subdivide a Pool.
	<b>Conversation</b>	Extend a Class element to group a set of Message Flows based on a certain concept.



Page	Item	Use to
	<b>Start Event</b>	Define the initiating event in a process.
	<b>Intermediate Event</b>	Define an intermediate event in a process.
	<b>End Event</b>	Define the terminating event in a process.
	<b>Gateway</b>	Define a decision point in a business process.  If a condition is <b>true</b> then processing continues one way; if <b>false</b> , then another.
	<b>Activity</b>	Represents work that is performed within a <i>Business Process</i> . An <i>Activity</i> can be modeled as a: <ul style="list-style-type: none"> <li>• <b>Sub-Process</b> - a compound <i>Activity</i> that is defined as a flow of other <b>BPMN 2.0</b> elements or</li> <li>• <b>Task</b> - an atomic <i>Activity</i> that cannot be broken down into a smaller unit</li> </ul> As a sub-process, the Activity can be made a composite element that links to a child diagram containing the flow of other BPMN elements.
	<b>Choreography</b>	Extend an Activity element to represent a process unit of information exchange between participants.  Enterprise Architect currently supports two to four participant bands within one Choreography Activity/Task.
	<b>Data Object</b>	Provide or store the information for an Activity.
	<b>Data Store</b>	Represent a mechanism for an Activity to retrieve or update stored information.
	<b>Group</b>	Extend a Boundary element to group other elements.
<b>BPMN 2.0 Collaboration Connectors</b>	<b>Text Annotation</b>	Create a comment.
	<b>Message Flow</b>	Extend a Control Flow relationship to define the flow of communications in the process.
	<b>Conversation Link</b>	Connect a Conversation Node to or from a Participant.

Page	Item	Use to
	<b>Sequence Flow</b>	Extend a Control Flow relationship to define the flow of activity.
	<b>Association</b>	Link the information and artifacts with BPMN graphic elements.

### Example BPMN 2.0 Collaboration Diagram



### Notes

- The appearance and specification of some elements and connectors are defined by Tagged Values

### Learn more

- [Composite Elements](#) <sup>[936]</sup>
- [Change BPMN Element Appearance](#) <sup>[1862]</sup>

### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Modeling Languages | Business | BPMN | Collaboration**
- (Alt+F1) | **Enterprise Architect | Business Modeling | BPMN | Collaboration Modeling**

#### 9.3.4.2.4 BPMN 2.0 Conversation Toolbox Pages

To create BPMN elements and relationships on **Conversation** diagrams in either BPMN 2.0 or BPEL formats, you can use the BPMN 2.0 **Conversation** pages of the Diagram Toolbox.

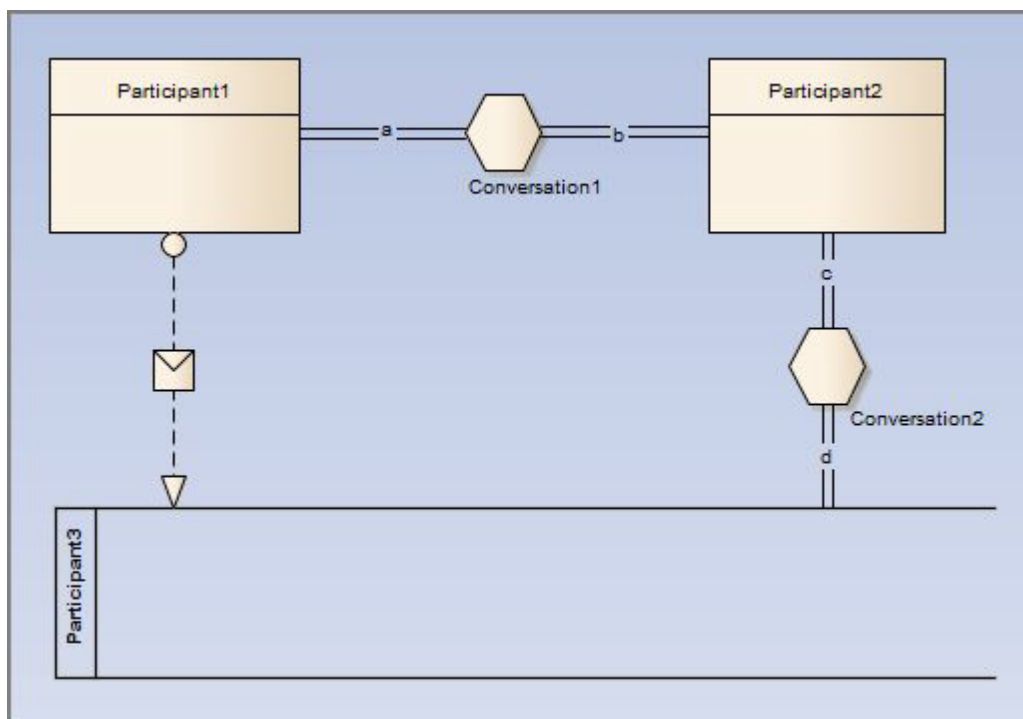
**Access** [Diagram](#) | [Diagram Toolbox: More tools](#) | [BPMN 2.0](#) | [BPMN 2.0 Conversation](#)

#### Element and Connector Descriptions

Page	Item	Use to
<b>BPMN 2.0 Conversation</b>	<b>Conversation Model</b>	Extend a composite Activity that defines the logical relationship of Message exchanges between two or more business entities.
	<b>Conversation</b>	Extend a Class element to group a set of Message Flows together based on a certain concept.
	<b>Participant</b>	Represent a specific organization or individual that is involved in a business process.
	<b>Pool</b>	Extend a Partition element to logically organize an Activity.
	<b>Activity</b>	Represents work that is performed within a <i>Business Process</i> . An <i>Activity</i> can be modeled as a: <ul style="list-style-type: none"> <li>• <b>Sub-Process</b> - a compound <i>Activity</i> that is defined as a flow of other <b>BPMN 2.0</b> elements or</li> <li>• <b>Task</b> - an atomic <i>Activity</i> that cannot be broken down into a smaller unit</li> </ul> <p>As a sub-process, the Activity can be made a composite element that links to a child diagram containing the flow of other BPMN elements.</p>
	<b>Start Event</b>	Define the initiating event in a process.
	<b>Intermediate Event</b>	Define an intermediate event in a process.
	<b>End Event</b>	Define the terminating event in a process.
	<b>Gateway</b>	Define a decision point in a business process.  If a condition is <b>true</b> then processing continues one way; if <b>false</b> , then another.

Page	Item	Use to
	<b>Text Annotation</b>	Create a comment.
<b>BPMN 2.0 Conversation Connectors</b>	<b>Conversation Link</b>	Connect a Conversation Node to a Participant, in either direction. <b>Either</b> the source <b>or</b> target element for a Conversation Link <b>must</b> be a Conversation Node element.
	<b>Message Flow</b>	Extend a Control Flow relationship to define the flow of communications in the process.
	<b>Sequence Flow</b>	Extend a Control Flow relationship to define the flow of activity.
	<b>Association</b>	Link the information and artifacts with BPMN graphic elements.

#### Example BPMN 2.0 Conversation Diagram



#### Notes

- The appearance and specification of some elements and connectors are defined by Tagged Values

Learn more

- [Composite Elements](#)<sup>[936]</sup>
- [Change BPMN Element Appearance](#)<sup>[1862]</sup>

Learning Center topics

- (Alt+F1) | **Enterprise Architect | Modeling Languages | Business | BPMN | Conversation**
- (Alt+F1) | **Enterprise Architect | Business Modeling | BPMN | Conversation Modeling**

**9.3.4.2.5 BPMN 2.0 Type Toolbox Page**

When you are working on a BPMN 2.0 diagram, you can also create a range of elements and connectors that are common to all of the BPMN 2.0 diagram types, by dragging them from the BPMN 2.0 **Types** page of the Diagram Toolbox.

**Access**    **Diagram | Diagram Toolbox: More tools | BPMN 2.0 | BPMN 2.0 <diagram type>**

Element and Connector Descriptions

Page	Item	Use To
<b>BPMN 2.0 Types</b>	<ul style="list-style-type: none"> <li>• Assignment</li> <li>• CorrelationKey</li> <li>• CorrelationProperty</li> <li>• CorrelationPropertyBinding</li> <li>• CorrelationPropertyRetrieval</li> <li>• CorrelationSubscription</li> <li>• Category</li> <li>• CategoryValue</li> <li>• ComplexBehaviorDefinition</li> <li>• ItemDefinition</li> <li>• Error</li> <li>• Escalation</li> <li>• Signal</li> <li>• IOSpecification</li> <li>• InputSet</li> <li>• OutputSet</li> <li>• InputOutputBinding</li> <li>• Interface</li> <li>• Endpoint</li> </ul>	<p>Define the non-graphic elements or properties (Tagged Values) of the Core BPMN 2.0 graphic elements.</p> <p>See <a href="#">Change BPMN Element Appearance</a><sup>[1862]</sup>.</p>


Page	Item	Use To
	<ul style="list-style-type: none"> <li>• Operation</li> <li>• Participant</li> <li>• Resource</li> <li>• ResourceRole</li> <li>• ResourceParameter</li> <li>• Property</li> <li>• ParticipantAssociation</li> <li>• ConversationAssociation</li> </ul>	

### 9.3.4.3 Change BPMN Element Appearance

A number of Tagged Values directly affect the appearance of the BPMN element or connector they apply to. Therefore, you can change the appearance of the object by changing the value of the tag.

Access **View | Tagged Values**

#### Define the appearance of BPMN elements and connectors

Step	Action	See also
1	Open the Tagged Values window.	<a href="#">Tagged Values</a> 
2	Select the required element or relationship in a diagram.  The Tagged Values window shows the appropriate Tagged Values and provides a list of values to assign to each one.	
3	Set the required value, based on the value descriptions in the table below.	

#### Values For Tags

Element	Tags	Values Setting
<b>Activity</b>	<i>activityType</i>	Set to: <ul style="list-style-type: none"> <li>• <b>Task</b> for no decoration</li> <li>• <b>Sub-Process</b> to display the <i>plus-in-box</i> decoration on the bottom edge of the shape</li> </ul>

Element	Tags	Values Setting
	<i>adHoc</i>	Set to <b>true</b> to display the <i>tilde</i> decoration on the bottom edge of the shape, indicating that this is an ad-hoc Activity.
	<i>isACallActivity</i>	Set to <b>true</b> to show the boundary of the Activity element as a thick line, indicating that this is a Call Activity.
	<i>isATransaction</i>	Set to <b>true</b> to give the Activity element a double-lined border, indicating that this is a Transaction.
	<i>isForCompensation</i>	Set to <b>true</b> to display the <i>rewind</i> icon on the bottom edge of the shape, indicating that this is a Compensation Activity.
	<i>isSequential</i>	Set to <b>true</b> to display three horizontal bars on the bottom edge of the shape.  The <i>loopCharacteristics</i> tag must first be set to <b>MultilInstance</b> .
	<i>loopCharacteristics</i>	Set to: <ul style="list-style-type: none"> <li>• <b>Standard</b> to display three vertical bars at the bottom edge of the shape</li> <li>• <b>MultilInstance</b> to display a loop icon at the bottom edge of the shape</li> </ul>
	<i>taskType</i>	When set to different value, the associated decoration is displayed at the top-left corner of the shape.
	<i>triggeredByEvent</i>	Set to <b>true</b> to show the boundary of the Activity element as a dotted line.
<b>Choreography</b>	<i>choreographyType</i>	Set to: <ul style="list-style-type: none"> <li>• <b>Task</b> for no decoration</li> <li>• <b>Sub-Process</b> to display the <i>plus-in-box</i> decoration on the bottom edge of the shape.</li> </ul>
	<i>initiatingParticipant</i>	Set a value to show the corresponding participant band in a light color, which indicates that this is the initiating participant.
	<i>isACallChoreography</i>	Set to <b>true</b> show the boundary of the Choreography element as a thick line.
	<i>loopType</i>	Set to: <ul style="list-style-type: none"> <li>• <b>Standard</b> to display three vertical bars at the bottom edge of the</li> </ul>

Element	Tags	Values Setting
		<p>shape</p> <ul style="list-style-type: none"> <li>• <b>MultilInstance</b> to display a loop icon at the bottom edge of the shape</li> </ul> <p>Each Participant has its own MultilInstance setting.</p>
	<i>numberOfParticipants</i>	Set to the appropriate numerical value to indicate the number of Participant bands for a choreography Activity or Task.
	<i>participantA</i> or <i>participantARef</i>	<p><i>participantA</i> - Type the name of the Participant to show on the Participant band of the Choreography.</p> <p>If the Participant is known and referenced, use the <i>participantARef</i> tag and leave the <i>participantA</i> tag blank; the name of the referenced Participant is shown on the Participant band of the Choreography.</p> <p>The same usage is applied to other Participant bands.</p>
<b>Conversation</b>	<i>isACallConversation</i>	Set to <b>true</b> to show the boundary of the Conversation element as a thick line.
	<i>isComposite</i>	Set to <b>true</b> to display a <i>plus-in-box</i> decoration on the bottom edge of the shape, indicating that the Conversation is a composite element.
<b>Data Object</b>	<i>dataInOut</i>	<p>Set to:</p> <ul style="list-style-type: none"> <li>• <b>Input</b> to add a light-colored arrow at the top-left corner of the Data Object, indicating that it represents a Data Input element</li> <li>• <b>Output</b> to add a dark-colored arrow at the top-left corner of the Data Object, indicating that it represents a Data Output element</li> </ul>
	<i>isCollection</i>	Set to <b>true</b> to add three vertical lines at the bottom edge of the shape, indicating that the object is a collection.
<b>Event</b>	<i>cancelActivity</i>	<p>For Intermediate Event only.</p> <p>Set to <b>false</b> to show the Intermediate Event border as a dashed line.</p>
	<i>catchOrThrow</i>	<p>For Intermediate Event only.</p> <p>Set to:</p> <ul style="list-style-type: none"> <li>• <b>Catch</b> or</li> <li>• <b>Throw</b></li> </ul>
	<i>eventDefinition</i>	Use to change the decoration of a Start Event, Intermediate Event or



Element	Tags	Values Setting
		End Event.
	<i>isInterrupting</i>	For Start Event only.  Set to <b>false</b> to show the border of the Start Event as a dashed line, indicating that the Start Event is not an interrupting event.
<b>Gateway</b>	<i>eventGatewayType</i>	Use to define two types of Event-based Gateway: <ul style="list-style-type: none"> <li>• <b>Exclusive</b> and</li> <li>• <b>Parallel</b></li> </ul>
	<i>gatewayType</i>	Set to: <ul style="list-style-type: none"> <li>• <b>Exclusive</b> to render the Gateway as a diamond shape without any marker in the middle; if you also set the <i>markerVisible</i> tag to <b>true</b>, the Gateway is rendered as a diamond shape with an X inside</li> <li>• <b>Complex</b> to render the Gateway as a diamond shape with an asterisk (*) inside</li> <li>• <b>Inclusive</b> to render the Gateway as a diamond shape with a circle (o) inside</li> <li>• <b>Parallel</b> to render the Gateway as a diamond shape with a plus (+) inside</li> <li>• <b>Event</b> to define the Gateway as Event-based.</li> </ul>
	<i>instantiate</i>	Use only to define the Exclusive and Parallel Event-based Gateway.
<b>Message</b>	<i>IsInitiate</i>	Set to <b>true</b> to make the envelope color gray, indicating that this is an initiating Message.
<b>Message Flow</b>	<i>messageVisible</i>	Set to: <ul style="list-style-type: none"> <li>• <b>Unspecified</b> to make the Message Flow connector without decoration</li> <li>• <b>Initiating</b> to add a white envelope to the Message Flow connector</li> <li>• <b>Non-Initiating</b> to add a gray envelope to the Message Flow connector</li> </ul>
<b>Participant</b>	<i>participantMultiplicity</i>	Set to <b>true</b> to display three vertical bars at the bottom edge of a rectangle Pool element.  Participant is rendered as a normal UML Class shape exclusively on Conversation diagrams.

Element	Tags	Values Setting
<b>Pool</b>	<i>blackBoxPool</i>	Set to <b>true</b> to render the Pool element as a rectangle, which is used in a Main Pool diagram.
	<i>participantMultiplicity</i>	Set to <b>true</b> to display three vertical bars at the bottom edge of a rectangle Pool element.
<b>Sequence Flow</b>	<i>conditionType</i>	Set to: <ul style="list-style-type: none"> <li>• <b>Expression</b> to show an unfilled diamond marker at the source end of the connector</li> <li>• <b>Default</b> to show a slash marker at the source end of the connector</li> </ul>

#### Notes

- Some BPMN elements changed in appearance between BPMN version 1.0 and BPMN version 1.1
- In Enterprise Architect releases later than 7.1, if you work on a BPMN 1.0 model created in an earlier release, existing elements default to their version 1.0 appearance; new elements assume the BPMN version 1.1 appearance and automatically have a Tagged Value **BPMNVersion** set to 1.1
- If you want a new element to revert to the BPMN version 1.0 appearance, set the **BPMNVersion** Tagged Value to 1.0
- Conversely, if you want an older element to assume the BPMN version 1.1 appearance, assign the **BPMNVersion** Tagged Value to it, with the value 1.1

#### 9.3.4.4 Migrate BPMN 1.0 Model to BPMN 1.1

It is possible to migrate a BPMN 1.0 model (or part of a model) to BPMN 1.1 or BPMN1.1::BPEL, using the Automation Interface. There are two functions you can use:

- *MigrateToBPMN11()*
  - `proj.MigrateToBPMN11 sGUID, "BPMN"`
  - `proj.MigrateToBPMN11 sGUID, "BPEL"`
- *Migrate()*
  - `proj.Migrate sGUID, "BPMN", "BPMN1.1"`
  - `proj.Migrate sGUID, "BPMN", "BPMN1.1::BPEL"`

These functions update the Tagged Values and, if required, stereotypes to BPMN 1.1 for all elements, attributes, connectors and diagrams under the selected package or element.

#### Example

The following VB script calls the *MigrateToBPMN11()* function to migrate the Tagged Values to BPMN 1.1:

```
Sub MigrateElement (sGUID, lngPackageID)

    Dim proj as EA.Project
    set proj = Repository.GetProjectInterface
    proj.MigrateToBPMN11 sGUID, "BPMN"
```

```

' refresh the model
If lngPackageID <> 0 Then
    Repository.RefreshModelView (lngPackageID)
End If

End Sub

Sub MigrateSelectedItem

    Dim selType
    Dim selElement as EA.Element
    Dim selPackage as EA.Package

    selType = GetTreeSelectedItemType

    If selType = 4 Then ' means Element
        set selElement = GetTreeSelectedObject
        MigrateElement selElement.ElementGUID, selElement.PackageID
        MsgBox "Complete", 0, "BPMN 1.1 Migration"

    ElseIf selType = 5 Then ' means Package
        set selPackage = GetTreeSelectedObject
        MigrateElement selPackage.PackageGUID, selPackage.PackageID
        MsgBox "Complete", 0, "BPMN 1.1 Migration"

    Else
        MsgBox "Select a Package or Element in the Project Browser to
        initiate migration", 0, "BPMN 1.1 Migration"

    End If

End Sub

Sub Main

    MigrateSelectedItem

End Sub

Main

```



### Notes

- In BPMN 1.0, various tags have free-text direct-entry value fields, and you can provide additional information on these tags in the Tagged Value Note dialog, for display at the bottom of the Tagged Values window

In BPMN 1.1, some of these tags (such as the *Categories* tag on a *BusinessProcess* stereotyped element) have been changed to **Memo** type, and you use the Tagged Value Note dialog to enter the value; therefore, you cannot have additional notes for these tags, all information must be within the tag's value

- For such tags, when migrating from BPMN 1.0 to BPMN 1.1, the BPMN 1.0 tag value is moved into the BPMN 1.1 tag **Notes** field and the BPMN 1.0 tag notes are discarded; if you want to preserve the tag notes text, take a copy of the BPMN 1.0 model before migration to enable you to copy the tag notes text into the tag value after migration

### Learn more

- [MigrateToBPMN11\(\)](#)  <sup>[2978]</sup>
- [Migrate\(\)](#)  <sup>[2978]</sup>

### 9.3.4.5 Migrate BPMN 1.1 Model to BPMN 2.0

It is possible to migrate a BPMN 1.1 model (or part of a model) to BPMN 2.0, using the Automation Interface function *Migrate()*. This function updates the Tagged Values and, if required, stereotypes to BPMN 2.0 for all elements, attributes, connectors and diagrams under the selected package or element.

#### Example

The following VB script calls the *Migrate()* function to migrate the package or element to BPMN 2.0:

```
Sub MigrateElement (sGUID, lngPackageID)
    Dim proj as EA.Project
    set proj = Repository.GetProjectInterface
    proj.Migrate sGUID, "BPMN1.1", "BPMN2.0"

    'refresh the model
    If lngPackageID <> 0 Then
        Repository.RefreshModelView (lngPackageID)
    End If
End Sub

Sub MigrateSelectedItem
    Dim selType
    Dim selElement as EA.Element
    Dim selPackage as EA.Package
    selType = GetTreeSelectedItemType
    If selType = 4 Then 'means Element
        set selElement = GetTreeSelectedObject
        MigrateElement selElement.ElementGUID, selElement.PackageID
        MsgBox "Element Migration Completed", 0, "BPMN 2.0 Migration"
    ElseIf selType = 5 Then 'means Package
        set selPackage = GetTreeSelectedObject
        MigrateElement selPackage.PackageGUID, selPackage.PackageID
        MsgBox "Package Migration Completed", 0, "BPMN 2.0 Migration"
    Else
        MsgBox "Select a Package or Element in the Project Browser to
        initiate migration", 0, "BPMN 2.0 Migration"
    End If
End Sub

Sub Main
    MigrateSelectedItem
End Sub

Main
```

#### Notes

- Please backup your project before you run the BPMN 2.0 Migrator
- Normalization occurs on the following tags:
  - BPMN1.1::Activity::InMessageRef
  - BPMN1.1::Activity::OutMessageRef
  - BPMN1.1::Activity::IORules
  - BPMN1.1::Activity::InputSets
  - BPMN1.1::Activity::OutputSets
  - BPMN1.1::Activity::ComplexMI\_FlowCondition
  - BPMN1.1::Activity::Performers
  - BPMN1.1::BusinessProcess::InputSets

- BPMN1.1::BusinessProcess::OutputSets
- BPMN1.1::BusinessProcess::Performers
- BPMN1.1::EndEvent::ErrorCode
- BPMN1.1::IntermediateEvent::ErrorCode

Taking *BPMN1.1::Activity::InMessageRef* as an example, the following steps take place:

- (1) Create a new element *BPMN2.0::Operation*
- (2) Insert the reference tag *BPMN2.0::Activity::operationRef*
- (3) Migrate *BPMN1.1::Activity::InMessageRef* to *BPMN2.0::Operation::InMessageRef*

- Denormalization occurs on the following tags:
  - BPMN1.1::Transaction::TransactionProtocol
  - BPMN1.1::WebService::Operation
  - BPMN1.1::WebService::Interface
  - BPMN1.1::WebService::ParticipantRef
  - BPMN1.1::Condition::ConditionExpression
  - BPMN1.1::BPELProcess::InputSets
  - BPMN1.1::BPELProcess::OutputSets

Taking *BPMN1.1::Transaction::TransactionMethod* as an example, the following steps take place:

- (1) Find element *BPMN1.1::Transaction* by '*BPMN1.1::Transaction::TransactionProtocol*
- (2) Migrate *BPMN1.1::Transaction::TransactionMethod* to *BPMN2.0::transactionMethod*
- (3) Migrate *BPMN1.1::Transaction::TransactionProtocol* to *BPMN2.0::transactionProtocol*

- If a BPMN1.1 stereo tag is discarded in BPMN2.0, it is preserved if the tag has a value

#### Learn more

- [Migrate\(\)](#)<sup>[2978]</sup>

### 9.3.4.6 BPMN 2.0 XML

It is possible to serialize **BPMN 2.0 Models** in **BPMN 2.0 XML**. The serialized XML file contains both the model semantics as well as the diagram-interchange information.

**Access** Click on a package in the Project Browser, then:

**Project | Publish Model**  
**Project | Model Import/Export | Export Package to XMI File: Publish, or**  
**Right-click | Import/Export | Export Package to XMI: Publish**

#### Serialize a Model


Step	Action	See also
1	Open the Publish Model Package dialog.	<a href="#">Publish Model Package</a> <sup>[476]</sup>

2	In the <b>XML Type</b> field, select <b>BPMN 2.0 XML</b> .	
3	Click on the <b>Export</b> button to initiate the BPMN 2.0 XML serialization.	

### Notes

- The package being serialized is assumed to be self-contained
- Only the contents of the selected package are considered for serialization - child packages (and their contents) are ignored
- Serialization of the **Group** element in the BPMN 2.0 **Business Process** Toolbox is not supported
- Serialization of the following items in the BPMN 2.0 **Types** section of the Toolbox is not supported:
  - *InputOutputBinding*
  - *Participant Association*
  - *Conversation Association*

### Learn more

- [XML Schema for BPMN 2.0 XML](#) (Online Resource)
- [BPMN 2.0 Toolbox Pages](#) 

## 9.3.5 BPEL Models

The following text is derived from the **BPEL** entry in the online Wikipedia:

**Business Process Execution Language (BPEL)**, short for **Web Services Business Process Execution Language (WS-BPEL)**, is an executable language for specifying interactions with **Web Services**. Processes in Business Process Execution Language export and import information by using Web Service interfaces exclusively.

Web service interactions can be described in two ways :



1. Executable business processes, which model the actual behavior of a participant in a business interaction.
2. Abstract business processes, which are partially specified processes that are not intended to be executed. An Abstract Process may hide some of the required concrete operational details.

**BPEL** is an **Orchestration** language, serialized in XML, which specifies an executable process that involves message exchanges with other systems. This messaging facility depends on the use of the **Web Services Description Language (WSDL) 1.1** to describe outgoing and incoming messages.

### BPEL in Enterprise Architect:

Enterprise Architect supports generating **BPEL** from executable processes in the **Business and Software Engineering** and the **Ultimate** editions. Although there is no standard graphical notation for **WS-BPEL**, Enterprise Architect uses **BPMN** profile as a graphical front-end to capture **BPEL** Process descriptions. With the help of the **BPMN** Profile, Enterprise Architect enables you to develop **BPEL** diagrams quickly and simply. While **BPMN** provides a graphical notation for visualizing business process, **BPEL** provides a way for visualizing this graphical business process in XML.




Enterprise Architect supports generating:

- BPEL 1.1  from models created using constructs in BPMN 1.1 Profile
- BPEL 2.0  from models created using constructs in BPMN 2.0 Profile

### Notes

- BPEL is supported in the **Business and Software Engineering** and the **Ultimate** editions of Enterprise Architect
- Enable BPMN 1.1 Technology for BPEL 1.1 and BPMN 2.0 Technology for BPEL 2.0 modeling in the MDG Technologies dialog ( **Settings | MDG Technologies** )

### Learn more

- [BPEL](#) (Online Resource)
- [Orchestration](#) (Online Resource)
- [BPEL 1.1 Model](#) 
- [BPEL 2.0 Model](#) 
- [BPMN Model](#) 


### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Modeling Languages | Business | BPMN | BPEL**
- (Alt+F1) | **Enterprise Architect | Business Modeling | BPEL | Getting Started**

#### 9.3.5.1 BPEL 1.1 Model

Enterprise Architect uses **BPMN 1.1** as a graphical front-end to capture **BPEL 1.1** Process description ( as BPEL 1.1 does not have not a standard graphical notation ). Enterprise Architect uses the partial mapping specified in the *OMG BPMN 1.1 Specification* to map BPMN 1.1 constructs to BPEL 1.1.

### How to

Step	Action	See also
1	Create a <b>BPEL 1.1 Package Structure</b> .	<a href="#">BPEL 1.1 Package Structure</a>   <

	<div> <div>d</div> <div>Click on the <b>BPEL Process</b> icon from the BPMN 1.1 Core toolbox and add it to the created BPEL diagram.</div> </div>	
2	Open the BPEL diagram under the <i>BPELProcess</i> stereotyped element.	
3	<p>Model the <i>BPEL Process</i> using the constructs in the BPMN 1.1 Core toolbox; use the following constructs in this toolbox to model BPEL 1.1:</p> <p>BPMN 1.1 - Core page:</p> <ul style="list-style-type: none"> <li>• BPEL Process</li> <li>• Activity</li> <li>• Start Event</li> <li>• Intermediate Event</li> <li>• End Event</li> <li>• Gateway</li> <li>• Pool</li> </ul> <p>BPMN 1.1 - Relationships page:</p> <ul style="list-style-type: none"> <li>• Sequence Flow</li> <li>• Message Flow</li> <li>• Association</li> </ul> <p>BPMN 1.1 Types page:</p> <ul style="list-style-type: none"> <li>• Assignment</li> <li>• Condition</li> <li>• Message</li> <li>• Participant</li> <li>• Web Service</li> </ul>	<a href="#">BPMN 1.1 Core Toolbox</a> <sup>[1848]</sup> <a href="#">Model BPEL 1.1 Process</a> <sup>[1876]</sup>
4	Create <b>Web Service</b> for the <i>BPEL Process</i> and other related <i>Pools</i> , if required.	<a href="#">BPEL Web Service</a> <sup>[1890]</sup>
5	Generate <b>BPEL 1.1</b> code from the <i>BPEL Process</i> .	<a href="#">Generate BPEL 1.1</a> <sup>[1893]</sup>

### Notes

- Generating BPEL 1.1 from a BPMN 1.1 model is supported in the Business and Software Engineering and the Ultimate editions of Enterprise Architect
- Ensure that BPMN 1.1 Technology is enabled in the MDG Technologies dialog ( **Settings | MDG Technologies** )



### BPEL 1.1 Example

- The Enterprise Architect Example file ( *EAExample.EAP* ) has a sample BPMN 1.1 model from which BPEL 1.1 can be generated
- If you have installed Enterprise Architect at the default location, open this file: *C:\Program Files\Sparx Systems\EA\EAExample.EAP*
- The BPMN 1.1 model package is in: **Project Models | Analysis and Business Modeling | BPEL Example | BPEL 1.1 Model**

### Modeling Restrictions

- Every *BPEL Process* and *Sub-Process* should start with a *Start Event* and end with an *End Event*
  - A *Start Event* or an *End Event* should not be attached to the boundary of a *Sub-Process*
- *Sequence Flow* Looping is not supported - only *Activity* looping is supported; all *Sequence Flows* should flow downstream and not upstream
- Mapping of an *Intermediate Event* with multiple triggers to BPEL 1.1 is not supported
- Mapping of *Multi-Instance Parallel While* loops to BPEL 1.1 is not supported
- Mapping of *Independent Sub-Process* to **BPEL 1.1** is not supported
- *Pools* are treated as *black box* (i.e. they do not expose any details) and hence they cannot contain any child elements or have any incoming/outgoing *Sequence Flow* connectors

#### 9.3.5.1.1 Create BPEL 1.1 Model Structure

A BPEL 1.1 model consists of a *BPEL Process* ( containing a BPEL diagram and mappable BPMN 1.1 constructs ) and other supporting elements ( like BPMN 1.1 *Assignment*, BPMN 1.1 *Web Service* ) required for generating BPEL 1.1 code. A sample BPEL 1.1 package structure can be created in the Project Browser, using the Select Model(s) (Model Wizard) dialog. You can use this package structure as a template for developing your *BPEL Process*.

### Access

- **Project Browser package context menu | Add a Model using Wizard** ( or )
- **Project Browser toolbar: New Model from Pattern**

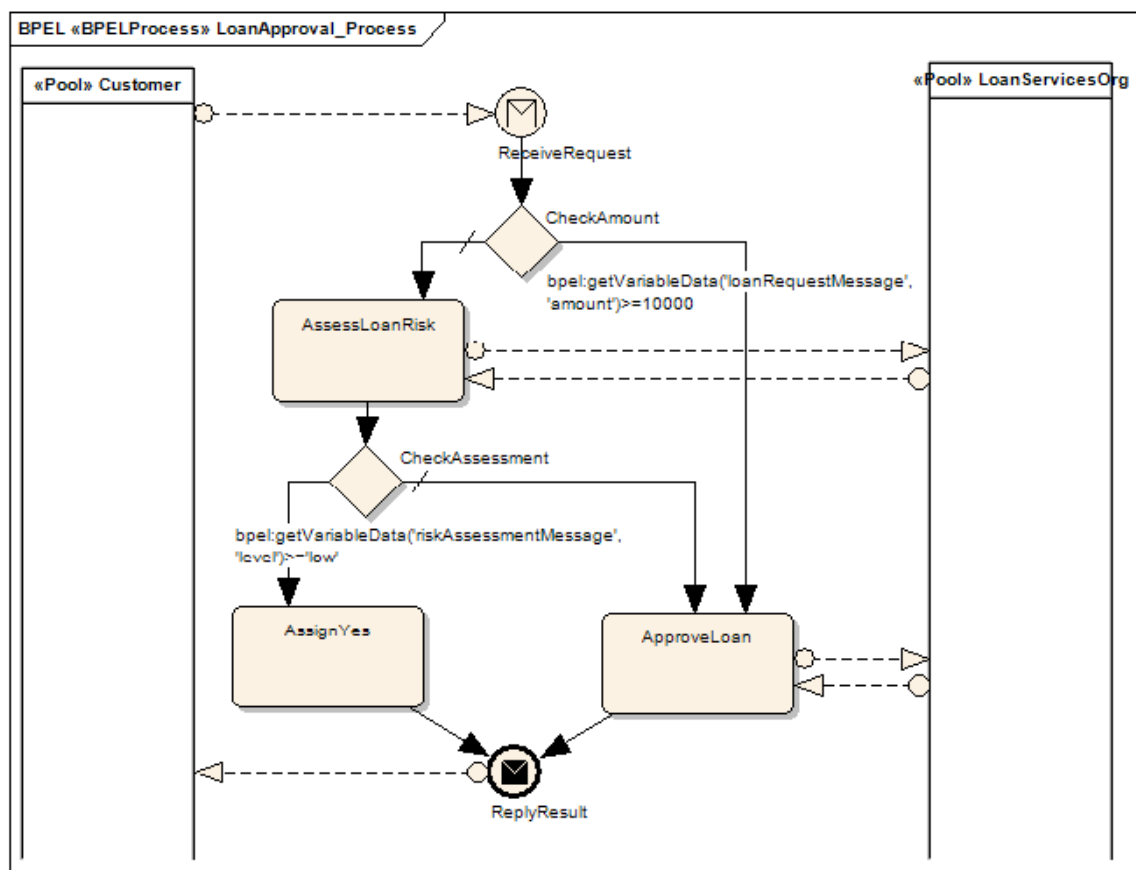
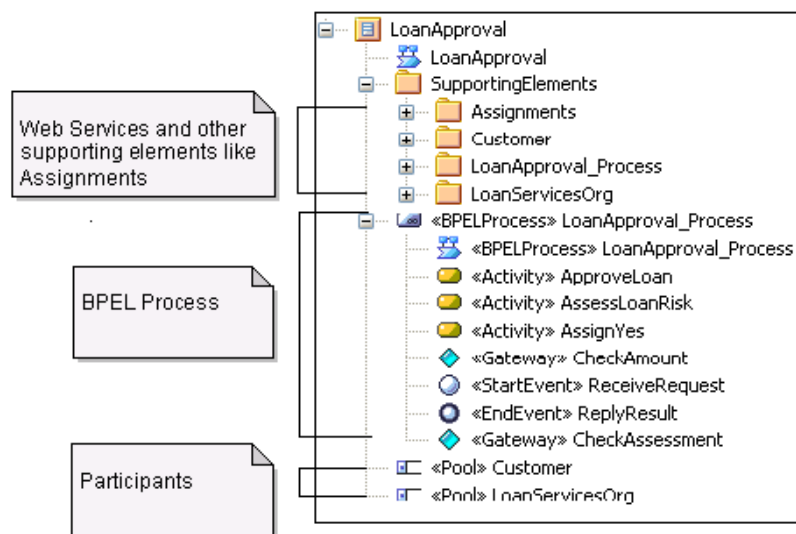
### Create a new BPEL 1.1 Package structure as a starting point for developing a BPEL Process

Step	Action	See also
1	Select the root node or a package in the Project Browser.	
2	Click on the <b>New Model from Pattern</b> icon in the toolbar.	<a href="#">Project Browser Toolbar</a> <sup>[66]</sup>
3	Select the value <b>BPMN 1.1</b> in the Technology section.	<a href="#">Model Wizard</a> <sup>[75]</sup>
4	Check the option <b>BPEL 1.1 Model</b> in the Name section.	

5	Click on the <b>OK</b> button to create the sample BPEL 1.1 package structure.	

**Example BPEL 1.1 Package Structure:**

The BPEL Process *LoanApproval\_Process* acts as container for the BPEL diagram and elements. The *SupportingElements* package contains supporting components like **Assignments** and **Web Services**.



### Learn more

- [Model a BPEL 1.1 Process](#) <sup>1876</sup>
- [Create a BPEL 1.1 Web Service](#) <sup>1890</sup>
- [Generate BPEL 1.1](#) <sup>1893</sup>

### 9.3.5.1.2 Model a BPEL 1.1 Process

The *BPEL Process* in Enterprise Architect represents the top-level container for the BPMN 1.1 elements, from which BPEL 1.1 can be generated. Conceptually it maps to the BPEL *process* element.

#### BPEL Properties:

- Double-click on the *BPEL Process* in the BPEL diagram
- Right-click on the *BPEL Process* | **BPEL** | **BPEL Properties**

#### Reference

Field/Button	Usage	See also
<b>Name</b>	Specify the name for the <i>BPEL Process</i> .	
<b>Query Language</b>	Specify the value of the language used in the <i>BPEL Process</i> for the selection of nodes in <i>Assignments</i> . Defaults to <b>XPath 1.0</b> .	
<b>OK</b>	Save the values entered in the dialog.	
<b>Cancel</b>	Discard the values entered in the dialog.	
<b>Help</b>	Display this Help topic.	
<b>General</b>	Open the UML Properties dialog.	<a href="#">Properties</a> [956]

#### Learn more

The *BPEL Process* element acts as a container for the following BPMN 1.1 constructs:

- [Start Event](#) [1877]
- [Intermediate Event](#) [1878]
- [Activity](#) [1881]
- [Gateway](#) [1884]
- [End Event](#) [1885]
- [Sequence Flow](#) [1887]

The following BPMN 1.1 constructs can be optionally used in BPEL modeling:

- [Pool](#) [1888]
- [Assignment](#) [1889]

### 9.3.5.1.2.1 Start Event

ABPMN 1.1 *Start Event* indicates where a particular *Process* begins. Every *Process* in Enterprise Architect must begin with a *Start Event*.

#### BPEL Properties:

- Double-click on the *Start Event* in the BPEL diagram ( or )
- Right-click on the *Start Event* | **BPEL** | **BPEL Properties**

#### Reference

Field/Button	Usage	See also												
Name	Specify the name for the <i>Start Event</i> .													
Trigger Type	Select a trigger for the <i>Start Event</i> , namely: <ul style="list-style-type: none"><li>• Conditional</li><li>• Link</li><li>• Message</li><li>• Multiple</li><li>• None</li><li>• Signal</li><li>• Timer</li></ul>													
Details	<div>Depending on the selected <b>Trigger Type</b>, the Details tab changes as follows:</div> <table><tr><th>Field</th><th>Usage</th></tr><tr><td colspan="2"><b>Conditional Trigger Type</b></td></tr><tr><td>Condition</td><td>Select a BPMN 1.1 <i>Condition</i> from the list of <i>Conditions</i> created in the <i>SupportingElements</i> package.</td></tr><tr><td colspan="2"><b>Message Trigger Type</b></td></tr><tr><td>Web Service</td><td>Select a BPMN 1.1 <i>Web Service</i> from the package (<i>SupportingElements</i> package) that represents involved <i>Participants</i>.</td></tr><tr><td>Message</td><td>Select a BPMN 1.1 <i>Message</i> from the list of <i>Messages</i> the selected Web Service.</td></tr></table>	Field	Usage	<b>Conditional Trigger Type</b>		Condition	Select a BPMN 1.1 <i>Condition</i> from the list of <i>Conditions</i> created in the <i>SupportingElements</i> package.	<b>Message Trigger Type</b>		Web Service	Select a BPMN 1.1 <i>Web Service</i> from the package ( <i>SupportingElements</i> package) that represents involved <i>Participants</i> .	Message	Select a BPMN 1.1 <i>Message</i> from the list of <i>Messages</i> the selected Web Service.	
Field	Usage													
<b>Conditional Trigger Type</b>														
Condition	Select a BPMN 1.1 <i>Condition</i> from the list of <i>Conditions</i> created in the <i>SupportingElements</i> package.													
<b>Message Trigger Type</b>														
Web Service	Select a BPMN 1.1 <i>Web Service</i> from the package ( <i>SupportingElements</i> package) that represents involved <i>Participants</i> .													
Message	Select a BPMN 1.1 <i>Message</i> from the list of <i>Messages</i> the selected Web Service.													

	<div></div> <div><b>Multiple Trigger Type</b></div> <table><tr><td><b>Events</b></td><td>Select additional BPMN 1.1 <i>Start Events</i> from the events in this <i>Process</i> that might trigger the <i>Process</i> clicking the ( ... ) button.</td></tr></table> <div></div> <div><b>Timer Trigger Type</b></div> <table><tr><td><b>Time Cycle</b></td><td>Specify the value for the time duration.</td></tr><tr><td><b>Time Date</b></td><td>Specify the value for the time date.</td></tr></table> <div></div>		<b>Events</b>	Select additional BPMN 1.1 <i>Start Events</i> from the events in this <i>Process</i> that might trigger the <i>Process</i> clicking the ( ... ) button.	<b>Time Cycle</b>	Specify the value for the time duration.	<b>Time Date</b>	Specify the value for the time date.	
<b>Events</b>	Select additional BPMN 1.1 <i>Start Events</i> from the events in this <i>Process</i> that might trigger the <i>Process</i> clicking the ( ... ) button.								
<b>Time Cycle</b>	Specify the value for the time duration.								
<b>Time Date</b>	Specify the value for the time date.								
<b>Assignments</b>	Select one or more <i>Assignment</i> elements created in the <i>SupportingElements</i> package in this tab ( Optional ).	<a href="#">Assignment</a> <sup>[1889]</sup> <a href="#">Create BPEL 1.1 Model Structure</a> <sup>[1873]</sup>							
<b>OK</b>	Save the values entered in the dialog.								
<b>Cancel</b>	Discard the values entered in the dialog.								
<b>Help</b>	Display this Help topic.								
<b>General</b>	Open the UML Properties dialog.	<a href="#">Properties</a> <sup>[956]</sup>							

**Notes**

- Either set **Time Cycle** or **Time Date** for **Timer Trigger Type** but not both as they are mutually exclusive fields
- **Link**, **None** and **Signal Event** types cannot be mapped to BPEL 1.1


**9.3.5.1.2.2 Intermediate Event**


A BPMN 1.1 *Intermediate Event* indicates where an event occurs somewhere between the start and end of a *Process*.

**BPEL Properties:**

- Double-click on the *Intermediate Event* in the BPEL diagram
- Right-click on the *Intermediate Event* | **BPEL** | **BPEL Properties**

Reference

Field/Button	Usage	See also																								
Name	Specify the name for the <i>Intermediate Event</i> .																									
Trigger Type	Select the type for the <i>Intermediate Event</i> , namely: <ul style="list-style-type: none"><li>• Cancel</li><li>• Compensation</li><li>• Conditional</li><li>• Error</li><li>• Link</li><li>• Message</li><li>• Multiple</li><li>• None</li><li>• Signal</li><li>• Timer</li></ul>																									
Details	<div>Depending on the selected <b>Trigger Type</b>, the Details tab changes as follows:</div> <table><tr><th>Field</th><th>Usage</th></tr><tr><td colspan="2"><b>Compensation Trigger Type</b></td></tr><tr><td>Activity</td><td>Select a BPMN 1.1 <i>Task</i> from the list of all <i>Activities</i> in the <i>Process</i>.</td></tr><tr><td colspan="2"></td></tr><tr><td colspan="2"><b>Conditional Trigger Type</b></td></tr><tr><td>Condition</td><td>Select a BPMN 1.1 <i>Condition</i> from the list of <i>Conditions</i> created in the <i>SupportingElements</i> package.</td></tr><tr><td colspan="2"></td></tr><tr><td colspan="2"><b>Error Trigger Type</b></td></tr><tr><td>Error Code</td><td>Specify the required error code.</td></tr><tr><td colspan="2"></td></tr><tr><td colspan="2"><b>Link Trigger Type</b></td></tr><tr><td>Link Event</td><td>Select a <i>Link Event</i> element ( to act as a target).</td></tr></table>	Field	Usage	<b>Compensation Trigger Type</b>		Activity	Select a BPMN 1.1 <i>Task</i> from the list of all <i>Activities</i> in the <i>Process</i> .			<b>Conditional Trigger Type</b>		Condition	Select a BPMN 1.1 <i>Condition</i> from the list of <i>Conditions</i> created in the <i>SupportingElements</i> package.			<b>Error Trigger Type</b>		Error Code	Specify the required error code.			<b>Link Trigger Type</b>		Link Event	Select a <i>Link Event</i> element ( to act as a target).	<a href="#">Create BPEL 1.1 Model Structure</a> 
Field	Usage																									
<b>Compensation Trigger Type</b>																										
Activity	Select a BPMN 1.1 <i>Task</i> from the list of all <i>Activities</i> in the <i>Process</i> .																									
<b>Conditional Trigger Type</b>																										
Condition	Select a BPMN 1.1 <i>Condition</i> from the list of <i>Conditions</i> created in the <i>SupportingElements</i> package.																									
<b>Error Trigger Type</b>																										
Error Code	Specify the required error code.																									
<b>Link Trigger Type</b>																										
Link Event	Select a <i>Link Event</i> element ( to act as a target).																									

		<i>Link Event</i> ) from the list under the current <i>Process Sub-Process</i> .
	<b>Message Trigger Type</b>	
	<b>Web Service</b>	Select a BPMN 1.1 <i>Web Service</i> from the package (the <i>SupportingElements</i> package) that represents the involved <i>Participants</i> .
	<b>Message</b>	Select a BPMN 1.1 <i>Message</i> from the list of <i>Messages</i> in the selected <i>Web Service</i> .
	<b>Multiple Trigger Type</b>	
	<b>Events</b>	Select additional BPMN 1.1 <i>Intermediate Events</i> from the list of events in this <i>Process</i> by clicking the <i>Select</i> button.
	<b>Timer Trigger Type</b>	
	<b>Time Cycle</b>	Specify the value for the time duration.
	<b>Time Date</b>	Specify the value for the time date.
<b>OK</b>	Save the values entered in the dialog.	
<b>Cancel</b>	Discard the values entered in the dialog.	
<b>Help</b>	Display this Help topic.	
<b>General</b>	Open the UML Properties dialog.	<a href="#">Properties</a> 

**Notes**

- **Cancel**, **None** and **Signal Event** types cannot be mapped to BPEL 1.1
- Either set **Time Cycle** or **Time Date** for **Timer Trigger Type** but not both as they are mutually exclusive fields



- For a **Compensation Intermediate Event** edge-mounted on an *Activity*, create a BPMN 1.1 *Association* from this event to the *Compensation Activity*; ensure that the *IsCompensation* tag for the *Activity* is set to true
- **Link Intermediate Event** can be used either as a GOTO or an off-page connector; therefore, this event can have either incoming or outgoing *Sequence Flows* - but not both

#### 9.3.5.1.2.3 Activity

A BPMN 1.1 *Activity* represents work that is performed within a *Process*. An *Activity* can be modeled as a:

- **Sub-Process** - a compound *Activity* that is defined as a flow of other **BPMN 1.1** elements or
- **Task** - an atomic *Activity* that cannot be broken down into a smaller unit

Activities - both *Tasks* and *Sub-Processes* - can also, optionally, act as *Looping constructs*. The *OMG BPMN 1.1 Specification* defines two types of *Looping construct*:

- **Standard Loop** ( *while* or *until* )
- **Multi-Instance Loop** ( *for each* )

#### BPEL Properties:

- Double-click on the *Activity* in the BPEL diagram
- Right-click on the *Activity* | **BPEL** | **BPEL 1.1 Properties**

#### Reference

Field/Button	Usage	See also
<b>Name</b>	Specify the name for the <i>Activity</i> .	
<b>Type</b>	Specify whether the <i>Activity</i> is a : <ul style="list-style-type: none"> <li>• <b>Task</b> ( or )</li> <li>• <b>Sub-Process</b></li> </ul>	
<b>Task Type / SubProcess Type</b>	Depending on the value selected in the <b>Type</b> field, the <b>Task Type / SubProcess Type</b> field has the following values: <p><b>Task Type :</b></p> <ul style="list-style-type: none"> <li>• Manual</li> <li>• None</li> <li>• Receive</li> <li>• Reference</li> <li>• Script</li> <li>• Send</li> <li>• Service</li> <li>• User</li> </ul>	

	<div>SubProcess Type :</div> <div><div><div></div></div><div><div></div></div><div><div></div></div></div>																									
Details	<div>Depending on the selected Task Type, the Details tab changes as follows:</div> <table><tr><th>Field</th><th>Usage</th></tr><tr><td colspan="2">Receive Task Type</td></tr><tr><td>Web Service</td><td>Select a BPMN 1.1 Web Service from the pack (the SupportingElements package) that represents the involved Participants.</td></tr><tr><td>Message</td><td>Select a BPMN 1.1 Message from the list of Messages in the selected Web Service.</td></tr><tr><td>Instantiate</td><td>Select True if this is the first Activity after the Start, otherwise select False.</td></tr><tr><td colspan="2">Reference Task Type</td></tr><tr><td>Activity</td><td>Select a BPMN 1.1 Task from the list of all Activities in the Process.</td></tr><tr><td colspan="2">Send Task Type</td></tr><tr><td>Web Service</td><td>Select a BPMN 1.1 Web Service from the pack (the SupportingElements package) that represents the involved Participants.</td></tr><tr><td>Message</td><td>Select a BPMN 1.1 Message from the list of Messages in the selected Web Service.</td></tr><tr><td colspan="2">Service / User Task Type</td></tr><tr><td>Web Service</td><td>Select a BPMN 1.1 Web Service from the pack (the SupportingElements package) that represents the involved Participants.</td></tr></table>	Field	Usage	Receive Task Type		Web Service	Select a BPMN 1.1 Web Service from the pack (the SupportingElements package) that represents the involved Participants.	Message	Select a BPMN 1.1 Message from the list of Messages in the selected Web Service.	Instantiate	Select True if this is the first Activity after the Start, otherwise select False.	Reference Task Type		Activity	Select a BPMN 1.1 Task from the list of all Activities in the Process.	Send Task Type		Web Service	Select a BPMN 1.1 Web Service from the pack (the SupportingElements package) that represents the involved Participants.	Message	Select a BPMN 1.1 Message from the list of Messages in the selected Web Service.	Service / User Task Type		Web Service	Select a BPMN 1.1 Web Service from the pack (the SupportingElements package) that represents the involved Participants.	
Field	Usage																									
Receive Task Type																										
Web Service	Select a BPMN 1.1 Web Service from the pack (the SupportingElements package) that represents the involved Participants.																									
Message	Select a BPMN 1.1 Message from the list of Messages in the selected Web Service.																									
Instantiate	Select True if this is the first Activity after the Start, otherwise select False.																									
Reference Task Type																										
Activity	Select a BPMN 1.1 Task from the list of all Activities in the Process.																									
Send Task Type																										
Web Service	Select a BPMN 1.1 Web Service from the pack (the SupportingElements package) that represents the involved Participants.																									
Message	Select a BPMN 1.1 Message from the list of Messages in the selected Web Service.																									
Service / User Task Type																										
Web Service	Select a BPMN 1.1 Web Service from the pack (the SupportingElements package) that represents the involved Participants.																									

	<table><tr><td></td><td></td></tr><tr><td><b>Input Message</b></td><td>Select a BPMN 1.1 <i>Message</i> from the list of <i>M</i> in the selected Web Service that represents the element.</td></tr><tr><td><b>Output Message</b></td><td>Select a BPMN 1.1 <i>Message</i> from the list of <i>M</i> in the selected Web Service that represents the element.</td></tr></table> <p>Depending on the selected <b>SubProcess Type</b>, the Details tab changes as follows :</p> <table><tr><th>Field</th><th>Usage</th></tr><tr><td colspan="2"><b>References SubProcess Type</b></td></tr><tr><td><b>SubProcess</b></td><td>Select a BPMN 1.1 <i>Sub-Process</i> from the list of <i>Activities</i> in the <i>Process</i>.</td></tr></table>			<b>Input Message</b>	Select a BPMN 1.1 <i>Message</i> from the list of <i>M</i> in the selected Web Service that represents the element.	<b>Output Message</b>	Select a BPMN 1.1 <i>Message</i> from the list of <i>M</i> in the selected Web Service that represents the element.	Field	Usage	<b>References SubProcess Type</b>		<b>SubProcess</b>	Select a BPMN 1.1 <i>Sub-Process</i> from the list of <i>Activities</i> in the <i>Process</i> .	
<b>Input Message</b>	Select a BPMN 1.1 <i>Message</i> from the list of <i>M</i> in the selected Web Service that represents the element.													
<b>Output Message</b>	Select a BPMN 1.1 <i>Message</i> from the list of <i>M</i> in the selected Web Service that represents the element.													
Field	Usage													
<b>References SubProcess Type</b>														
<b>SubProcess</b>	Select a BPMN 1.1 <i>Sub-Process</i> from the list of <i>Activities</i> in the <i>Process</i> .													
<b>Assignments</b>	Select one or more <i>Assignment</i> elements created in the <i>SupportingElements</i> package in this tab (Optional).	<a href="#">Assignment</a> <sup>[1889]</sup> <a href="#">Create BPEL 1.1 Model Structure</a> <sup>[1873]</sup>												
<b>Loop Details</b>	<p>Activities can be repeated sequentially, behaving like a loop. Specify the <i>Activity</i> looping details in this tab (Optional).</p> <table><tr><th>Field</th><th>Usage</th></tr><tr><td><b>Type</b></td><td>Select:<ul style="list-style-type: none"><li>• <b>Standard</b> for creating a <i>while</i> or <i>until</i> loop.</li><li>• <b>MultInstance</b> for creating a <i>for each</i> loop.</li></ul></td></tr><tr><td><b>Condition</b></td><td>Enter a boolean condition for Standard loop or numeric condition for MultInstance loop.</td></tr><tr><td><b>Min Value</b></td><td>Specify the minimum value for the loop evaluation for a Standard loop( Optional ).</td></tr><tr><td><b>Max Value</b></td><td>Specify the maximum value for the loop evaluation for a Standard loop( Optional ).</td></tr><tr><td><b>Test Time</b></td><td>Select <b>After</b> to define a <i>while</i> Standard loop.</td></tr></table>	Field	Usage	<b>Type</b>	Select: <ul style="list-style-type: none"><li>• <b>Standard</b> for creating a <i>while</i> or <i>until</i> loop.</li><li>• <b>MultInstance</b> for creating a <i>for each</i> loop.</li></ul>	<b>Condition</b>	Enter a boolean condition for Standard loop or numeric condition for MultInstance loop.	<b>Min Value</b>	Specify the minimum value for the loop evaluation for a Standard loop( Optional ).	<b>Max Value</b>	Specify the maximum value for the loop evaluation for a Standard loop( Optional ).	<b>Test Time</b>	Select <b>After</b> to define a <i>while</i> Standard loop.	
Field	Usage													
<b>Type</b>	Select: <ul style="list-style-type: none"><li>• <b>Standard</b> for creating a <i>while</i> or <i>until</i> loop.</li><li>• <b>MultInstance</b> for creating a <i>for each</i> loop.</li></ul>													
<b>Condition</b>	Enter a boolean condition for Standard loop or numeric condition for MultInstance loop.													
<b>Min Value</b>	Specify the minimum value for the loop evaluation for a Standard loop( Optional ).													
<b>Max Value</b>	Specify the maximum value for the loop evaluation for a Standard loop( Optional ).													
<b>Test Time</b>	Select <b>After</b> to define a <i>while</i> Standard loop.													

	<div>Before to define an <i>until</i> Standard loop.</div>	
<b>OK</b>	Save the values entered in the dialog.	
<b>Cancel</b>	Discard the values entered in the dialog.	
<b>Help</b>	Display this Help topic.	
<b>General</b>	Open the UML Properties dialog.	<a href="#">Properties</a> <sup>956</sup>

#### Notes

- **Manual** and **Script** types cannot be mapped to BPEL 1.1
- **Reusable Sub-Process** types cannot be mapped to BPEL 1.1

#### 9.3.5.1.2.4 Gateway

A BPMN 1.1 *Gateway* control the way in which *Sequence Flows* converge and diverge within a *Process*. They provide a gating mechanism that either allows or blocks a *Sequence Flow*.

#### BPEL Properties:

- Double-click on the *Gateway* in the BPEL diagram
- Right-click on the *Gateway* | **BPEL** | **BPEL Properties**

#### Reference

Field/Button	Usage	See also
<b>Name</b>	Specify the name for the <i>Gateway</i> .	
<b>Gateway</b>	Select the type for the <i>Gateway</i> , namely: <ul style="list-style-type: none"> <li>• Complex</li> <li>• Exclusive</li> <li>• Inclusive</li> <li>• Parallel</li> </ul>	
<b>Details</b>	Depending on the selected <b>Gateway</b> , the Details tab changes as follows:	

	<table><tr><th>Field</th><th>Usage</th></tr><tr><td colspan="2"><b>Exclusive Gateway</b></td></tr><tr><td><b>Exclusive Type</b></td><td><b>Data</b> - allows alternate paths to be taken based on the evaluation of conditions.  <b>Event</b> - allows alternate paths to be taken based on the occurrence of events.</td></tr></table>	Field	Usage	<b>Exclusive Gateway</b>		<b>Exclusive Type</b>	<b>Data</b> - allows alternate paths to be taken based on the evaluation of conditions.  <b>Event</b> - allows alternate paths to be taken based on the occurrence of events.	
Field	Usage							
<b>Exclusive Gateway</b>								
<b>Exclusive Type</b>	<b>Data</b> - allows alternate paths to be taken based on the evaluation of conditions.  <b>Event</b> - allows alternate paths to be taken based on the occurrence of events.							
<b>OK</b>	Save the values entered in the dialog.							
<b>Cancel</b>	Discard the values entered in the dialog.							
<b>Help</b>	Display this Help topic.							
<b>General</b>	Open the UML Properties dialog.	<a href="#">Properties</a> 9561						

**Notes**

- The target of the outgoing *Sequence Flows* of this Event *Exclusive Gateway* must be either a:
  - Receive Task** or a
  - Message** or **Timer Intermediate Event**
- If a *Message Task* is one of the targets of the outgoing *Sequence Flow* of an Event *Exclusive Gateway*, then a *Message Intermediate Event* cannot be the target of the other outgoing *Sequence Flows* of this Event *Exclusive Gateway*

**9.3.5.1.2.5 End Event**

A BPMN 1.1 *End Event* indicates where a particular *Process* ends. Every *Process* in Enterprise Architect must end with an *End Event*.




**BPEL Properties:**

- Double-click on the *End Event* in the BPEL diagram
- Right-click on the *End Event* | **BPEL** | **BPEL Properties**

**Reference**

Field/Button	Usage	See also
<b>Name</b>	Specify the name for the <i>End Event</i> .	

<b>Result Type</b>	Select the type for the <i>End Event</i> , namely: <ul style="list-style-type: none"> <li>• Cancel</li> <li>• Compensation</li> <li>• Error</li> <li>• Link</li> <li>• Message</li> <li>• Multiple</li> <li>• None</li> <li>• Signal</li> <li>• Terminate</li> </ul>																					
<b>Details</b>	<p>Depending on the selected <b>Result Type</b>, the Details tab changes as follows:</p> <table border="1"> <thead> <tr> <th>Field</th><th>Usage</th></tr> </thead> <tbody> <tr> <td colspan="2"><b>Compensation Result Type</b></td></tr> <tr> <td><b>Activity</b></td><td>Select a BPMN 1.1 <i>Task</i> from the list of all <i>Activities</i> in the <i>Process</i>.</td></tr> <tr> <td colspan="2"><b>Error Trigger Type</b></td></tr> <tr> <td><b>Error Code</b></td><td>Specify the required error code.</td></tr> <tr> <td colspan="2"><b>Message Result Type</b></td></tr> <tr> <td><b>Web Service</b></td><td>Select a BPMN 1.1 <i>Web Service</i> from the package (the <i>SupportingElements</i> package) that represents the involved <i>Participants</i>.</td></tr> <tr> <td><b>Message</b></td><td>Select a BPMN 1.1 <i>Message</i> from the list of <i>Messages</i> in the selected <i>Web Service</i>.</td></tr> <tr> <td colspan="2"><b>Multiple Result Type</b></td></tr> <tr> <td><b>Events</b></td><td>Select additional BPMN 1.1 <i>Start Events</i> from the list of <i>Start Events</i> in this <i>Process</i> that might trigger the <i>Process</i> by clicking the ( ... ) button.</td></tr> </tbody> </table>	Field	Usage	<b>Compensation Result Type</b>		<b>Activity</b>	Select a BPMN 1.1 <i>Task</i> from the list of all <i>Activities</i> in the <i>Process</i> .	<b>Error Trigger Type</b>		<b>Error Code</b>	Specify the required error code.	<b>Message Result Type</b>		<b>Web Service</b>	Select a BPMN 1.1 <i>Web Service</i> from the package (the <i>SupportingElements</i> package) that represents the involved <i>Participants</i> .	<b>Message</b>	Select a BPMN 1.1 <i>Message</i> from the list of <i>Messages</i> in the selected <i>Web Service</i> .	<b>Multiple Result Type</b>		<b>Events</b>	Select additional BPMN 1.1 <i>Start Events</i> from the list of <i>Start Events</i> in this <i>Process</i> that might trigger the <i>Process</i> by clicking the ( ... ) button.	
Field	Usage																					
<b>Compensation Result Type</b>																						
<b>Activity</b>	Select a BPMN 1.1 <i>Task</i> from the list of all <i>Activities</i> in the <i>Process</i> .																					
<b>Error Trigger Type</b>																						
<b>Error Code</b>	Specify the required error code.																					
<b>Message Result Type</b>																						
<b>Web Service</b>	Select a BPMN 1.1 <i>Web Service</i> from the package (the <i>SupportingElements</i> package) that represents the involved <i>Participants</i> .																					
<b>Message</b>	Select a BPMN 1.1 <i>Message</i> from the list of <i>Messages</i> in the selected <i>Web Service</i> .																					
<b>Multiple Result Type</b>																						
<b>Events</b>	Select additional BPMN 1.1 <i>Start Events</i> from the list of <i>Start Events</i> in this <i>Process</i> that might trigger the <i>Process</i> by clicking the ( ... ) button.																					

<b>Assignments</b>	Select one or more <i>Assignment</i> elements created in the <i>SupportingElements</i> package in this tab (Optional).	<a href="#">Assignment</a>  <a href="#">Create BPEL 1.1 Model Structure</a> 
<b>OK</b>	Save the values entered in the dialog.	
<b>Cancel</b>	Discard the values entered in the dialog.	
<b>Help</b>	Display this Help topic.	
<b>General</b>	Open the UML Properties dialog.	<a href="#">Properties</a> 

Notes

- **Cancel**, **Link**, **None** and **Signal** *Event* types cannot be mapped to BPEL 1.1

**9.3.5.1.2.6 Sequence Flow**

A BPMN 1.1 *Sequence Flow* connector shows the order in which the activities (*Tasks* and *Events*) are performed in a *BPEL Process*.

BPEL Properties:

- Double-click on the *Sequence Flow* in the BPEL diagram
- Right-click on the *Sequence Flow* | **BPEL** | **BPELProperties**

Reference

Field/Button	Usage	See also
<b>Condition Type</b>	Specify the type of the condition on the <i>Sequence Flow</i> , namely : <ul style="list-style-type: none"> <li>• None</li> <li>• Default</li> <li>• Expression</li> </ul>	
<b>Expression</b>	This field gets enabled when the <b>Condition Type</b> is set to <b>Expression</b> . Specify a boolean expression to act as a gating condition.	
<b>Ordering</b>	This field gets enabled when the <b>Condition Type</b> is set to <b>Expression</b> .	

	Specify a numerical value that determines the order in which the condition set in the <b>Expression</b> field is to be evaluated.	
<b>Assignments</b>	Select one or more <i>Assignment</i> elements created in the <i>SupportingElements</i> package in this tab (Optional).	<a href="#">Assignment</a> <sup>1885</sup> <a href="#">Create BPEL 1.1 Model Structure</a> <sup>1873</sup>
<b>OK</b>	Save the values entered in the dialog.	
<b>Cancel</b>	Discard the values entered in the dialog.	
<b>Help</b>	Display this Help topic.	
<b>General</b>	Open the UML Properties dialog.	<a href="#">Properties</a> <sup>956</sup>

#### 9.3.5.1.2.7 Pool

A BPMN 1.1 *Pool* represents a Participant in a *Process* and does not map to any specific BPEL 1.1 element. Enterprise Architect uses Pools to represent external Participants, with which the BPEL Process communicates. These are 'black box' *pools*; that is, they are abstract and do not expose any details (they do not contain any BPMN 1.1 elements inside them).

#### BPEL Properties:

- Double-click on the *Pool* in the BPEL diagram
- Right-click on the *Pool* | **BPEL** | **BPEL Properties**

#### Reference

Field/Button	Usage	See also
<b>Name</b>	Specify the name for the <i>Pool</i> .	
<b>OK</b>	Save the values entered in the dialog.	
<b>Cancel</b>	Discard the values entered in the dialog.	
<b>Help</b>	Display this Help topic.	
<b>General</b>	Open the UML Properties dialog.	<a href="#">Properties</a> <sup>956</sup>



**Notes**

- A *BPEL Process* should not contain a *Pool* as its child element
- A BPEL diagram under a *BPEL Process* contains an implicit *Pool* - so it is invalid to add a *Pool* in this BPEL diagram to represent the *BPEL Process*
- *Pool* cannot have any incoming or outgoing *Sequence Flow* connectors - it can have only incoming or outgoing *Message Flow* connectors

**9.3.5.1.2.8 Assignment**

A BPMN 1.1 *Assignment* element enables data to be copied between messages and new data to be inserted, using expressions within a *BPEL Process*. A BPMN 1.1 *Assignment* element maps to a BPEL 1.1 *assign* activity and copies the specified value from the source to the target.

In Enterprise Architect, *Assignment* elements should be created in the *Assignments* package in *SupportingElements*<sup>[1897]</sup>. If they are created elsewhere, they cannot be enacted correctly.




**Access:** [Diagram](#) | [Diagram Toolbox: More tools](#) | [BPMN 1.1](#) | [BPMN 1.1 Types](#)

**BPEL Properties:**

- Double-click on the *Assignment* in the BPEL diagram
- Right-click on the *Assignment* | [BPEL](#) | [BPEL Properties](#)

**Reference**

Field/Button	Usage	See also
<b>Name</b>	Specify the name for the <i>Assignment</i> .	
<b>Assign Time</b>	Select either <b>Start</b> or <b>End</b> to determine whether the assignment occurs at the start or end of an <i>Activity/Event</i> .	
<b>Copy From</b> section		
<b>Type</b>	Depending on the value selected in this field, further details are required.	
<b>Literal</b>	Specify a literal value.	
<b>Expression / Message</b>	Select a BPMN 1.1 <i>Message</i> in the package representing the <i>BPEL Process / Pool</i> under the <i>SupportingElements</i> package.	<a href="#">Create BPEL 1.1 Model Structure</a> <sup>[1873]</sup> <a href="#">Create a BPEL 1.1 Web Service</a> <sup>[1890]</sup>

<b>Part</b>	Select a BPMN 1.1 <i>Property</i> belonging to the selected <b>Message</b> (Optional).	
<b>Copy To</b> section		
<b>Message</b>	Select a BPMN 1.1 <i>Message</i> in the package representing the <i>BPEL Process / Pool</i> under the <i>SupportingElements</i> package.	<a href="#">Create BPEL 1.1 Model Structure</a>  <a href="#">Create a BPEL 1.1 Web Service</a> 
<b>Part</b>	Select a BPMN 1.1 <i>Property</i> belonging to the selected <b>Message</b> (Optional).	
<b>OK</b>	Save the values entered in the dialog.	
<b>Cancel</b>	Discard the values entered in the dialog.	
<b>Help</b>	Display this Help topic.	
<b>General</b>	Open the UML Properties dialog.	<a href="#">Properties</a> 

#### Notes

- Messages are created when you create Web Services
- If you select **Expression**, Enterprise Architect uses the *getVariableData XPATH 1.0* function to create the expression from the selected Message and Part

#### 9.3.5.1.3 Create BPEL 1.1 Web Service

BPEL is an **Orchestration** language, which orchestrates services that are exposed using **WSDL 1.1**. It coordinates the execution of the various operations of these involved services. BPEL 1.1 supports **WSDL 1.1 one-way** and **request-response** operations only (and not **solicit-response** and **notification** operations). Enterprise Architect enables you to create, for *BPEL Process* and *Pool* elements, **Web Service Operations** that support either *synchronous* (request-response) or *asynchronous* (one-way) interactions.

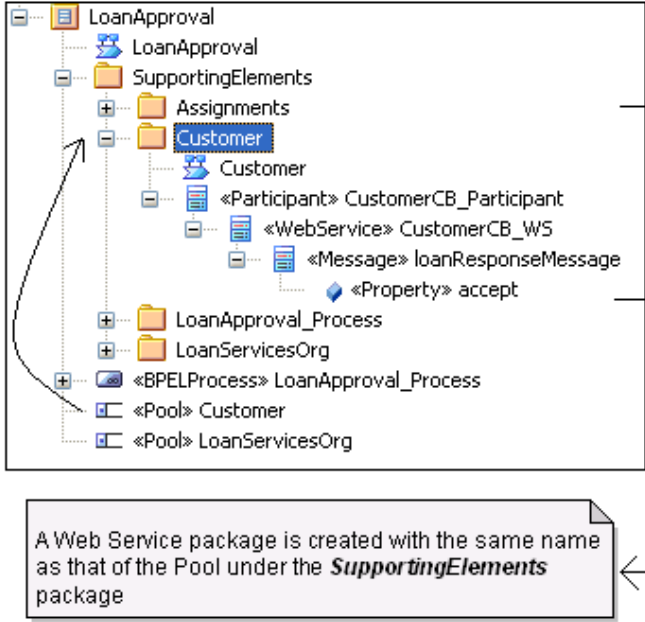
**Access** Right-click on a *BPEL Process* or *Pool* | **BPEL | Create WebService**

#### Reference

Field/Button	Usage	See also
<b>Operation</b>	Specify whether to create a new Web Service or create one from an existing WSDL created / imported into Enterprise Architect	<a href="#">Model WSDL</a>  <a href="#">Import WSDL</a> 

	( using the Import WSDL dialog) by selecting either : <ul style="list-style-type: none"> <li>• <b>Create New</b> ( or )</li> <li>• <b>Create from existing WSDL</b></li> </ul>																	
<b>Type</b>	Specify whether to create a <i>one-way</i> or <i>request-response</i> Operation by selecting either : <ul style="list-style-type: none"> <li>• <b>Asynchronous</b> ( or )</li> <li>• <b>Synchronous</b></li> </ul>																	
<b>Partnership Details</b>	<p>The interaction between the Web Service and the <i>BPEL Process</i> is modeled as a BPEL <i>partnerLink</i>. For this interaction, provide a:</p> <table border="1"> <thead> <tr> <th>Field</th><th>Usage</th></tr> </thead> <tbody> <tr> <td><b>Name</b></td><td>Name of the BPEL <i>partnerLink</i>.</td></tr> <tr> <td><b>Role</b></td><td>Name of either: <ul style="list-style-type: none"> <li>• <i>myRole</i> ( if this operation belongs to <i>BPEL Process</i> Web Service ) or</li> <li>• <i>partnerRole</i> ( if this operation belongs to Web Service of a <i>Pool</i> ) of the BPEL <i>partnerLink</i>.</li> </ul> </td></tr> </tbody> </table>	Field	Usage	<b>Name</b>	Name of the BPEL <i>partnerLink</i> .	<b>Role</b>	Name of either: <ul style="list-style-type: none"> <li>• <i>myRole</i> ( if this operation belongs to <i>BPEL Process</i> Web Service ) or</li> <li>• <i>partnerRole</i> ( if this operation belongs to Web Service of a <i>Pool</i> ) of the BPEL <i>partnerLink</i>.</li> </ul>											
Field	Usage																	
<b>Name</b>	Name of the BPEL <i>partnerLink</i> .																	
<b>Role</b>	Name of either: <ul style="list-style-type: none"> <li>• <i>myRole</i> ( if this operation belongs to <i>BPEL Process</i> Web Service ) or</li> <li>• <i>partnerRole</i> ( if this operation belongs to Web Service of a <i>Pool</i> ) of the BPEL <i>partnerLink</i>.</li> </ul>																	
<b>Web Service</b>	<table border="1"> <thead> <tr> <th>Field</th><th>Usage</th></tr> </thead> <tbody> <tr> <td colspan="2">When <b>Create New</b> is selected in the <b>Operation</b> field:</td></tr> <tr> <td><b>Web Service Name</b></td><td>Name of the WSDL 1.1 Web Service.</td></tr> <tr> <td><b>PortType</b></td><td>Name of the WSDL 1.1 <i>PortType</i> (Interface)</td></tr> <tr> <td><b>Operation</b></td><td>Name of the WSDL 1.1 <i>PortType Operation</i>.</td></tr> <tr> <td colspan="2">When <b>Create from existing WSDL PortType Operation</b> is selected the <b>Operation</b> field:</td></tr> <tr> <td><b>WSDL Package</b></td><td>Select an existing WSDL package created / imported into Enterprise Architect.</td></tr> <tr> <td><b>Web Service Name</b></td><td>Name of the WSDL 1.1 Web Service.</td></tr> </tbody> </table>	Field	Usage	When <b>Create New</b> is selected in the <b>Operation</b> field:		<b>Web Service Name</b>	Name of the WSDL 1.1 Web Service.	<b>PortType</b>	Name of the WSDL 1.1 <i>PortType</i> (Interface)	<b>Operation</b>	Name of the WSDL 1.1 <i>PortType Operation</i> .	When <b>Create from existing WSDL PortType Operation</b> is selected the <b>Operation</b> field:		<b>WSDL Package</b>	Select an existing WSDL package created / imported into Enterprise Architect.	<b>Web Service Name</b>	Name of the WSDL 1.1 Web Service.	
Field	Usage																	
When <b>Create New</b> is selected in the <b>Operation</b> field:																		
<b>Web Service Name</b>	Name of the WSDL 1.1 Web Service.																	
<b>PortType</b>	Name of the WSDL 1.1 <i>PortType</i> (Interface)																	
<b>Operation</b>	Name of the WSDL 1.1 <i>PortType Operation</i> .																	
When <b>Create from existing WSDL PortType Operation</b> is selected the <b>Operation</b> field:																		
<b>WSDL Package</b>	Select an existing WSDL package created / imported into Enterprise Architect.																	
<b>Web Service Name</b>	Name of the WSDL 1.1 Web Service.																	

	<table><tr><td></td><td><b>Default:</b> name of the selected WSDL Package.</td></tr><tr><td><b>PortType</b></td><td>Select a WSDL 1.1 <i>PortType</i> from the selected WSDL Package.</td></tr><tr><td><b>Operation</b></td><td>Select a WSDL 1.1 <i>PortType Operation</i> in the selected <b>PortType</b>.</td></tr></table>		<b>Default:</b> name of the selected WSDL Package.	<b>PortType</b>	Select a WSDL 1.1 <i>PortType</i> from the selected WSDL Package.	<b>Operation</b>	Select a WSDL 1.1 <i>PortType Operation</i> in the selected <b>PortType</b> .							
	<b>Default:</b> name of the selected WSDL Package.													
<b>PortType</b>	Select a WSDL 1.1 <i>PortType</i> from the selected WSDL Package.													
<b>Operation</b>	Select a WSDL 1.1 <i>PortType Operation</i> in the selected <b>PortType</b> .													
<b>Input</b>	<table><tr><td>Field</td><td>Usage</td></tr><tr><td colspan="2">When <b>Create New</b> is selected in the <b>Operation</b> field:</td></tr><tr><td><b>Message Name</b></td><td>Name of the WSDL 1.1 <i>Message</i>.</td></tr><tr><td><b>Properties</b></td><td>Press ( ... ) to enter the WSDL 1.1 <i>Message Name</i> and <b>XSD Type</b>.</td></tr><tr><td colspan="2"></td></tr><tr><td colspan="2">When <b>Create from existing WSDL PortType Operation</b> is selected in the <b>Operation</b> field:  The fields in this tab are pre-filled with the details of the input WSDL <i>Message</i> (of the WSDL 1.1 <i>PortType Operation</i> selected in the <b>Operation</b> field in the Web Service tab).</td></tr></table>	Field	Usage	When <b>Create New</b> is selected in the <b>Operation</b> field:		<b>Message Name</b>	Name of the WSDL 1.1 <i>Message</i> .	<b>Properties</b>	Press ( ... ) to enter the WSDL 1.1 <i>Message Name</i> and <b>XSD Type</b> .			When <b>Create from existing WSDL PortType Operation</b> is selected in the <b>Operation</b> field:  The fields in this tab are pre-filled with the details of the input WSDL <i>Message</i> (of the WSDL 1.1 <i>PortType Operation</i> selected in the <b>Operation</b> field in the Web Service tab).		
Field	Usage													
When <b>Create New</b> is selected in the <b>Operation</b> field:														
<b>Message Name</b>	Name of the WSDL 1.1 <i>Message</i> .													
<b>Properties</b>	Press ( ... ) to enter the WSDL 1.1 <i>Message Name</i> and <b>XSD Type</b> .													
When <b>Create from existing WSDL PortType Operation</b> is selected in the <b>Operation</b> field:  The fields in this tab are pre-filled with the details of the input WSDL <i>Message</i> (of the WSDL 1.1 <i>PortType Operation</i> selected in the <b>Operation</b> field in the Web Service tab).														
<b>Output</b>	<table><tr><td>Field</td><td>Usage</td></tr><tr><td colspan="2">When <b>Create New</b> is selected in the <b>Operation</b> field:</td></tr><tr><td><b>Message Name</b></td><td>The name of the WSDL 1.1 <i>Message</i>.</td></tr><tr><td><b>Properties</b></td><td>Press the ( ... ) button to enter the WSDL 1.1 <i>Message Part Name</i> and <b>XSD Type</b>.</td></tr><tr><td colspan="2"></td></tr><tr><td colspan="2">When <b>Create from existing WSDL PortType Operation</b> is selected in the <b>Operation</b> field:  The fields in this tab are pre-filled with the details of the output WSDL <i>Message</i> (of the WSDL 1.1 <i>PortType Operation</i> selected in the <b>Operation</b> field in the Web Service tab).</td></tr></table>	Field	Usage	When <b>Create New</b> is selected in the <b>Operation</b> field:		<b>Message Name</b>	The name of the WSDL 1.1 <i>Message</i> .	<b>Properties</b>	Press the ( ... ) button to enter the WSDL 1.1 <i>Message Part Name</i> and <b>XSD Type</b> .			When <b>Create from existing WSDL PortType Operation</b> is selected in the <b>Operation</b> field:  The fields in this tab are pre-filled with the details of the output WSDL <i>Message</i> (of the WSDL 1.1 <i>PortType Operation</i> selected in the <b>Operation</b> field in the Web Service tab).		
Field	Usage													
When <b>Create New</b> is selected in the <b>Operation</b> field:														
<b>Message Name</b>	The name of the WSDL 1.1 <i>Message</i> .													
<b>Properties</b>	Press the ( ... ) button to enter the WSDL 1.1 <i>Message Part Name</i> and <b>XSD Type</b> .													
When <b>Create from existing WSDL PortType Operation</b> is selected in the <b>Operation</b> field:  The fields in this tab are pre-filled with the details of the output WSDL <i>Message</i> (of the WSDL 1.1 <i>PortType Operation</i> selected in the <b>Operation</b> field in the Web Service tab).														

<b>OK</b>	<p>Create a Web Service Operation based on the values entered in the dialog.</p> <p>The Operation is created in a package, whose name is the same as that of the <i>BPEL Process / Pool</i> from which this dialog is invoked, under the <i>SupportingElements</i> package.</p> 	<a href="#">SupportingElements</a> <small>1897</small>
<b>Cancel</b>	<p>Discard the values entered in the dialog and abort creating the Web Service Operation.</p>	
<b>Help</b>	<p>Display this Help topic.</p>	

#### Notes

- The Output tab is not applicable for an Asynchronous operation

#### 9.3.5.1.4 Generate BPEL 1.1

**BPEL 1.1** code can be generated from a *BPEL Process*. Enterprise Architect validates the *BPEL Process* before generating the **BPEL 1.1** code. In addition to generating the **BPEL 1.1** code, **WSDL 1.1** files are generated for the *BPEL Process* and all the involved *Pools* ( provided **Web Services** are defined for them ).

**Access** Right-click on a BPEL Process and select the following context menu: **BPEL | Generate BPEL**

#### Reference

Field/Button	Usage	See also
<b>File Name</b>	Specify the path where the BPEL 1.1 file is to be generated.	
<b>Namespace Details</b>	Double-click on an entry ( if any ) in this field to open the Namespace Details dialog and add / edit the namespace details.  The entry <b>DefaultPool</b> represents the current <i>BPEL Process</i> .	
<b>Generate BPEL</b>	Validate the model and generate BPEL 1.1.	<a href="#">BPEL Model Validation</a> <sup>[1917]</sup>
<b>Close</b>	Close this dialog.	
<b>Help</b>	Display this Help topic.	
<b>View BPEL</b>	View the generated BPEL 1.1 file.	

#### Learn more

- [Model a BPEL 1.1 Process](#)<sup>[1876]</sup>
- [Create a BPEL 1.1 Web Service](#)<sup>[1890]</sup>

### 9.3.5.2 BPEL 2.0 Model

Enterprise Architect uses BPMN 2.0 as a graphical front-end to capture BPEL 2.0 *Process* description (as BPEL 2.0 does not have a standard graphical notation). Enterprise Architect uses the partial mapping specified in the BPMN 2.0 specification to map BPMN 2.0 constructs to BPEL 2.0.

#### How to

Step	Action	See also
1	Create a BPEL 2.0 Package Structure.	<a href="#">BPEL 2.0 Package Structure</a> <sup>[1897]</sup>  <

	<table><tr><td></td><td></td></tr><tr><td>d</td><td>Click on the <b>BPEL</b> icon from the BPMN 2.0 - Business Process To into the created BPEL diagram.</td></tr></table>			d	Click on the <b>BPEL</b> icon from the BPMN 2.0 - Business Process To into the created BPEL diagram.	
d	Click on the <b>BPEL</b> icon from the BPMN 2.0 - Business Process To into the created BPEL diagram.					
2	Open the BPEL diagram under the <i>BPELProcess</i> stereotyped element.					
3	<p>Model the <i>BPEL Process</i> using the constructs in the BPMN 2.0 - Business Process Toolbox. Use the following constructs in this toolbox to model BPEL 2.0:</p> <p>BPMN 2.0 - Business Process page:</p> <ul style="list-style-type: none"><li>• BPEL</li><li>• Activity</li><li>• Data Object</li><li>• Start Event</li><li>• Intermediate Event</li><li>• Event</li><li>• Gateway</li><li>• Pool</li></ul> <p>BPMN 2.0 - Business Process Connectors page:</p> <ul style="list-style-type: none"><li>• Sequence Flow</li><li>• Association</li><li>• Message Flow</li></ul> <p>BPMN 2.0 Types page:</p> <ul style="list-style-type: none"><li>• Assignment</li><li>• Error</li><li>• Interface</li><li>• Operation</li><li>• Participant</li><li>• Property</li></ul>	<p><a href="#">BPMN 2.0 Business Process Toolbox</a><sup>[1852]</sup></p> <p><a href="#">Model BPEL 2.0 Process</a><sup>[1898]</sup></p>				
4	Create Web Service Operations for the <i>BPEL Process</i> and other related <i>Pools</i> , if required.	<p><a href="#">Web Service Operation</a><sup>[1914]</sup></p>				
5	Generate BPEL 2.0 code from the <i>BPEL Process</i> .	<p><a href="#">Generate BPEL 2.0</a><sup>[1916]</sup></p>				

**Notes**

- Generating BPEL 2.0 from a BPMN 2.0 model is supported in the Business and Software Engineering

and the Ultimate editions of Enterprise Architect

- Ensure that BPMN 2.0 Technology is enabled in the MDG Technologies dialog (**Settings | MDG Technologies**)

#### **BPEL 2.0 Example:**

- The Enterprise Architect Example file ( *EAExample.EAP* ) has a sample BPMN 2.0 model from which BPEL 2.0 can be generated
- If you have installed Enterprise Architect at the default location, open this file: *C:\Program Files\Sparx Systems\EA\EAExample.EAP*
- The BPMN 2.0 model package is in: **Project Models | Analysis and Business Modeling | BPEL Example | BPEL 2.0 Model**

#### **Modeling Restrictions:**

- Every BPEL *Process* and *Sub-Process* should start with a *Start Event* and end with an *End Event*
  - Boundary *Start* and *End Events* are not supported
- *Sequence Flow* Looping is not supported in both **Normal** and **Exception Paths**
  - All *Sequence Flows* should flow downstream and not upstream
- *Sub-Process* cannot be a **Loop** node and have boundary *Intermediate Events*
- *Event Sub-Process* cannot act as a **Loop** Node
- *Assignments* are not supported on:
  - *Start Events* on *Event Sub-Process*
  - *End Events*
  - *Sub-Process*
  - Boundary *Intermediate Event*
  - *Gateway*
  - *Task* and *Intermediate Event* that immediately follow an *XOR Event Gateway*
- **Exception Path** has to merge back into the **Normal Path**
  - An exception to this rule is boundary *Compensation Intermediate Event* which should have a BPMN 2.0 *Association* to a *Compensation Activity* (that has no incoming or outgoing *Sequence Flows*)
  - Multiple **Exception Paths** from an *Activity* must join at the same location in the **Normal Path**
  - An **Exception Path** should not cross another **Exception Path**
- *Activities* in an **Exception Path** cannot have boundary **Intermediate Events**.
- *Pools* are treated as *black boxes* (that is, they do not expose any details) and hence they cannot contain any child elements or have any incoming/outgoing *Sequence Flow* connectors

#### **Learning Center topics**

- (Alt+F1) | **Enterprise Architect | Business Modeling | BPEL | Getting Started**



### 9.3.5.2.1 Create BPEL 2.0 Model Structure

A BPEL 2.0 model consists of a *BPEL Process* (containing a BPEL diagram and mappable BPMN 2.0 constructs) and other supporting elements (like BPMN 2.0 *Assignment*, BPMN 2.0 *Operation*) required for generating a BPEL 2.0 code.

A sample BPEL 2.0 package structure can be created in the Project Browser, using the Select Model(s) (Model Wizard) dialog. You can use this package structure as a template for developing your *BPEL Process*.

#### Access

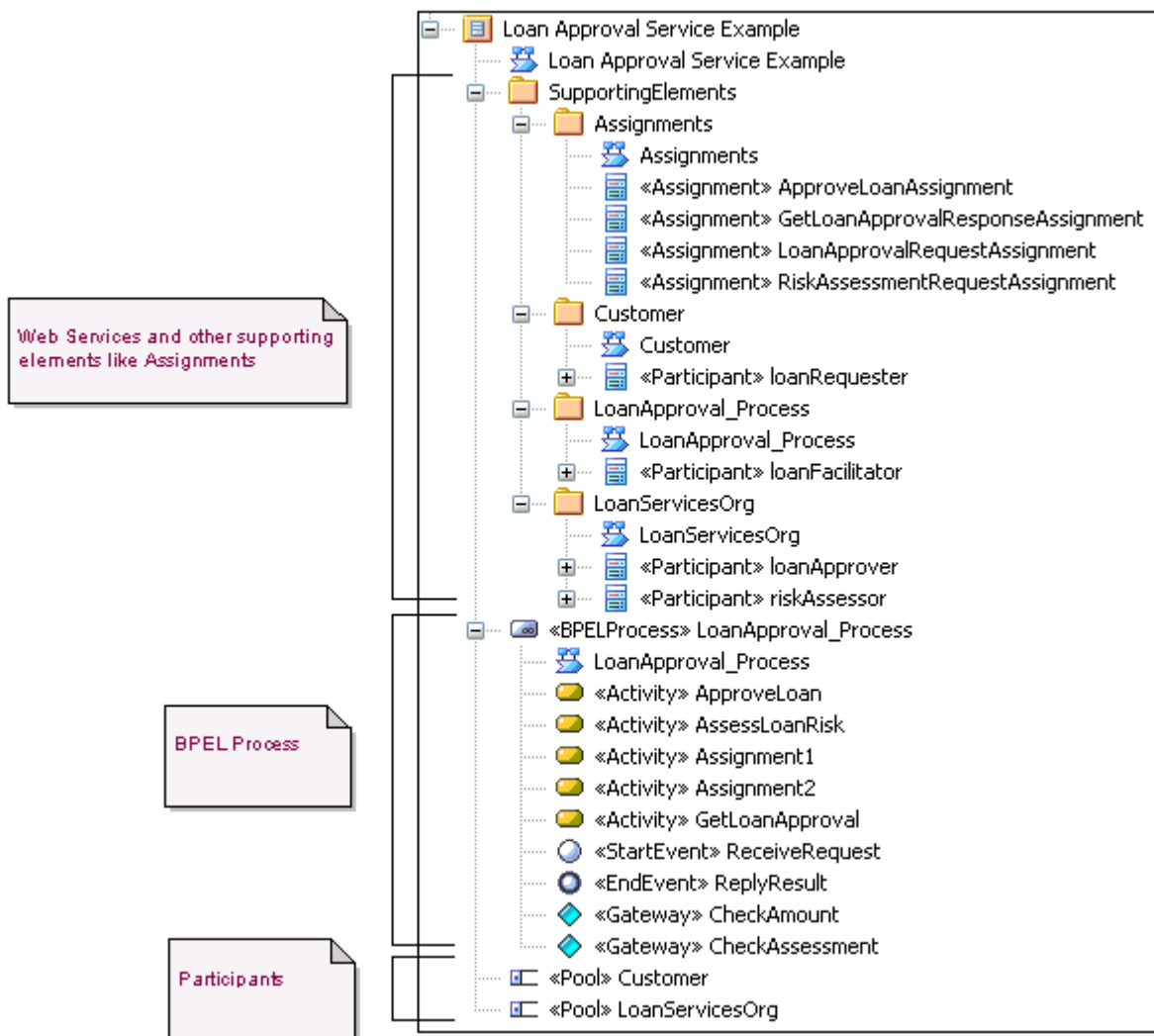
- **Project Browser package context menu | Add a Model using Wizard**, or
- **Project Browser toolbar: New Model from Pattern**

#### Create a BPEL 2.0 Package structure as a starting point for developing a BPEL Process

Step	Action	See also
1	Select the root node or a package in the Project Browser.	
2	Click on the <b>New Model from Pattern</b> icon in the Project Browser toolbar.	<a href="#">Project Browser Toolbar</a> <sup>[669]</sup>
3	Select the value <b>BPMN 2.0</b> in the <b>Technology</b> section.	<a href="#">Model Wizard</a> <sup>[753]</sup>
4	Check the option <b>BPEL 2.0 Model</b> in the <b>Name</b> section.	
5	Click on the <b>OK</b> button to create the sample BPEL 2.0 package structure.	

#### Example BPEL 2.0 Package Structure:

The *BPEL Process* **LoanApproval\_Process** acts as container for the BPEL diagram and elements. The *SupportingElements* package contains supporting components like **Assignments** and **Web Service Operations**.



#### Learn more

- [Model a BPEL 2.0 Process](#) <sup>[1898]</sup>
- [Create a Web Service Operation](#) <sup>[1914]</sup>
- [Generate BPEL 2.0](#) <sup>[1916]</sup>

#### 9.3.5.2.2 Model a BPEL 2.0 Process

The *BPEL Process* in Enterprise Architect represents the top-level container for the **BPMN 2.0** elements, from which **BPEL 2.0** can be generated. Conceptually it maps to the BPEL *process* element.

#### BPEL Properties:

- Double-click on the *BPEL Process* in the **BPEL** diagram ( or )
- Right-click on the *BPEL Process* in the **BPEL** diagram ( or the **Project Browser** ) and select the following context menu : **BPEL | BPEL 2.0 Properties**

**Reference**

Field/Button	Usage	See also
<b>Name</b>	Specify the name for the <i>BPEL Process</i> .	
<b>Query Language</b>	Specify the value of the language used in the <i>BPEL Process</i> for the selection of nodes in <i>Assignments</i> . Defaults to <b>XPath 1.0</b> .	
<b>OK</b>	Save the values entered in the dialog.	
<b>Cancel</b>	Discard the values entered in the dialog.	
<b>Help</b>	Display this Help topic.	
<b>General</b>	Open the UML Properties dialog.	<a href="#">Properties</a> [956]

**Learn more**

The *BPEL Process* element acts as a container for the following BPMN 2.0 constructs:

- [Start Event](#) [1900]
- [Intermediate Event](#) [1901]
- [Activity](#) [1904]
- [Gateway](#) [1906]
- [End Event](#) [1907]
- [Data Object](#) [1909]
- [Property](#) [1910]
- [Sequence Flow](#) [1910]

The following BPMN 2.0 constructs can be optionally used in BPEL modeling:

- [Pool](#) [1888]
- [Assignment](#) [1889]

**Learning Center topics**

- (Alt+F1) | **Enterprise Architect | Business Modeling | BPEL | Process Modeling**

### 9.3.5.2.2.1 Start Event

ABPMN 2.0 *Start Event* indicates where a particular *Process* begins. Every *Process* in Enterprise Architect must begin with a *Start Event*.

#### BPEL Properties:

- Double-click on the *Start Event* in the BPEL diagram, or
- Right-click on the *Start Event* | **BPEL** | **BPEL 2.0 Properties**

#### Reference

Field/Button	Usage	See also										
Name	Specify the name for the <i>Start Event</i> .											
Event Type	Select a trigger for the <i>Start Event</i> , namely: <ul style="list-style-type: none"><li>• Compensation</li><li>• Conditional</li><li>• Error</li><li>• Escalation</li><li>• Message</li><li>• Multiple</li><li>• None</li><li>• Parallel Multiple</li><li>• Signal</li><li>• Timer</li></ul>											
Details	<div>Depending on the selected <b>Event Type</b>, the Details tab changes as follows:</div> <table><tr><th>Field</th><th>Usage</th></tr><tr><td colspan="2"><b>Message Event Type</b></td></tr><tr><td><b>Pool Package</b></td><td>Select the package that represents one of the involved <i>Participants</i>.</td></tr><tr><td><b>Operation</b></td><td>Select an Operation from the list of operations selected <b>Pool Package</b>.</td></tr><tr><td colspan="2"><b>Timer Event Type</b></td></tr></table>	Field	Usage	<b>Message Event Type</b>		<b>Pool Package</b>	Select the package that represents one of the involved <i>Participants</i> .	<b>Operation</b>	Select an Operation from the list of operations selected <b>Pool Package</b> .	<b>Timer Event Type</b>		
Field	Usage											
<b>Message Event Type</b>												
<b>Pool Package</b>	Select the package that represents one of the involved <i>Participants</i> .											
<b>Operation</b>	Select an Operation from the list of operations selected <b>Pool Package</b> .											
<b>Timer Event Type</b>												

	<b>Time Cycle</b>	Specify the value for the time duration.	
	<b>Time Date</b>	Specify the value for the time date.	
	<b>Error Event Type</b>		
	<b>Error</b>	Select an <i>Error</i> element.	
<b>Assignments</b>	Select one or more <i>Assignment</i> elements created in the <i>SupportingElements</i> package in this tab (Optional).		<a href="#">Assignment</a> <sup>[1912]</sup> <a href="#">SupportingElements</a> <sup>[1897]</sup>
<b>OK</b>	Save the values entered in the dialog.		
<b>Cancel</b>	Discard the values entered in the dialog.		
<b>Help</b>	Display this Help topic.		
<b>General</b>	Open the UML Properties dialog.		<a href="#">Properties</a> <sup>[956]</sup>

**Notes**

- Either set **Time Cycle** or **Time Date** for **Timer Event Type** but not both, as they are mutually exclusive fields
- The Assignments tab is not available on *Start Events* that are used to start an *Event Sub-Process*
- **Compensation**, **Error** and **Timer** event types are valid only on an *Event Sub-Process*
- **Conditional**, **Escalation**, **Multiple**, **Parallel Multiple** and **Signal** event types cannot be mapped to BPEL 2.0

**9.3.5.2.2.2 Intermediate Event**

A BPMN 2.0 *Intermediate Event* indicates where an event occurs somewhere between the start and end of a *Process*.

**BPEL Properties:**

- Double-click on the *Intermediate Event* in the BPEL diagram
- Right-click on the *Intermediate Event* | **BPEL** | **BPEL 2.0 Properties**

**Reference**

Field/Button	Usage	See also																		
Name	Specify the name for the <i>Intermediate Event</i>																			
Event Type	Select the type for the <i>Intermediate Event</i> , namely: <ul style="list-style-type: none"><li>Cancel</li><li>Compensation</li><li>Conditional</li><li>Error</li><li>Escalation</li><li>Link</li><li>Message</li><li>Multiple</li><li>None</li><li>Parallel Multiple</li><li>Signal</li><li>Timer</li></ul>																			
Details	<div>Depending on the selected <b>Event Type</b>, the Details tab changes as follows:</div> <table><tr><th>Field</th><th>Usage</th></tr><tr><td colspan="2">Message Event Type</td></tr><tr><td>Pool Package</td><td>Select the package that represents one of involved <i>Participants</i>.</td></tr><tr><td>Operation</td><td>Select an Operation from the list of operations under the selected <b>Pool Package</b>.</td></tr><tr><td colspan="2">Link Event Type</td></tr><tr><td>Link Event</td><td>Select a <i>Link Event</i> element (to act as a target for this <i>Link Event</i>) from the list of <i>Link Event</i> under the current <i>Process / Sub-Process</i>.</td></tr><tr><td colspan="2">Timer Event Type</td></tr><tr><td>Time Cycle</td><td>Specify the value for the time duration.</td></tr><tr><td>Time Date</td><td>Specify the value for the time date.</td></tr></table>	Field	Usage	Message Event Type		Pool Package	Select the package that represents one of involved <i>Participants</i> .	Operation	Select an Operation from the list of operations under the selected <b>Pool Package</b> .	Link Event Type		Link Event	Select a <i>Link Event</i> element (to act as a target for this <i>Link Event</i> ) from the list of <i>Link Event</i> under the current <i>Process / Sub-Process</i> .	Timer Event Type		Time Cycle	Specify the value for the time duration.	Time Date	Specify the value for the time date.	
Field	Usage																			
Message Event Type																				
Pool Package	Select the package that represents one of involved <i>Participants</i> .																			
Operation	Select an Operation from the list of operations under the selected <b>Pool Package</b> .																			
Link Event Type																				
Link Event	Select a <i>Link Event</i> element (to act as a target for this <i>Link Event</i> ) from the list of <i>Link Event</i> under the current <i>Process / Sub-Process</i> .																			
Timer Event Type																				
Time Cycle	Specify the value for the time duration.																			
Time Date	Specify the value for the time date.																			

	<table><tr><td></td><td></td></tr><tr><td colspan="2"></td></tr><tr><td colspan="2"><b>Compensation Event Type</b></td></tr><tr><td><b>Activity</b></td><td>Select an <i>Activity</i> from the list of Activity elements under the current <i>Process</i> / <i>Sub-Process</i>.</td></tr><tr><td colspan="2"></td></tr><tr><td colspan="2"><b>Error Event Type</b></td></tr><tr><td><b>Error</b></td><td>Select an <i>Error</i> element.</td></tr><tr><td colspan="2"></td></tr></table>					<b>Compensation Event Type</b>		<b>Activity</b>	Select an <i>Activity</i> from the list of Activity elements under the current <i>Process</i> / <i>Sub-Process</i> .			<b>Error Event Type</b>		<b>Error</b>	Select an <i>Error</i> element.			
<b>Compensation Event Type</b>																		
<b>Activity</b>	Select an <i>Activity</i> from the list of Activity elements under the current <i>Process</i> / <i>Sub-Process</i> .																	
<b>Error Event Type</b>																		
<b>Error</b>	Select an <i>Error</i> element.																	
<b>Assignments</b>	Select one or more <i>Assignment</i> elements created in the <i>SupportingElements</i> package in this tab (Optional).	<a href="#">Assignment</a> <sup>[1912]</sup> <a href="#">SupportingElements</a> <sup>[1897]</sup>																
<b>OK</b>	Save the values entered in the dialog.																	
<b>Cancel</b>	Discard the values entered in the dialog.																	
<b>Help</b>	Display this Help topic.																	
<b>General</b>	Open the UML Properties dialog.	<a href="#">Properties</a> <sup>[956]</sup>																

### Notes

- **Cancel**, **Conditional**, **Escalation**, **Multiple**, **None**, **Parallel Multiple** and **Signal** event type cannot be mapped to BPEL 2.0
- The Assignments tab is not available on *Intermediate Events* that are attached to the boundary of an *Activity*
- Either set **Time Cycle** or **Time Date** for **Timer Trigger Type** but not both as they are mutually exclusive fields
- The **Error** event type is valid only on an *Intermediate Event* attached to the boundary of an *Activity*
- The **Compensation** event type is valid only on an *Event Sub-Process* or when attached to the boundary of an *Activity*
- The **Link Intermediate Event** can be used either as a *GOTO* or an *off-page* connector; therefore, this event can have either incoming or outgoing *Sequence Flows* - but not both

### 9.3.5.2.2.3 Activity

A BPMN 2.0 *Activity* represents work that is performed within a *Process*. An *Activity* can be modeled as a:

- **Sub-Process** - a compound *Activity* that is defined as a flow of other BPMN 2.0 elements or
- **Task** - an atomic *Activity* that cannot be broken down into a smaller unit

Activities - both *Tasks* and *Sub-Processes* - can also, optionally, act as *Looping constructs*. The *OMG BPMN 2.0 Specification* defines two types of *Looping construct*:

- **Standard Loop** (*while* or *until*)
- **Multi-Instance Loop** (*for each*)

#### BPEL Properties:


- Double-click on the *Activity* in the BPEL diagram
- Right-click on the *Activity* | **BPEL | BPEL 2.0 Properties**

#### Reference

Field/Option/ Button	Action	See also
<b>Name</b>	Type a name for the <i>Activity</i> .	
<b>Type</b>	Specify whether the <i>Activity</i> is a: <ul style="list-style-type: none"> <li>• <b>Task</b> or</li> <li>• <b>Sub-Process</b></li> </ul>	
<b>Task Type / SubProcess Type</b>	Depending on the value selected in the <b>Type</b> field, the <b>Task Type / SubProcess Type</b> field has the following values: <p><b>Task Type :</b></p> <ul style="list-style-type: none"> <li>• Abstract</li> <li>• Business Rule</li> <li>• Manual</li> <li>• Receive</li> <li>• Script</li> <li>• Send</li> <li>• Service</li> <li>• User</li> </ul> <p><b>SubProcess Type :</b></p> <ul style="list-style-type: none"> <li>• Ad-Hoc</li> <li>• Call Activity</li> <li>• Compensation</li> </ul>	



	<ul style="list-style-type: none"><li>• Embedded</li><li>• Event</li><li>• Transaction</li></ul>													
Details	<p>Depending on the selected <b>Task Type</b>, the Details tab changes as follows:</p> <table><tr><th>Field</th><th>Action</th></tr><tr><td colspan="2"><b>Receive / Send / Service Task Type</b></td></tr><tr><td><b>Pool Package</b></td><td>Select the package that represents one of the <i>Participants</i>.</td></tr><tr><td><b>Operation</b></td><td>Select an Operation from the list of operation( selected <b>Pool Package</b>.</td></tr></table>	Field	Action	<b>Receive / Send / Service Task Type</b>		<b>Pool Package</b>	Select the package that represents one of the <i>Participants</i> .	<b>Operation</b>	Select an Operation from the list of operation( selected <b>Pool Package</b> .					
Field	Action													
<b>Receive / Send / Service Task Type</b>														
<b>Pool Package</b>	Select the package that represents one of the <i>Participants</i> .													
<b>Operation</b>	Select an Operation from the list of operation( selected <b>Pool Package</b> .													
Assignments	(Optional) Select one or more <i>Assignment</i> elements created in the <i>SupportingElements</i> package in this tab.	<a href="#">Assignment</a> <sup>[1912]</sup> <a href="#">SupportingElements</a> <sup>[1897]</sup>												
Loop Details	<p>(Optional) Activities can be repeated sequentially, behaving like a loop. Specify the <i>Activity</i> looping details in this tab.</p> <table><tr><th>Field</th><th>Action</th></tr><tr><td><b>Loop Type</b></td><td>Select:<ul style="list-style-type: none"><li>• <b>Standard</b> for creating a <i>while</i> or <i>until</i> loop</li><li>• <b>Multinstance</b> for creating a <i>for each</i> loop</li></ul></td></tr><tr><td><b>Evaluate condition at the beginning of loop</b></td><td>Check this option to create a <i>while</i> <b>Standard</b> loop. Defaults to unselected for an <i>until</i> loop.</td></tr><tr><td><b>Loop Condition</b></td><td>Enter a boolean condition for a <b>Standard</b> loop condition for a <b>Multinstance</b> loop.</td></tr><tr><td><b>Loop Maximum</b></td><td>(Optional) Specify the maximum value for the for a <b>Standard</b> loop.</td></tr><tr><td><b>Loop Ordering</b></td><td>For a <b>Multinstance</b> Loop Type select:</td></tr></table>	Field	Action	<b>Loop Type</b>	Select: <ul style="list-style-type: none"><li>• <b>Standard</b> for creating a <i>while</i> or <i>until</i> loop</li><li>• <b>Multinstance</b> for creating a <i>for each</i> loop</li></ul>	<b>Evaluate condition at the beginning of loop</b>	Check this option to create a <i>while</i> <b>Standard</b> loop. Defaults to unselected for an <i>until</i> loop.	<b>Loop Condition</b>	Enter a boolean condition for a <b>Standard</b> loop condition for a <b>Multinstance</b> loop.	<b>Loop Maximum</b>	(Optional) Specify the maximum value for the for a <b>Standard</b> loop.	<b>Loop Ordering</b>	For a <b>Multinstance</b> Loop Type select:	
Field	Action													
<b>Loop Type</b>	Select: <ul style="list-style-type: none"><li>• <b>Standard</b> for creating a <i>while</i> or <i>until</i> loop</li><li>• <b>Multinstance</b> for creating a <i>for each</i> loop</li></ul>													
<b>Evaluate condition at the beginning of loop</b>	Check this option to create a <i>while</i> <b>Standard</b> loop. Defaults to unselected for an <i>until</i> loop.													
<b>Loop Condition</b>	Enter a boolean condition for a <b>Standard</b> loop condition for a <b>Multinstance</b> loop.													
<b>Loop Maximum</b>	(Optional) Specify the maximum value for the for a <b>Standard</b> loop.													
<b>Loop Ordering</b>	For a <b>Multinstance</b> Loop Type select:													

	<div> <div></div> <ul style="list-style-type: none"> <li>• <b>Parallel</b> for generating instances in parallel</li> <li>• <b>Sequential</b> for generating instances in sequence</li> </ul> </div>	
<b>OK</b>	Click on this button to save the values entered in the dialog.	
<b>Cancel</b>	Click on this button to discard the values entered in the dialog.	
<b>Help</b>	Click on this button to display this Help topic.	
<b>General</b>	Click on this button to open the UML Properties dialog.	<a href="#">Properties</a> 

#### Notes

- The Assignments tab is not applicable for *Sub-Process*
- The Loop Details tab is not applicable for *Event Sub-Process*
- **Business Rule**, **Manual**, **Script** and **User Task** types cannot be mapped to BPEL 2.0
- **Ad-Hoc**, **Call Activity** and **Transaction Sub-Process** types cannot be mapped to BPEL 2.0

#### 9.3.5.2.2.4 Gateway

A BPMN 2.0 *Gateway* controls the way in which *Sequence Flows* converge and diverge within a *Process*. They provide a gating mechanism that either allows or blocks a *Sequence Flow*.

#### BPEL Properties:

- Double-click on the *Gateway* in the BPEL diagram
- Right-click on the *Gateway* | **BPEL | BPEL 2.0 Properties**

#### Reference

Field/Button	Usage	See also
<b>Name</b>	Specify the name for the <i>Gateway</i> .	
<b>Gateway</b>	Select the type for the <i>Gateway</i> , namely: <ul style="list-style-type: none"> <li>• Complex</li> <li>• Event</li> <li>• Exclusive</li> <li>• Inclusive</li> <li>• Parallel</li> </ul>	

<b>Details</b>	<p>Depending on the selected <b>Gateway</b>, the Details tab changes as follows:</p> <table><tr><td>Field</td><td>Usage</td></tr><tr><td colspan="2"><b>Event Gateway</b></td></tr><tr><td><b>Event Type</b></td><td>Set to <b>Exclusive</b> to represent an Exclusive Event Based Gateway.</td></tr></table>	Field	Usage	<b>Event Gateway</b>		<b>Event Type</b>	Set to <b>Exclusive</b> to represent an Exclusive Event Based Gateway.	
Field	Usage							
<b>Event Gateway</b>								
<b>Event Type</b>	Set to <b>Exclusive</b> to represent an Exclusive Event Based Gateway.							
<b>OK</b>	Save the values entered in the dialog.							
<b>Cancel</b>	Discard the values entered in the dialog.							
<b>Help</b>	Display this Help topic.							
<b>General</b>	Open the UML Properties dialog.	<a href="#">Properties</a> 9561						

#### Notes

- The target of the outgoing *Sequence Flows* of this **Event Exclusive Gateway** must be either a:
  - Receive Task** or a
  - Message** or **Timer Intermediate Event**
- If a **Message Task** is one of the targets of the outgoing *Sequence Flow* of an **Event Exclusive Gateway**, then a **Message Intermediate Event** cannot be the target of the other outgoing *Sequence Flows* of this **Event Exclusive Gateway**

#### 9.3.5.2.2.5 End Event


A BPMN 2.0 *End Event* indicates where a particular *Process* ends. Every *Process* in Enterprise Architect must end with an *End Event*.

#### BPEL Properties:

- Double-click on the *End Event* in the BPEL diagram
- Right-click on the *End Event* | **BPEL** | **BPEL 2.0 Properties**

#### Reference


Field/Button	Usage	See also																
Name	Specify the name for the <i>End Event</i> .																	
Event Type	Select the type for the <i>End Event</i> , namely: <ul style="list-style-type: none"><li>• Cancel</li><li>• Compensation</li><li>• Error</li><li>• Escalation</li><li>• Message</li><li>• Multiple</li><li>• None</li><li>• Signal</li><li>• Terminate</li></ul>																	
Details	<div>Depending on the selected <b>Event Type</b>, the Details tab changes as follows:</div> <table><tr><th>Field</th><th>Usage</th></tr><tr><td colspan="2"><b>Message Event Type</b></td></tr><tr><td><b>Pool Package</b></td><td>Select the package that represents one of the involved <i>Participants</i>.</td></tr><tr><td><b>Operation</b></td><td>Select an Operation from the list of operation the selected <b>Pool Package</b>.</td></tr><tr><td colspan="2"><b>Compensation Event Type</b></td></tr><tr><td><b>Activity</b></td><td>Select an <i>Activity</i> from the list of Activity element under the current <i>Process / Sub-Process</i>.</td></tr><tr><td colspan="2"><b>Error Event Type</b></td></tr><tr><td><b>Error</b></td><td>Select an <i>Error</i> element.</td></tr></table>	Field	Usage	<b>Message Event Type</b>		<b>Pool Package</b>	Select the package that represents one of the involved <i>Participants</i> .	<b>Operation</b>	Select an Operation from the list of operation the selected <b>Pool Package</b> .	<b>Compensation Event Type</b>		<b>Activity</b>	Select an <i>Activity</i> from the list of Activity element under the current <i>Process / Sub-Process</i> .	<b>Error Event Type</b>		<b>Error</b>	Select an <i>Error</i> element.	
Field	Usage																	
<b>Message Event Type</b>																		
<b>Pool Package</b>	Select the package that represents one of the involved <i>Participants</i> .																	
<b>Operation</b>	Select an Operation from the list of operation the selected <b>Pool Package</b> .																	
<b>Compensation Event Type</b>																		
<b>Activity</b>	Select an <i>Activity</i> from the list of Activity element under the current <i>Process / Sub-Process</i> .																	
<b>Error Event Type</b>																		
<b>Error</b>	Select an <i>Error</i> element.																	
OK	Save the values entered in the dialog.																	

<b>Cancel</b>	Discard the values entered in the dialog.	
<b>Help</b>	Display this Help topic.	
<b>General</b>	Open the UML Properties dialog.	<a href="#">Properties</a> 

#### Notes

- The **Compensation** Event type is valid only on an *Event Sub-Process*
- **Cancel**, **Escalation**, **Multiple** and **Signal** Event types cannot be mapped to BPEL 2.0


#### 9.3.5.2.2.6 Data Object

A BPMN 2.0 *Data Object* is mapped to a BPEL 2.0 *Variable* and occurs in the context of a Process or Sub-Process, just like a [Property](#) . A *Data Object* cannot have any incoming or outgoing *Sequence Flow* or *Message Flow*.

#### BPEL Properties:

- Double-click on the *Data Object* in the BPEL diagram
- Right-click on the *Data Object* | **BPEL** | **BPEL 2.0 Properties**

#### Reference

Field/Button	Usage	See also
<b>Name</b>	Specify the name for the <i>Data Object</i> .	
<b>Type</b>	Either: <ul style="list-style-type: none"> <li>• Select an <b>XML Schema</b> <i>basic</i> type from the drop-down list or</li> <li>• Use the ( ... ) button to select <b>XML Schema</b> <i>ComplexType</i>, <i>SimpleType</i>, <i>Union</i> or <i>Enumeration</i></li> </ul>	
<b>OK</b>	Save the values entered in the dialog.	
<b>Cancel</b>	Discard the values entered in the dialog.	
<b>Help</b>	Display this Help topic.	
<b>General</b>	Open the UML Properties dialog.	<a href="#">Properties</a> 

### 9.3.5.2.2.7 Property

A BPMN 2.0 *Property* is mapped to a BPEL 2.0 *Variable* and occurs in the context of a *Process* or *Sub-Process*, just like a [Data Object](#) <sup>[1909]</sup>.

However, unlike a *Data Object*, *Property* is not displayed on a BPEL diagram. Enterprise Architect supports *Property* on *BPMN 2.0 BPEL* and *BPMN 2.0 Activity* elements.

[Access](#) **Diagram | Diagram Toolbox: More tools | BPMN 2.0 | BPMN 2.0 Business Process | BPMN 2.0 Types**

#### BPEL Properties:

- Right-click on the *Property* | **BPEL | BPEL 2.0 Properties**

#### Reference

Field/Button	Usage	See also
<b>Name</b>	Specify the name for the <i>Property</i> .	
<b>Type</b>	Either: <ul style="list-style-type: none"> <li>• Select an <b>XML Schema</b> <i>basic</i> type from the drop-down list or</li> <li>• Use the ( ... ) button to select <b>XML Schema</b> <i>ComplexType</i>, <i>SimpleType</i>, <i>Union</i> or <i>Enumeration</i></li> </ul>	
<b>OK</b>	Save the values entered in the dialog.	
<b>Cancel</b>	Discard the values entered in the dialog.	
<b>Help</b>	Display this Help topic.	
<b>General</b>	Open the UML Properties dialog.	<a href="#">General Properties of Attributes</a> <sup>[1001]</sup>

### 9.3.5.2.2.8 Sequence Flow

A BPMN 2.0 **Sequence Flow** connector shows the order in which the activities (**Tasks** and **Events**) are performed in a **BPEL Process**.

#### Access BPEL Properties

- Double-click on the Sequence Flow in the BPEL diagram
- Right-click on the Sequence Flow | **BPEL | BPEL 2.0 Properties**

Define the Sequence Flow

Field/Button	Action	See also
<b>Condition Type</b>	Click on the drop-down arrow and select the type of the condition on the Sequence Flow, namely: <ul style="list-style-type: none"> <li>• <b>None</b></li> <li>• <b>Default</b></li> <li>• <b>Expression</b></li> </ul>	
<b>Expression</b>	This field is enabled when the <b>Condition Type</b> is set to <b>Expression</b> . Type in a boolean expression to act as a gating condition.	
<b>Ordering</b>	This field is enabled when the <b>Condition Type</b> is set to <b>Expression</b> . Type in a numerical value that determines the order in which the condition set in the <b>Expression</b> field is to be evaluated.	
<b>OK</b>	Click on this button to save the values entered in the dialog.	
<b>Cancel</b>	Click on this button to discard the values entered in the dialog.	
<b>Help</b>	Click on this button to display this Help topic.	
<b>General</b>	Click on this button to open the UML Properties dialog.	<a href="#">Properties</a> [956]

**9.3.5.2.2.9 Pool**

A BPMN 2.0 *Pool* represents a Participant in a *Process* and does not map to any specific BPEL 2.0 element. Enterprise Architect uses *Pools* to represent external Participants, with which the *BPEL Process* communicates. These are 'black box' *pools*; that is, they are abstract and do not expose any details (they do not contain any BPMN 2.0 elements inside them).

**BPEL Properties:**

- Double-click on the *Pool* in the BPEL diagram
- Right-click on the *Pool* | **BPEL** | **BPEL 2.0 Properties**

**Reference**

Field/Button	Usage	See also
--------------	-------	----------

<b>Name</b>	Specify the name for the <i>Pool</i> .	
<b>OK</b>	Save the values entered in the dialog.	
<b>Cancel</b>	Discard the values entered in the dialog.	
<b>Help</b>	Display this Help topic.	
<b>General</b>	Open the UML Properties dialog.	<a href="#">Properties</a> 9561

#### Notes

- A *BPEL Process* should not contain a *Pool* as its child element
- A *BPEL* diagram under a *BPEL Process* contains an implicit *Pool*, so it is invalid to add a *Pool* in this *BPEL* diagram to represent the *BPEL Process*
- *Pool* cannot have any incoming or outgoing *Sequence Flow* connectors - it can have only incoming or outgoing *Message Flow* connectors

#### 9.3.5.2.2.10 Assignment

A BPMN 2.0 *Assignment* element enables data to be copied between messages and variables within a *BPEL Process*. An *Assignment* element maps to a BPEL 2.0 *assign* activity and copies the specified value from the source to the target.

In Enterprise Architect, *Assignment* elements should be created in the *Assignments* package in *SupportingElements*. If they are created elsewhere, they cannot be enacted correctly.

**Access** [Diagram](#) | **Diagram Toolbox: More tools** | **BPMN 2.0** | **BPMN 2.0 Business Process** | **BPMN 2.0 Types**

#### BPEL Properties:

- Double-click on the *Assignment* in the **BPEL** diagram
- Right-click on the *Assignment* | **BPEL** | **BPEL 2.0 Properties**

#### Reference

Field/Button	Usage	See also
<b>Name</b>	Specify the name for the <i>Assignment</i> .	
<b>Copy From</b>		



<b>Type</b>	Depending on the value selected in this field, further details are required.	
<b>Literal</b>	Specify a literal value.	
<b>Expression</b>	Specify an expression.	
<b>Message</b>	Select a BPMN 2.0 <i>Message</i> in the package representing the <i>BPEL Process / Pool</i> under the <i>SupportingElements</i> package.	<a href="#">SupportingElement s</a> <sup>[1897]</sup> <a href="#">Web Service Operation</a> <sup>[1914]</sup>
<b>Part</b>	Select a BPMN 2.0 <i>Property</i> belonging to the selected <b>Message</b> .	
<b>Variable</b>	Select a BPMN 2.0 <i>Data Object</i> or BPMN 2.0 <i>Property</i> created under the <i>BPEL Process</i> .	
<b>Copy To</b>		
<b>Message</b>	Select a BPMN 2.0 <i>Message</i> in the package representing the <i>BPEL Process / Pool</i> under the <i>SupportingElements</i> package.	<a href="#">SupportingElement s</a> <sup>[1897]</sup> <a href="#">Web Service Operation</a> <sup>[1914]</sup>
<b>Part</b>	Select a BPMN 2.0 <i>Property</i> belonging to the selected <b>Message</b> .	
<b>Variable</b>	Select a BPMN 2.0 <i>Data Object</i> or BPMN 2.0 <i>Property</i> created under the <i>BPEL Process</i> .	
<b>OK</b>	Save the values entered in the dialog.	
<b>Cancel</b>	Discard the values entered in the dialog.	
<b>Help</b>	Display this Help topic.	
<b>General</b>	Open the UML Properties dialog.	<a href="#">Properties</a> <sup>[956]</sup>

**Notes**

- Messages are created when you create a Web Service Operation

### 9.3.5.2.3 Create BPEL 2.0 Web Service Operation

BPEL is an **Orchestration** language, which orchestrates services that are exposed using **WSDL 1.1**. It coordinates the execution of the various operations of these involved services. BPEL 2.0 supports **WSDL 1.1 one-way** and **request-response** operations only (and not **solicit-response** and **notification** operations). Enterprise Architect enables you to create, for *BPEL Process* and *Pool* elements, **Web Service Operations** that support either *synchronous* (request-response) or *asynchronous* (one-way) interactions.

**Access** Right-click on a **BPEL Process or Pool** | **BPEL** | **Create WebService**

#### Reference

Field/Button	Usage	See also						
Operation	<p>Specify whether to create a new Web Service Operation or create one from an existing WSDL created / imported into Enterprise Architect ( using the Import WSDL dialog ) by selecting either:</p> <ul style="list-style-type: none"><li>• <b>Create New</b> or</li><li>• <b>Create from existing WSDL PortType Operation</b></li></ul>	<a href="#">Model WSDL</a> <sup>[2426]</sup> <a href="#">Import WSDL</a> <sup>[2447]</sup>						
Type	<p>Specify whether to create a <i>one-way</i> or <i>request-response</i> Operation by selecting either :</p> <ul style="list-style-type: none"><li>• <b>Asynchronous</b> or</li><li>• <b>Synchronous</b></li></ul>							
Partnership Details	<p>The interaction between the Web Service and the <i>BPEL Process</i> is modeled as a BPEL <i>partnerLink</i>. For this interaction, provide:</p> <table><tr><th>Field</th><th>Usage</th></tr><tr><td>Name</td><td>Name of the BPEL <i>partnerLink</i>.</td></tr><tr><td>Role</td><td><p>Name of either:</p><ul style="list-style-type: none"><li>• <i>myRole</i> (if this operation belongs to <i>BPEL Process</i> Web Service) or</li><li>• <i>partnerRole</i> (if this operation belong Web Service of a <i>Pool</i>)</li></ul><p>of the BPEL <i>partnerLink</i>.</p></td></tr></table>	Field	Usage	Name	Name of the BPEL <i>partnerLink</i> .	Role	<p>Name of either:</p> <ul style="list-style-type: none"><li>• <i>myRole</i> (if this operation belongs to <i>BPEL Process</i> Web Service) or</li><li>• <i>partnerRole</i> (if this operation belong Web Service of a <i>Pool</i>)</li></ul> <p>of the BPEL <i>partnerLink</i>.</p>	
Field	Usage							
Name	Name of the BPEL <i>partnerLink</i> .							
Role	<p>Name of either:</p> <ul style="list-style-type: none"><li>• <i>myRole</i> (if this operation belongs to <i>BPEL Process</i> Web Service) or</li><li>• <i>partnerRole</i> (if this operation belong Web Service of a <i>Pool</i>)</li></ul> <p>of the BPEL <i>partnerLink</i>.</p>							
Web Service	<table><tr><th>Field</th><th>Usage</th></tr><tr><td colspan="2">When <b>Create New</b> is selected in the <b>Operation</b> field:</td></tr><tr><td>PortType</td><td>Name of the <b>WSDL 1.1</b> <i>PortType</i> ( Interface</td></tr></table>	Field	Usage	When <b>Create New</b> is selected in the <b>Operation</b> field:		PortType	Name of the <b>WSDL 1.1</b> <i>PortType</i> ( Interface	
Field	Usage							
When <b>Create New</b> is selected in the <b>Operation</b> field:								
PortType	Name of the <b>WSDL 1.1</b> <i>PortType</i> ( Interface							

	<table><tr><td></td><td></td></tr><tr><td><b>Operation</b></td><td>Name of the <b>WSDL 1.1 PortType Operation</b>.</td></tr><tr><td colspan="2"></td></tr><tr><td colspan="2">When <b>Create from existing WSDL PortType Operation</b> is selected the <b>Operation</b> field:</td></tr><tr><td><b>WSDL Package</b></td><td>Select an existing WSDL package created / imported into Enterprise Architect.</td></tr><tr><td><b>PortType</b></td><td>Select a <b>WSDL 1.1 PortType</b> from the selected <b>WSDL Package</b>.</td></tr><tr><td><b>Operation</b></td><td>Select a <b>WSDL 1.1 PortType Operation</b> in the selected <b>PortType</b>.</td></tr><tr><td colspan="2"></td></tr></table>			<b>Operation</b>	Name of the <b>WSDL 1.1 PortType Operation</b> .			When <b>Create from existing WSDL PortType Operation</b> is selected the <b>Operation</b> field:		<b>WSDL Package</b>	Select an existing WSDL package created / imported into Enterprise Architect.	<b>PortType</b>	Select a <b>WSDL 1.1 PortType</b> from the selected <b>WSDL Package</b> .	<b>Operation</b>	Select a <b>WSDL 1.1 PortType Operation</b> in the selected <b>PortType</b> .			
<b>Operation</b>	Name of the <b>WSDL 1.1 PortType Operation</b> .																	
When <b>Create from existing WSDL PortType Operation</b> is selected the <b>Operation</b> field:																		
<b>WSDL Package</b>	Select an existing WSDL package created / imported into Enterprise Architect.																	
<b>PortType</b>	Select a <b>WSDL 1.1 PortType</b> from the selected <b>WSDL Package</b> .																	
<b>Operation</b>	Select a <b>WSDL 1.1 PortType Operation</b> in the selected <b>PortType</b> .																	
<b>Input</b>	<table><tr><td>Field</td><td>Usage</td></tr><tr><td colspan="2">When <b>Create New</b> is selected in the <b>Operation</b> field:</td></tr><tr><td><b>Message Name</b></td><td>Name of the <b>WSDL 1.1 Message</b>.</td></tr><tr><td><b>Properties</b></td><td>Click on the ( ... ) button to enter the <b>WSDL 1.1 Message Part Name</b> and <b>XSD Type</b>.</td></tr><tr><td colspan="2"></td></tr><tr><td colspan="2">When <b>Create from existing WSDL PortType Operation</b> is selected the <b>Operation</b> field:<ul style="list-style-type: none"><li>The fields in this tab are pre-filled with the details of the input <b>WSDL 1.1 Message</b> (of the <b>WSDL 1.1 PortType Operation</b> selected in the <b>Operation</b> field in the Web Service tab)</li></ul></td></tr><tr><td colspan="2"></td></tr></table>	Field	Usage	When <b>Create New</b> is selected in the <b>Operation</b> field:		<b>Message Name</b>	Name of the <b>WSDL 1.1 Message</b> .	<b>Properties</b>	Click on the ( ... ) button to enter the <b>WSDL 1.1 Message Part Name</b> and <b>XSD Type</b> .			When <b>Create from existing WSDL PortType Operation</b> is selected the <b>Operation</b> field: <ul style="list-style-type: none"><li>The fields in this tab are pre-filled with the details of the input <b>WSDL 1.1 Message</b> (of the <b>WSDL 1.1 PortType Operation</b> selected in the <b>Operation</b> field in the Web Service tab)</li></ul>						
Field	Usage																	
When <b>Create New</b> is selected in the <b>Operation</b> field:																		
<b>Message Name</b>	Name of the <b>WSDL 1.1 Message</b> .																	
<b>Properties</b>	Click on the ( ... ) button to enter the <b>WSDL 1.1 Message Part Name</b> and <b>XSD Type</b> .																	
When <b>Create from existing WSDL PortType Operation</b> is selected the <b>Operation</b> field: <ul style="list-style-type: none"><li>The fields in this tab are pre-filled with the details of the input <b>WSDL 1.1 Message</b> (of the <b>WSDL 1.1 PortType Operation</b> selected in the <b>Operation</b> field in the Web Service tab)</li></ul>																		
<b>Output</b>	<table><tr><td>Field</td><td>Usage</td></tr><tr><td colspan="2">When <b>Create New</b> is selected in the <b>Operation</b> field:</td></tr><tr><td><b>Message Name</b></td><td>Name of the <b>WSDL 1.1 Message</b>.</td></tr><tr><td><b>Properties</b></td><td>Click on the ( ... ) button to enter the <b>WSDL 1.1 Message Part Name</b> and <b>XSD Type</b>.</td></tr><tr><td colspan="2"></td></tr></table>	Field	Usage	When <b>Create New</b> is selected in the <b>Operation</b> field:		<b>Message Name</b>	Name of the <b>WSDL 1.1 Message</b> .	<b>Properties</b>	Click on the ( ... ) button to enter the <b>WSDL 1.1 Message Part Name</b> and <b>XSD Type</b> .									
Field	Usage																	
When <b>Create New</b> is selected in the <b>Operation</b> field:																		
<b>Message Name</b>	Name of the <b>WSDL 1.1 Message</b> .																	
<b>Properties</b>	Click on the ( ... ) button to enter the <b>WSDL 1.1 Message Part Name</b> and <b>XSD Type</b> .																	

	<p>When <b>Create from existing WSDL PortType Operation</b> is selected the <b>Operation</b> field:</p> <ul style="list-style-type: none"> <li>The fields in this tab are pre-filled with the details of the ou <b>WSDL 1.1 Message</b> (of the WSDL 1.1 <i>PortType Operation</i> selected in the <b>Operation</b> field in the Web Service tab)</li> </ul>	
<b>OK</b>	<p>Create a Web Service Operation based on the values entered in the dialog.</p> <p>The operation is created in a package that has the same name as the <i>BPEL Process / Pool</i> from which this dialog is invoked, under the <i>SupportingElements</i> package.</p> <p>A Package will be created under the <i>SupportingElements</i> package, with the same name as that of the <i>Pool</i>. This Package will act as a container for all the Web Service Operations created for that <i>Pool</i>.</p>	<a href="#">SupportingElements</a> <sup>1897</sup>
<b>Cancel</b>	Discard the values entered in the dialog and abort creating the Web Service Operation.	
<b>Help</b>	Display this Help topic.	

#### Notes

- The Output tab is not applicable for an Asynchronous operation

#### 9.3.5.2.4 Generate BPEL 2.0

BPEL 2.0 code can be generated from a *BPEL Process*. Enterprise Architect validates the *BPEL Process* before generating the BPEL 2.0 code. In addition to generating the BPEL 2.0 code, WSDL 1.1 files are generated for the *BPEL Process* and all the involved *Pools* (provided Web Service Operations are defined for them).

**Access** Right-click on BPEL Process | BPEL | Generate BPEL 2.0

#### Reference

Field/Button	Usage	See also
<b>File Name</b>	Specify the path where the BPEL 2.0 file is to be generated.	
<b>Namespace Details</b>	Double-click on an entry ( if any ) in this field to open the Namespace Details dialog and add / edit the namespace details.  The entry <b>DefaultPool</b> represents the current <i>BPEL Process</i> .	
<b>Generate BPEL</b>	Validate the model and generate BPEL 2.0.	<a href="#">BPEL Model Validation</a> <sup>[1917]</sup>
<b>Close</b>	Close this dialog.	
<b>Help</b>	Display this Help topic.	
<b>View BPEL</b>	View the generated BPEL 2.0 file.	

#### Learn more

- [Model a BPEL 2.0 Process](#)<sup>[1898]</sup>
- [Create a Web Service Operation](#)<sup>[1914]</sup>

#### Learning Center topics

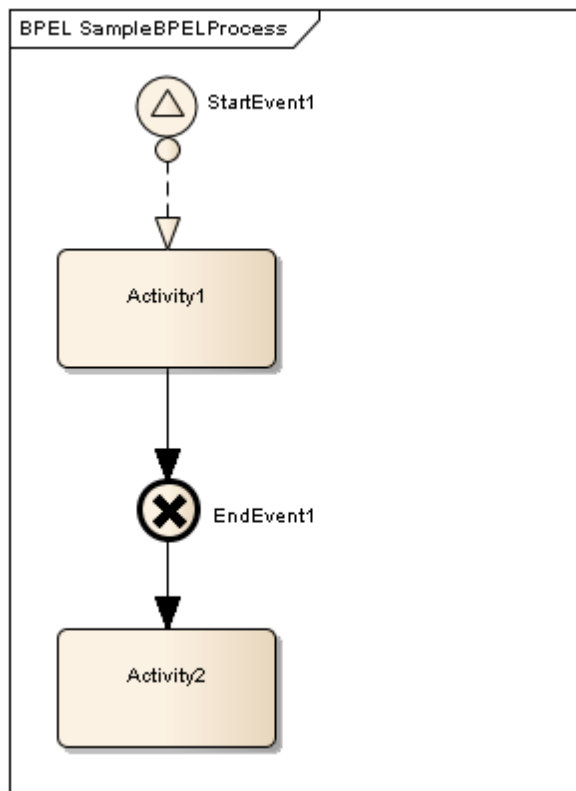
- (Alt+F1) | **Enterprise Architect** | **Business Modeling** | **BPEL** | **Generate BPEL**

### 9.3.5.3 BPEL Model Validation

The BPEL model is validated for both *syntax* and *semantic* errors before generating the BPEL code. Upon successful validation, the BPEL file (and the associated WSDL 1.1 files, if any) are generated. The results of the validation, as well as the progress of the BPEL generation, are displayed in the BPEL Progress tab of the Output window.

#### Example BPEL Model Violation:

The following model shows several basic BPEL violations:



- *StartEvent1* is of type **Signal**, which cannot be mapped to BPEL
- The *Message Flow* connector between *StartEvent1* and *Activity1* is invalid, as *Message Flow* cannot be used to connect *Activities* or *Events* within a *Process*
- *EndEvent1* is of type **Cancel**, which cannot be mapped to BPEL
- *EndEvent1* cannot have any outgoing *Sequence Flow* connector, as it represents the end of a *Process*
- *Activity2* is not valid, as only an *End Event* represents the end of a *Process*

### Notes

- Double-click on a validation error entry in the BPEL Progress tab of the Output window to go to the source of the error in the Project Browser

### Learn more

- [Generate BPEL 1.1](#) <sup>1893</sup>
- [BPEL 1.1 Modeling Restrictions](#) <sup>1871</sup>
- [Generate BPEL 2.0](#) <sup>1916</sup>
- [BPEL 2.0 Modeling Restrictions](#) <sup>1894</sup>
- [System Output Window](#) <sup>169</sup>

### Learning Center topics

- (Alt+F1) | [Enterprise Architect](#) | [Business Modeling](#) | [Business Rules](#) | [Validate Business Rules](#)

### 9.3.6 SPEM

The **MDG Technology for SPEM 2.0** is one of the business modeling tools integrated with Enterprise Architect.

"The **Software and Systems Process Engineering Meta-model (SPEM)** is a process engineering meta-model as well as conceptual framework, which can provide the necessary concepts for modeling, documenting, presenting, managing, interchanging, and enacting development methods and processes. An implementation of this meta-model would be targeted at process engineers, project leads, project and program managers who are responsible for maintaining and implementing processes for their development organizations or individual projects."

(Quoted from the *Object Management Group (OMG) Software & Systems Process Engineering Meta-Model Specification (Version 2.0, April 01 2008)*)

SPEM is a Profile of UML, which uses UML as a notation and takes an object-oriented approach. To accommodate UML 2, the SPEM specification was upgraded to **2.0** in April 2008. SPEM 2.0 focuses on providing the additional information structures that you require for processes modeled with UML 2 Activities or BPMN/BPDM, to describe an actual development process.

[Access](#) [Diagram](#) | [Diagram Toolbox: More Tools](#) | [SPEM](#)

#### SPEM Integration

Topic	Detail	See also
<b>SPEM in Enterprise Architect</b>	<p>Developing SPEM diagrams is quick and simple, using the MDG Technology for SPEM 2.0. The SPEM facilities are provided in the form of:</p> <ul style="list-style-type: none"> <li>• A SPEM diagram type, accessed through the New Diagram dialog</li> <li>• A set of SPEM pages in the Diagram Toolbox, providing SPEM elements (stereotyped UML elements)</li> <li>• SPEM element and relationship entries in the Toolbox Shortcut Menu and Quick Linker</li> </ul>	<p><a href="#">Add New Diagram</a> <sup>[822]</sup></p> <p><a href="#">SPEM Toolbox Pages</a> <sup>[1920]</sup></p> <p><a href="#">Example SPEM Diagram</a> <sup>[1924]</sup></p> <p><a href="#">Toolbox Shortcut Menu</a> <sup>[799]</sup></p> <p><a href="#">Quick Linker</a> <sup>[896]</sup></p>
<b>Disable SPEM</b>	<p>If you prefer not to use the MDG Technology for SPEM 2.0, you can disable it (and subsequently re-enable it) using the <b>MDG Technologies</b> dialog (<a href="#">Settings</a>   <a href="#">MDG Technologies</a>).</p>	<p><a href="#">MDG Technologies</a> <sup>[1477]</sup></p>

#### Learn more

- <http://www.omg.org/spec/SPEM/2.0/PDF> (Online Resource)

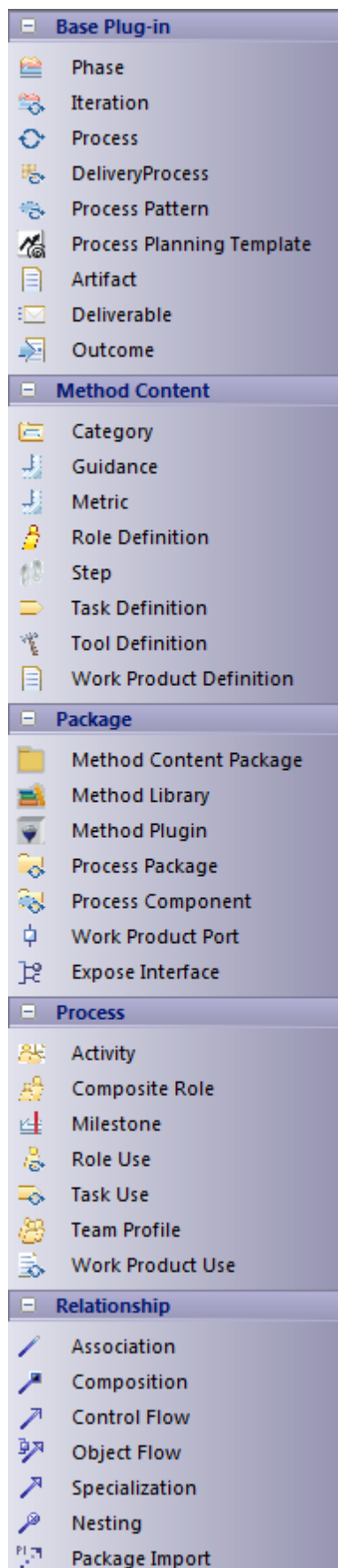
### 9.3.6.1 *SPEM Toolbox Pages*

The SPEM pages of the Diagram Toolbox provide a wide range of specialized elements and connectors for modeling development processes on SPEM diagrams. To create elements and relationships on a SPEM diagram, you can drag the appropriate icons from the Toolbox pages.

Access   **Diagram | Diagram Toolbox: More Tools | SPEM**

Toolbox Pages





Each element and connector provided in the SPEM Toolbox pages is described in the table below.

Page	Object	Action
<b>Base Plug-in</b>	<b>Phase</b>	Create a predefined special Activity representing a significant period in a project.
	<b>Iteration</b>	Group a set of nested Activities that are repeated more than once. Typically, Iteration is an Activity for which the default value of the <i>isRepeatable</i> attribute is <b>True</b> .
	<b>Process</b>	Represent a special Activity that describes a structure for particular types of development projects, or parts of them.
	<b>Delivery Process</b>	Represent a special Process describing a complete and integrated approach for implementing a specific project type.
	<b>Process Pattern</b>	Represent a special Process to describe a reusable cluster of Activities in a general process area that provides a consistent development approach to common problems.
	<b>Process Planning Template</b>	Represent a special Process that is prepared for instantiation by a project planning tool.
	<b>Artifact</b>	Represent a Work Product Definition that provides a description and definition for tangible work product types.
	<b>Deliverable</b>	Represent a Work Product Definition that provides a description and definition for packaging other Work Products, and that can be delivered to an internal or external party.
	<b>Outcome</b>	Represent a Work Product Definition that provides a description and definition for non-tangible work products.
<b>Method Content</b>	<b>Category</b>	Categorize content based on the user's criteria.
	<b>Guidance</b>	Identify reference items such as Guidelines, Templates, Checklists, Tool Mentors, Estimates, Supporting Materials, Reports and Concepts.
	<b>Metric</b>	Define a standard measurement for instances of Method Content elements.

Page	Object	Action
	<b>Role Definition</b>	Define a set of related skills, competencies, and responsibilities.
	<b>Step</b>	Represent parts or subunits of a Task Definition.
	<b>Task Definition</b>	Describe an assignable unit of work. Every Task Definition is assigned to specific Role Definitions. A Task is associated with input and output Work Products.
	<b>Tool Definition</b>	Describe the tools that are recommended or necessary for completing a specific Task.
	<b>Work Product Definition</b>	Define any forms of document, report or outcome that are consumed, produced or modified by Tasks.
<b>Package</b>	<b>Method Content Package</b>	Create a physical container to organize the Method Content elements.
	<b>Method Library</b>	Create an overall physical container for all SPEM 2.0 elements.
	<b>Method Plugin</b>	Create a physical container for Method Content Package and Process Packages. It can be used stand-alone as well as extended to many other Method Plugins.
	<b>Process Package</b>	Create a physical container that contains different kinds of Process element.
	<b>Process Component</b>	Offers the choice of creating a Process Component <b>package</b> - a special type of Process Package that provides the mechanism of encapsulation - or a Process Component <b>element</b> .
	<b>Work Product Port</b>	Defines the Work Products input and output for a Process Component. A Work Product Port identifies one type of Work Product for one Process Component, and defines whether the Work Product type is to be required (input) or supplied (output) by the Process Component, and whether this input or output is optional or not.
	<b>Expose Interface</b>	Represents a required or supplied interface to identify the fact that the element provides or requires an interface. A small dialog displays on which you type or select the name of the interface and whether it is provided or required.

Page	Object	Action
<b>Process</b>	<b>Activity</b>	Define basic units of work within a Process as well as the Process itself.
	<b>Composite Role</b>	Represent an aggregation of Role Definition references for an Activity.
	<b>Milestone</b>	Represent any significant events in a development project.
	<b>Process</b>	Create a special Activity that describes a structure for particular types of development project.
	<b>Role Use</b>	Represent a Role Definition in the context of one specific Activity.
	<b>Task Use</b>	Represent a Task Definition in the context of one specific Activity.
	<b>Team Profile</b>	Define a nested hierarchy of teams and team members.
	<b>Work Product Use</b>	Represent a Work Product Definition in the context of one specific Activity.

#### Learn more

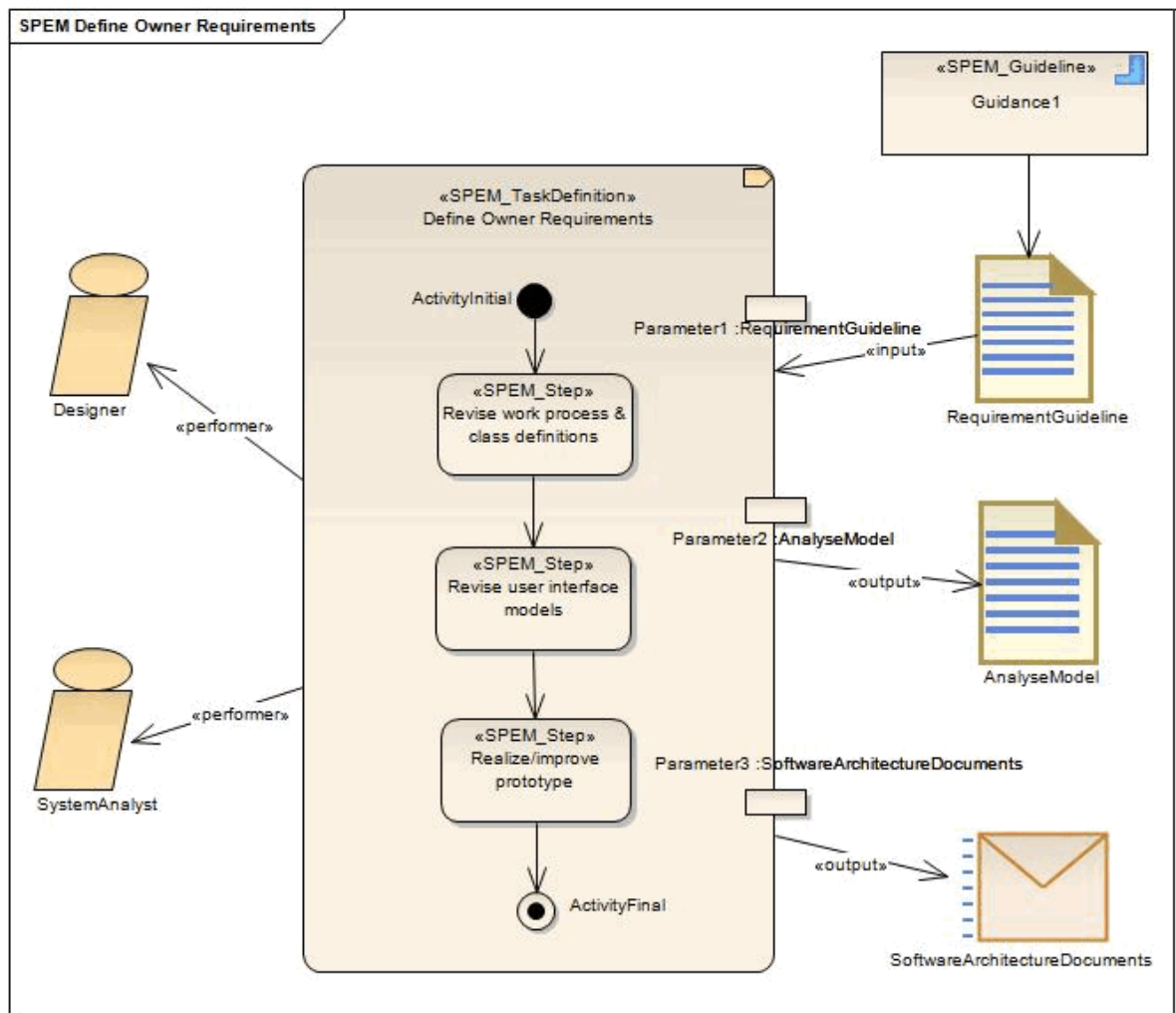
- [SPEM](#)<sup>[1919]</sup>
- [Example SPEM Diagram](#)<sup>[1924]</sup>

### 9.3.6.2 Example SPEM Diagram

You can model the development processes underpinning software and business process modeling using SPEM diagrams, and the wide range of specialized elements and connectors provided in the SPEM Toolbox.

Access **Project Browser package context menu | Add Diagram | Type = SPEM, Diagram Types = SPEM Diagram**

#### Example Diagram



### SPEM Element appearance

Every SPEM stereotype can be presented in one of two ways:

- Iconic presentation, where the shape is the object's icon (as shown for *Designer*, which has the Role Definition icon shape), or
- Textual presentation, where the shape is a rectangular outline with the object's icon as a decoration in the top right corner (as shown for *SPEM\_TaskDefinition*)

To set the presentation format for an element, use the *iconstyle* Tagged Value. To use:

- Iconic presentation, set the *iconstyle* Tagged Value to **True**
- Textual presentation, set the Tagged Value to **False**

### Learn more

- [SPEM](#)<sup>[1919]</sup>
- [SPEM Toolbox Pages](#)<sup>[1920]</sup>

### 9.3.7 ArchiMate

The **MDG Technology for ArchiMate 2.0** is one of the business modeling tools integrated with Enterprise Architect.

ArchiMate an open-standard enterprise architecture language from The Open Group, based on the IEEE 1471 standard. It offers a common language for describing the construction and operation of business processes, organizational structures, information flows, IT systems and technical infrastructure, which Enterprise Architects can use to describe, analyse and visualize the relationships among business domains clearly.

**Access** [Diagram](#) | [Diagram Toolbox: More Tools](#) | [ArchiMate2](#) | [<diagram type>](#)

#### ArchiMate Integration

Topic	Detail	See also
<b>ArchiMate in Enterprise Architect</b>	<p>Developing ArchiMate diagrams is fast and simple using the ArchiMate MDG Technology. The ArchiMate facilities are provided in the form of:</p> <ul style="list-style-type: none"> <li>Several ArchiMate diagram types, accessed through the New Diagram dialog</li> <li>A comprehensive set of ArchiMate pages in the Toolbox, for each diagram type</li> <li>ArchiMate element and relationship entries in the Toolbox Shortcut Menu and Quick Linker</li> </ul>	<a href="#">New Diagram</a> [822] <a href="#">Toolbox Shortcut</a> [799] <a href="#">Quick Linker</a> [896]
<b>ArchiMate Toolbox Pages</b>	<p>For each of the Business, Application and Technology diagram types, the toolbox pages show separate categories of elements - Information, Behavior and Structure - which provides clear differentiation between elements in different layers; for example, Business Service, Application Service and Infrastructure Service.</p> <p>Similarly, there are separate connector pages - Structural, Dynamic, and Other - although the connectors have parallels with standard UML connectors.</p> <p>The ArchiMate integration provides two further diagram types:</p> <ul style="list-style-type: none"> <li>Motivation             <ul style="list-style-type: none"> <li>For modeling stakeholders, drivers for change, business goals, principles and requirements</li> <li>Contains additional elements - Stakeholder, Driver, Assessment, Goal, Requirement, Constraint and Principle</li> <li>Contains an additional relationship - Influence</li> </ul> </li> <li>Implementation and Migration             <ul style="list-style-type: none"> <li>Supports project portfolio management, gap analysis, and transition and migration planning</li> <li>Contains additional elements - Work Package, Deliverable, Plateau and Gap</li> </ul> </li> </ul> <p>You can modify the appearance of elements by toggling the <b>Advanced</b>  </p>	<a href="#">Rectangle Notation</a> [1355]

Topic	Detail	See also
	<p><b>Use Rectangle Notation</b> context menu option on the element in a diagram.</p> <ul style="list-style-type: none"> <li>For Artifact, Process, Function, Interaction, Service, Event, Actor, Role, Collaboration, Interface, Component, Node and Device elements: <ul style="list-style-type: none"> <li>Select <b>Use Rectangle Notation</b> to show a rectangle with a decoration in the top corner</li> <li>Deselect <b>Use Rectangle Notation</b> to show the iconic representation of the element</li> </ul> </li> </ul> <p>You can set an Interface element to <i>provided</i>, <i>required</i>, <i>symmetric</i> or <i>assembly</i>, through its <b>interfacekind</b> Tagged Value. If <b>Use Rectangle Notation</b> is unselected, the orientation of the interface shape is determined automatically by the edge to which an incoming connector is attached.</p> <p>Because there is a large range of ArchiMate elements, using the <b>Quick Linker</b> arrow is very helpful in guiding you in selecting appropriate source and target elements and relationship types to model your enterprise architecture.</p>	
<b>Disable ArchiMate</b>	<p>If you prefer not to use the MDG Technology for ArchiMate, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog (<b>Settings   MDG Technologies</b>).</p>	<a href="#">MDG Technologies</a> <small>[147]</small>

### Notes

- Enterprise Architect also supports ArchiMate 1.0; if you wish, you can migrate your ArchiMate 1.0 model to ArchiMate 2.0

### Learn more

- [Migrate Archimate 1.0 to Archimate 2.0](#) [1927]

### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Modeling Languages | Business | ArchiMate**

## 9.3.7.1 Migrate ArchiMate 1.0 to ArchiMate 2.0

If you have created a model for ArchiMate 1.0, you can migrate all or part of it to ArchiMate 2.0 using the Automation Interface function *Migrate()*. This function updates the Tagged Values and, if required, stereotypes to ArchiMate for all elements, attributes, connectors and diagrams under the selected package or element.

### Example

The following VB script calls the *Migrate()* function to migrate the package or element to ArchiMate 2.0:

```

Sub MigrateElement (sGUID, lngPackageID)
    Dim proj as EA.Project
    set proj = Repository.GetProjectInterface
    proj.Migrate sGUID, "ArchiMate", "ArchiMate2"

    'refresh the model
    If lngPackageID <> 0 Then
        Repository.RefreshModelView (lngPackageID)
    End If
End Sub

Sub MigrateSelectedItem
    Dim selType
    Dim selElement as EA.Element
    Dim selPackage as EA.Package
    selType = GetTreeSelectedItemType
    If selType = 4 Then 'means Element
        set selElement = GetTreeSelectedObject
        MigrateElement selElement.ElementGUID, selElement.PackageID
        MsgBox "Element Migration Completed", 0, "ArchiMate2 Migration"
    ElseIf selType = 5 Then 'means Package
        set selPackage = GetTreeSelectedObject
        MigrateElement selPackage.PackageGUID, selPackage.PackageID
        MsgBox "Package Migration Completed", 0, "ArchiMate2 Migration"
    Else
        MsgBox "Select a Package or Element in the Project Browser to
        initiate migration", 0, "ArchiMate2 Migration"
    End If
End Sub

Sub Main
    MigrateSelectedItem
End Sub

Main

```

### Notes

- All diagrams are converted to Business Layer diagrams
- Collaboration elements are converted to Business Collaboration or Application Collaboration depending on the value of the **collaborationtype** Tagged Value
- Object elements are converted to Business Object, Contract or Data Object depending on the value of the **objecttype** Tagged Value
- Interface elements are converted to Business Interface, Application Interface or Infrastructure Interface depending on the value of the **interfacetype** Tagged Value
- Function elements are converted to Business Function or Application Function depending on the value of the **functiontype** Tagged Value
- Interaction elements are converted to Business Interaction or Application Interaction depending on the value of the **interactiontype** Tagged Value
- Service elements are converted to Business Service, Application Service or Infrastructure Service depending on the value of the **servicetype** Tagged Value
- The **iconstyle** Tagged Value is removed and the **Use Rectangle Notation** menu option is set on or off as appropriate
- Process elements become Business Process elements
- Event elements become Business Event elements
- Actor elements become Business Actor elements



- Role elements become Business Role elements
- Component elements become Application Component elements
- Software elements become System Software elements
- Specialisation connectors become Specialization connectors
- Realisation connectors become Realization connectors
- Network Connector connectors become Network
- Node elements now extend UML Class elements
- Software elements now extend UML Class elements
- Junction elements now extend UML Decision elements

#### Learn more

- [Migrate\(\)](#)<sup>[2978]</sup>

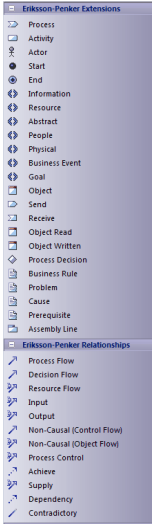
### 9.3.8 Eriksson-Penker Extensions

*Eriksson-Penker* extensions (developed by H. E. Eriksson and M. Penker) provide a framework for UML business processing model extensions, to which an enterprise architect can add stereotypes and properties appropriate to their business.

**Access**   [Diagram](#) | [Diagram Toolbox: More Tools](#) | [Eriksson-Penker Extensions](#)

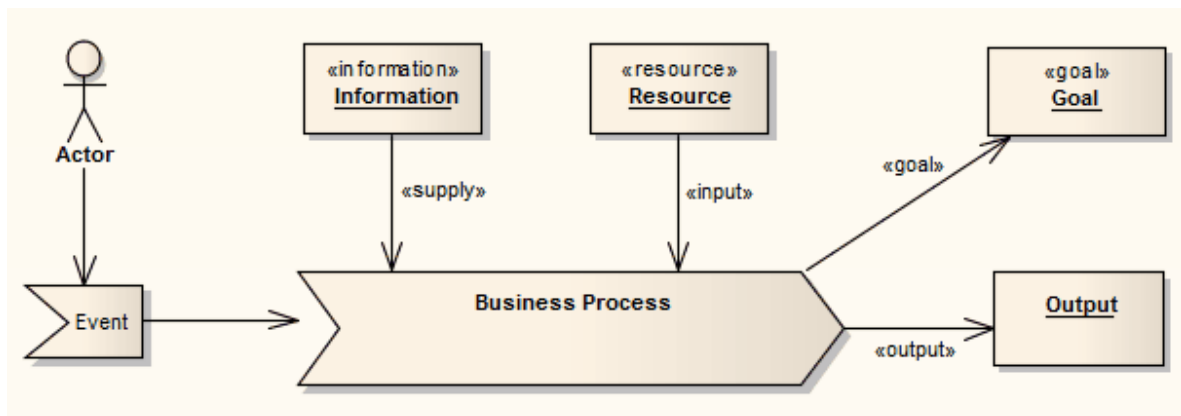
#### Topics

Images	Topic	Detail	See also
	<b>Eriksson-Penker in Enterprise Architect</b>	<p>Enterprise Architect provides - through the integration of MDG Technologies with the installer - two well-respected and proven UML extensions that further enhance the capture of business activities, processes, objects and information flows.</p> <p>One of these is <b>Business Process Modeling Notation</b> (BPMN).</p> <p>The other is the Eriksson-Penker profile which, through a set of stereotypes, provides a unique and powerful means of visualizing and communicating business processes and the necessary flow of information within an organization.</p> <p>The Eriksson-Penker extensions are provided in the form of:</p> <ul style="list-style-type: none"> <li>• An Eriksson-Penker diagram type, accessed through the New Diagram dialog</li> <li>• An Eriksson-Penker page in the Toolbox</li> <li>• Eriksson-Penker element and relationship entries in the Toolbox Shortcut Menu and Quick Linker</li> </ul>	<p><a href="#">BPMN Models</a><sup>[1845]</sup></p> <p><a href="#">New Diagram</a><sup>[822]</sup></p> <p><a href="#">Toolbox Shortcut</a><sup>[799]</sup></p> <p><a href="#">Quick Linker</a><sup>[896]</sup></p>

Images	Topic	Detail	See also
	<b>Eriksson-Penker Toolbox Page</b>	See the image to the left .	

### Example

The following is an example of a simple Eriksson-Penker diagram:



### Learn more

- [The Business Process Model](#) (Online Resource)

### Learning Center topics

- (Alt+F1) | **Enterprise Architect** | **Modeling Languages** | **Business** | **Eriksson Penker**

### 9.3.9 Decision Models

Enterprise Architect provides the **Decision Tables** facility, based on the OMG *Decision Model and Notation (DMN) specification* (2014). As stated in this document:

"The purpose of DMN is to provide the constructs that are needed to model decisions, so that organizational decision-making can be readily depicted in diagrams, accurately defined by business analysts, and (optionally) automated. ...

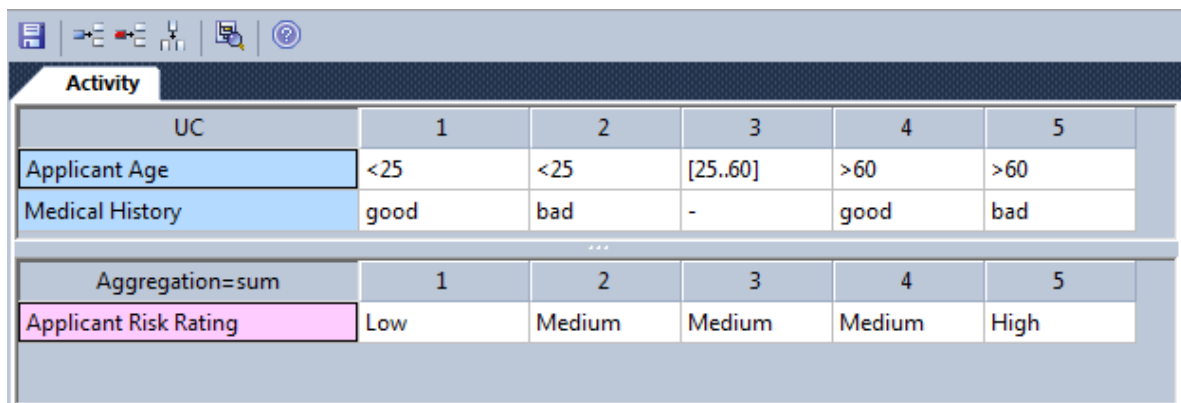
... DMN provides a common notation that is readily understandable by all business users, from the business analysts needing to create initial decision requirements and ... more detailed decision models, to the technical developers responsible for automating the decisions in processes, and... the business people who will manage and monitor those decisions. DMN creates a standardized bridge for the gap between the business decision design and decision implementation."

DMN is designed to be usable beside BPMN, and to support the exchange of decision models across organizations via XML.

The Decision Tables facility provides a simple method of modeling how a decision is made, and provides the capability of generating code for the decisions.

**Access** (Project Browser or Diagram) Click on Activity element | View | Decision Table or (Diagram) Right-click on Activity element | Decision Table

This example illustrates how you could model a decision on an insurance Risk Rating based on the applicant's age and medical history.



UC	1	2	3	4	5
Applicant Age	<25	<25	[25..60]	>60	>60
Medical History	good	bad	-	good	bad

...

Aggregation=sum	1	2	3	4	5
Applicant Risk Rating	Low	Medium	Medium	Medium	High

#### Learn more




- [Decision Table Editor](#)<sup>1931</sup>
- [Code Generation from Decision Models](#)<sup>1935</sup>



#### 9.3.9.1 Decision Table Editor




When you select to create a new Decision Table for an Activity element, or edit an existing one, the name of the Activity element displays in the tab under the Decision Table toolbar.

**Access** (Project Browser or Diagram) Click on Activity element | View | Decision Table (Diagram) Right-click on Activity element | Decision Table

Complete the Decision Table

Task	Action	See also
<b>Enter a condition</b>	<p>You can enter a condition into the table in a number of ways:</p> <ul style="list-style-type: none"> <li>• Type the text of the condition into the first available row, overtyping the <b>Add Condition</b> text</li> <li>• Drag onto the <b>Add Condition</b> cell the appropriate behavioral object from the Project Browser, such as a model element, Activity Parameter, Operation or Attribute, or</li> <li>• Click on the <b>Add Condition</b> cell and press <b>Ctrl+Spacebar</b>; <b>Intelli-Sense</b> is invoked to provide a list of possible objects that you can select from to complete the condition</li> </ul>	
<b>Add another condition</b>	<p>You can add further condition rows to the list in any of these ways:</p> <ul style="list-style-type: none"> <li>• Click on the  icon underneath the list of conditions</li> <li>• Right-click anywhere on the condition panel and select the <b>Add Condition</b> menu option, or</li> <li>• Click on the  (<b>Add Condition</b>) icon in the Decision Table toolbar</li> </ul> <p>Each condition that you add is bound to the previous conditions, so that they are evaluated in combination.</p>	
<b>List allowable values for the condition</b>	<p>(Optional) You can provide a comma-separated value (CSV) list of the values that you could assign to a condition. If you do this, those values are offered as suggestions to enter in the value fields along the same row.</p> <p>The <b>allowable values</b> column is hidden by default. To display it, click on the  icon in the Decision Table toolbar.</p> <p>In the allowable values cell against each condition, type in all the possible values separated by commas. These values can include operators such as &gt;, &lt;, = and ... .</p> <p>Attributes dragged from the Project Browser and classified by an Enumeration will complete the cell automatically and make the field Read Only.</p>	
<b>Set condition values</b>	<p>Each of the conditions you set have defined values that act as a decision point. For example, a person under the age of 18 years old cannot hire a car, so the values of <b>Yes</b> and <b>No</b> are decision points for the condition <b>18 or over</b>.</p> <p>In each of the columns 1, 2, 3 and so on, provide a condition value by either:</p> <ul style="list-style-type: none"> <li>• Typing the value into the cell (including operators such as &gt;, &lt;, = and ...) or</li> <li>• Right-clicking on the cell and selecting the <b>Allowable Values</b></li> </ul>	

Task	Action	See also
	<p>  <b>&lt;value&gt;</b> option (as set in the allowable values cell)</p>	
<b>Add further condition value columns</b>	<p>To add further value columns to the table, either:</p> <ul style="list-style-type: none"> <li>Right-click anywhere on the condition panel and select the <b>Add Column</b> menu option, or</li> <li>Click on the  (<b>Add Column</b>) icon in the Decision Table toolbar</li> </ul> <p>These options add the column to both the Condition and Conclusion panels.</p>	
<b>Set the Policy and Completeness</b>	<p>The Decision <b>Hit Policy</b> defines if and how the conditions are combined in making a decision. <b>Completeness</b> identifies whether or not the decision set is complete.</p> <p>To set these two flags, right-click on the <b>Condition</b> header (top left corner of the Condition panel) and select the options:</p> <ul style="list-style-type: none"> <li><b>Hit Policy   &lt;value&gt;</b> and</li> <li><b>Completeness   Complete</b> or <b>Incomplete</b></li> </ul> <p>If you do not want to display the completeness, select the <b>Completeness   Clear</b> option.</p>	
<b>Delete Column or Condition</b>	<p>To remove a condition that no longer applies, or a column of values that are no longer tested, right-click on the appropriate column or row and select the options:</p> <ul style="list-style-type: none"> <li><b>Delete Condition</b> or</li> <li><b>Delete Column</b></li> </ul> <p>A prompt displays to confirm the deletion. You cannot delete the title column or condition number row, nor can you delete the allowable values column.</p>	
<b>Enter a Conclusion</b>	<p>You can enter a conclusion into the table in a number of ways:</p> <ul style="list-style-type: none"> <li>Type the text of the conclusion into the first available row, overtyping the <b>Add Conclusion</b> text</li> <li>Drag onto the <b>Add Conclusion</b> cell the appropriate behavioral object from the Project Browser, such as a model element, Activity Parameter, Operation or Attribute</li> </ul>	
<b>Add Further Conclusions</b>	<p>You can add further conclusion rows to the list in any of these ways:</p> <ul style="list-style-type: none"> <li>Click on the  icon underneath the list of conclusions</li> <li>Right-click anywhere on the conclusion panel and select the <b>Add Conclusion</b> menu option, or</li> </ul>	

Task	Action	See also
	<ul style="list-style-type: none"> <li>Click on the  (<b>Add Conclusion</b>) icon in the Decision Table toolbar</li> </ul> <p>Each conclusion that you add is bound to the previous conclusions, so that the final decision takes them all into account.</p>	
<b>List allowable values for the conclusion</b>	<p>(Optional) You can provide a comma-separated value (CSV) list of the values that you could assign to a conclusion. If you do this, those values are offered as suggestions to enter in the value fields along the same row.</p> <p>The <b>allowable values</b> column is hidden by default. To display it, click on the  icon in the Decision Table toolbar.</p> <p>In the allowable values cell against each conclusion, type in all the possible values separated by commas. These values can include operators such as &gt;, &lt;, = and ... .</p> <p>Attributes dragged from the Project Browser and classified by an Enumeration will complete the cell automatically and make the field Read Only.</p>	
<b>Set Conclusion values</b>	<p>Each of the conclusions you set have values that define the outcome of the conditional values in the same column, when the conditions are met.</p> <p>In each of the columns 1, 2, 3 and so on, provide a conclusion value by either:</p> <ul style="list-style-type: none"> <li>Typing the value into the cell (including operators such as &gt;, &lt;, = and ...) or</li> <li>Right-clicking on the cell and selecting the <b>Allowable Values   &lt;value&gt;</b> option (as set in the allowable values cell)</li> </ul>	
<b>Add Further Conclusion Columns</b>	<p>To add further value columns to the table, either:</p> <ul style="list-style-type: none"> <li>Right-click anywhere on the conclusion panel and select the <b>Add Column</b> menu option, or</li> <li>Click on the  (<b>Add Column</b>) icon in the Decision Table toolbar</li> </ul> <p>These options add the column to both Condition and Conclusion panels.</p>	
<b>Set the Table Aggregation Values</b>	<p>The Table Aggregation value indicates how the conclusion values are to be combined to form the decision.</p> <p>Set the value by right-clicking on the <b>Conclusion</b> column heading and selecting the <b>Aggregation   &lt;value&gt;</b> option.</p> <p>If you do not want to set the Table Aggregation, select the <b>Aggregation   Clear</b> option.</p>	

Task	Action	See also
<b>Delete Column or Conclusion</b>	<p>To remove a conclusion that is no longer valid, or a column of result values that are no longer produced, right-click on the appropriate column or row and select the options:</p> <ul style="list-style-type: none"> <li>• <b>Delete Conclusion</b> or</li> <li>• <b>Delete Column</b></li> </ul> <p>A prompt displays to confirm the deletion. You cannot delete the title column or conclusion number row, nor can you delete the allowable values column.</p>	
<b>Check Object properties and location</b>	<p>If you have created a condition or conclusion by dragging an object from the Project Browser, you can right-click on its row and:</p> <ul style="list-style-type: none"> <li>• Display the Properties dialog for the object, by selecting the <b>Properties</b> menu option</li> <li>• Locate the object in the model by selecting the <b>Find in Project Browser</b> option</li> </ul>	
<b>Save your changes</b>	At regular intervals, and before you exit from the table, click on the <b>Save</b> icon in the Decision Table toolbar.	

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect** | **Modeling Basics** | **Decision Tables** | **Basic Decision Table**

### 9.3.9.2 Code Generation from Decision Models

It is possible to generate the method code of a Class from an Activity element, based on the logic defined in its Decision Table, by placing the Activity as a child of the Class element from which the code is generated.

#### Set up Class and generate code

Step	Action	See also
<b>1</b>	Create an Activity with a Decision Table.	<a href="#">Decision Table Editor</a> [1931]
<b>2</b>	Make the Activity element a child of a Class element.	<a href="#">Paste from Project Browser</a> [833]
<b>3</b>	Select the Class and press <b>F11</b> to open the Generate Code dialog, and generate the code.	<a href="#">Generate a Single Class</a> [2113]

Step	Action	See also
4	Press <b>F12</b> to view the generated source code.	

### Decision Table code templates

Code Generation from a Decision Table automatically applies these EASL code generation templates:

- **Behavior Body**
- **Decision Table**
- **Decision Logic**
- **Decision Condition**
- **Decision Action**

### Notes

- Currently **C++** is the only language implemented in the EASL templates for Decision Table code generation
- You can use `ActivityParameters` to define method parameters
- You can add `Attributes` to the `Activity` element to define local variables

### Learn more

- [EASL Code Generation Macros](#)<sup>[1688]</sup> (Behavioral Templates)
- [Activity Parameter Nodes](#)<sup>[1281]</sup>

### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Modeling Basics | Decision Tables | Decision Code Generation**



## 9.4 Data Models

Data Modelers and Information Architects are challenged with creating data models that span multiple levels of abstraction - from concept to physical implementation. They might also be responsible for maintaining traceability between these models. Enterprise Architect helps to meet these challenges with easy-to-use tools for building and maintaining all of the fundamental data models - Conceptual, Logical and Physical.

### Data Models

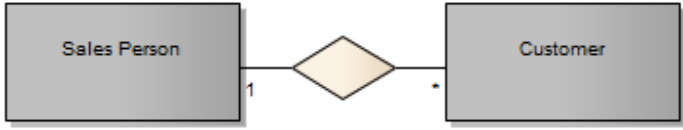

Type	Detail	See also
<b>Conceptual Data Models</b>	<p>Conceptual data models, also called Domain models, establish the basic concepts and semantics of a given domain and help to communicate these to a wide audience of stakeholders.</p> <p>Conceptual models also serve as a common vocabulary during the analysis stages of a project; they can be created in Enterprise Architect using Entity-Relationship or UML Class models.</p>	<a href="#">Conceptual Data Model</a> <sup>[1938]</sup>
<b>Logical Data Models</b>	<p>Logical data models add further detail to conceptual model elements and refine the structure of the domain; they can be defined using Entity-Relationship or UML Class models.</p> <p>One benefit of a Logical data model is that it provides a foundation on which to base the Physical model and subsequent database implementation.</p> <p>Entity-relationship modeling is an abstract and conceptual database modeling method, used to produce a schema or semantic data model of, for example, a relational database and its requirements, visualized in Entity-Relationship Diagrams (ERDs).</p> <p>ERDs assist you in building conceptual data models through to generating Data Definition Language (DDL) for the target DBMS.</p>	<a href="#">Logical Data Model</a> <sup>[1939]</sup>  <a href="#">Entity Relationship Diagrams (ERDs)</a> <sup>[1939]</sup>
<b>Logical to Physical Model</b>	A Logical model can be transformed to a Physical Data model using a DDL Transformation.	<a href="#">DDL Transformation</a> <sup>[2025]</sup>
<b>Physical Data Models</b>	<p>Physical data models in Enterprise Architect help you visualize your database structure and automatically derive the corresponding database schema; you use Enterprise Architect's UML Profile for Data Modeling specifically for this purpose.</p> <p>The profile provides useful extensions of the UML standard that map database concepts of tables and relationships onto the UML concepts of Classes and Associations; you can also model database keys, triggers, constraints, referential integrity and other relational database features.</p> <p>Because Enterprise Architect lets you visualize each type of data model in the same repository, you can easily manage dependencies between each level of abstraction; this helps you</p>	<a href="#">Database Engineering</a> <sup>[2334]</sup> <a href="#">Physical Data Model</a> <sup>[2335]</sup>

Type	Detail	See also
	maximize traceability and verify completeness of system implementation.	

### 9.4.1 Conceptual Data Model

A Conceptual data model is the most abstract form of data model. It is helpful for communicating ideas to a wide range of stakeholders because of its simplicity. Therefore platform-specific information, such as data types, is omitted from a Conceptual data model. Other implementation details, such as procedures and interface definitions, are also excluded. Below is an example of a Conceptual data model that is rendered using two of the notations supported by Enterprise Architect.

#### Topics

Topic	Detail	See also
<b>Example</b>	<p>Using Entity-Relationship (ER) notation, we represent the data concepts <i>Sales Person</i> and <i>Customer</i> as Entities with a 1-to-many relationship between them.</p>  <p>A Conceptual data model that uses Entity-Relationship notation</p> <p>We can represent the exactly the same semantic information using UML Classes and Associations.</p>  <p>An equivalent model using UML Class notation</p> <p>Whether you use UML or ER notation to represent data concepts in your project depends on the experience and preferences of the stakeholders involved.</p> <p>The detailed structure of the data concepts illustrated above is defined later by the Logical data model.</p>	<p><a href="#">Entity Relationship Diagrams (ERDs)</a> <sup>[1939]</sup></p> <p><a href="#">UML Class Modeling</a> <sup>[1184]</sup></p>

#### Learn more

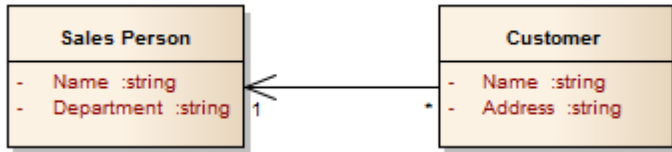
- [Logical data models](#) <sup>[1939]</sup>
- [Entity Relationship modeling](#) <sup>[1939]</sup>

- [UML Class modeling](#)<sup>[1184]</sup>

### 9.4.2 Logical Data Model

Logical data models help to define the detailed structure of the data elements in a system and the relationships between data elements. They refine the data elements introduced by a Conceptual data model and form the basis of the Physical data model. In Enterprise Architect, a Logical data model is typically represented using the UML Class notation.

#### Topics

Topic	Detail	See also
<b>Example</b>	<p>A simple example of a Logical data model is provided below.</p> <p>Note that the data elements <i>Sales Person</i> and <i>Customer</i> contain UML Attributes; the Attribute types, however, remain platform-independent.</p>  <pre> classDiagram     class SalesPerson {         - Name :string         - Department :string     }     class Customer {         - Name :string         - Address :string     }     SalesPerson "1" -- "*" Customer           </pre> <p>A Logical data model defined using UML Class notation</p> <p>Platform-specific Attribute types and other meta-data that relate to a specific DBMS implementation are defined by the Physical data model.</p>	<a href="#">UML Class Modeling</a> <sup>[1184]</sup>

#### Learn more

- [Physical data models](#)<sup>[2335]</sup>
- [Connect to Element Feature](#)<sup>[1110]</sup>

### 9.4.3 Entity Relationship Diagrams (ERDs)

According to the online [Wikipedia](#):

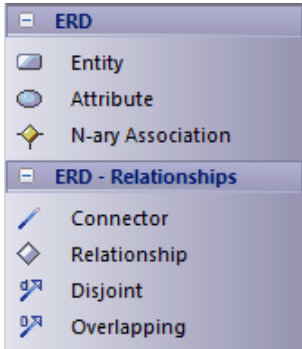
*An entity-relationship model (ERM) is an abstract and conceptual representation of data. Entity-relationship modeling is a database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion. Diagrams created by this process are called Entity-Relationship Diagrams, ER Diagrams, or ERDs.*

#### Entity Relationship Diagrams in Enterprise Architect

Entity Relationship Diagrams in Enterprise Architect are based on Chen's ERD building blocks: entities are represented as rectangles, attributes are represented as ellipses and relationships are represented as diamond-shape connectors.

ERD technology in Enterprise Architect assists you in every stage from building conceptual data models to generating Data Definition Language (DDL) for the target DBMS.

Topics

Topic	Detail	See also
<b>MDG Technology for ERD and ERD Transformations</b>	<p>Enterprise Architect enables you to develop Entity Relationship diagrams quickly and simply, through use of an MDG Technology integrated with the Enterprise Architect installer.</p> <p>The Entity Relationship diagram facilities are provided in the form of:</p> <ul style="list-style-type: none"> <li>• An Entity Relationship diagram type, accessed through the New Diagram dialog</li> <li>• An Entity Relationship Diagram page in the Toolbox</li> <li>• Entity Relationship element and relationship entries in the Toolbox Shortcut Menu and Quick Linker</li> </ul> <p>Enterprise Architect also provides transformation templates to transform Entity Relationship Diagrams into Data Modeling Diagrams, and vice versa.</p>	<a href="#">New Diagram Dialog</a> <sup>[822]</sup> <a href="#">Toolbox Shortcut</a> <sup>[799]</sup> <a href="#">Quick Linker</a> <sup>[896]</sup> <a href="#">ERD to Data Model Transformation</a> <sup>[2032]</sup> <a href="#">Data Model to ERD Transformation</a> <sup>[2024]</sup>
<b>Entity Relationship Diagram Toolbox Page</b> 	<p>You can access the Entity Relationship Diagram page of the Toolbox through the <b>More tools   Entity Relationship Diagrams</b> option.</p> <ul style="list-style-type: none"> <li>• <i>Entity</i> is an object or concept that is uniquely identifiable; the property of Multiplicity in the SourceRole and TargetRole definitions for the <i>Relationship</i> connector (below) can be used to define the cardinality of an Entity that participates in this relationship</li> <li>• <i>Attribute</i> is a property of an entity or a relationship type</li> <li>• <i>N-ary Association</i> represents unary (many-to-many recursive) or ternary relationships and can also be used to represent relationships that have attributes among the entities; the N-ary Association element should always be at the target end of a connector</li> <li>• <i>Connector</i> is a connector between an Entity and an Attribute, and between two Attributes</li> <li>• <i>Relationship</i> is a diamond-shape connector, representing the meaningful association among entities</li> <li>• <i>Disjoint</i> and <i>Overlapping</i> represent the relationships between the super-class Entity and the sub-class Entity</li> </ul>	

Topic	Detail	See also
<b>A typical Entity Relationship diagram</b>		
<b>Disable Entity Relationship Diagrams</b>	<p>If you prefer not to use Entity Relationship Diagrams in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog (<b>Settings   MDG Technologies</b>).</p>	<a href="#">MDG Technologies</a> <sup>1477</sup>

### Tagged Values

Some of the Entity Relationship diagram components can be modified by Tagged Values, as indicated below:

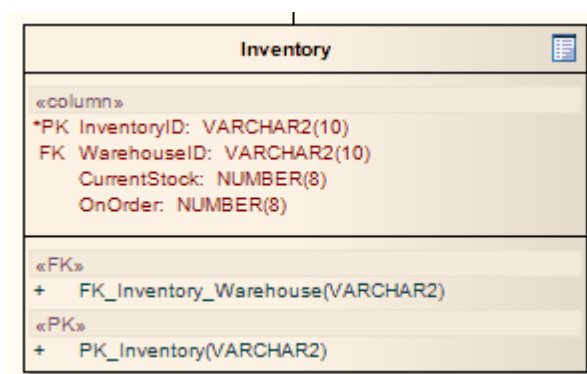
Component	Tagged Value	Notes
<b>Entity</b>	isWeakEntity	If <b>true</b> , this entity is a weak entity.
<b>Attribute</b>	attributeType	<p>Four options:</p> <ul style="list-style-type: none"> <li>• normal Attribute</li> <li>• primary key attribute</li> <li>• multi-valued Attribute</li> <li>• derived Attribute</li> </ul>
	commonDataType	Defines the common data type for each attribute.

Component	Tagged Value	Notes
	dbmsDataType	<p>Defines the customized DBMS data type for each attribute.</p> <p>You must define the customized type first through the <b>Settings   Database Datatypes</b> menu option.</p> <p>Also, set the <i>commonDataType</i> tag to <b>na</b> to activate the <i>dbmsDataType</i> tag.</p>
<b>N-ary Association</b>	isRecursive	<p>If <b>true</b>, the N-ary Association represents the many-to-many recursive relationship.</p> <p>For one-to-many and one-to-one recursive relationships, we suggest using the normal Relationship connector.</p> <p>Sometimes you might want to limit the stretch of the diamond-shape Relationship connectors; simply pick a Relationship connector, right-click to display the context menu, and select the <b>Bend Line at Cursor</b> option.</p>
<b>Relationship</b>	isWeak	If <b>true</b> , the Relationship is a weak relationship.
<b>Disjoin Overlapping</b>	Participation	Two options: <b>partial</b> and <b>total</b> .

#### Notes

- Entity Relationship Diagrams are supported in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect

### 9.4.4 Table



#### Description

A Table is a stereotyped Class. It is drawn with a small table icon in the upper right corner. You typically use this element in Data Modeling diagrams.

A Table element has a special Properties dialog, with settings for database type and the ability to set column information and data-related operations such as triggers and indexes. When setting up a Table, make sure you set the default database type for that Table, otherwise you do not have any data types to choose from when creating columns.

#### Toolbox icon

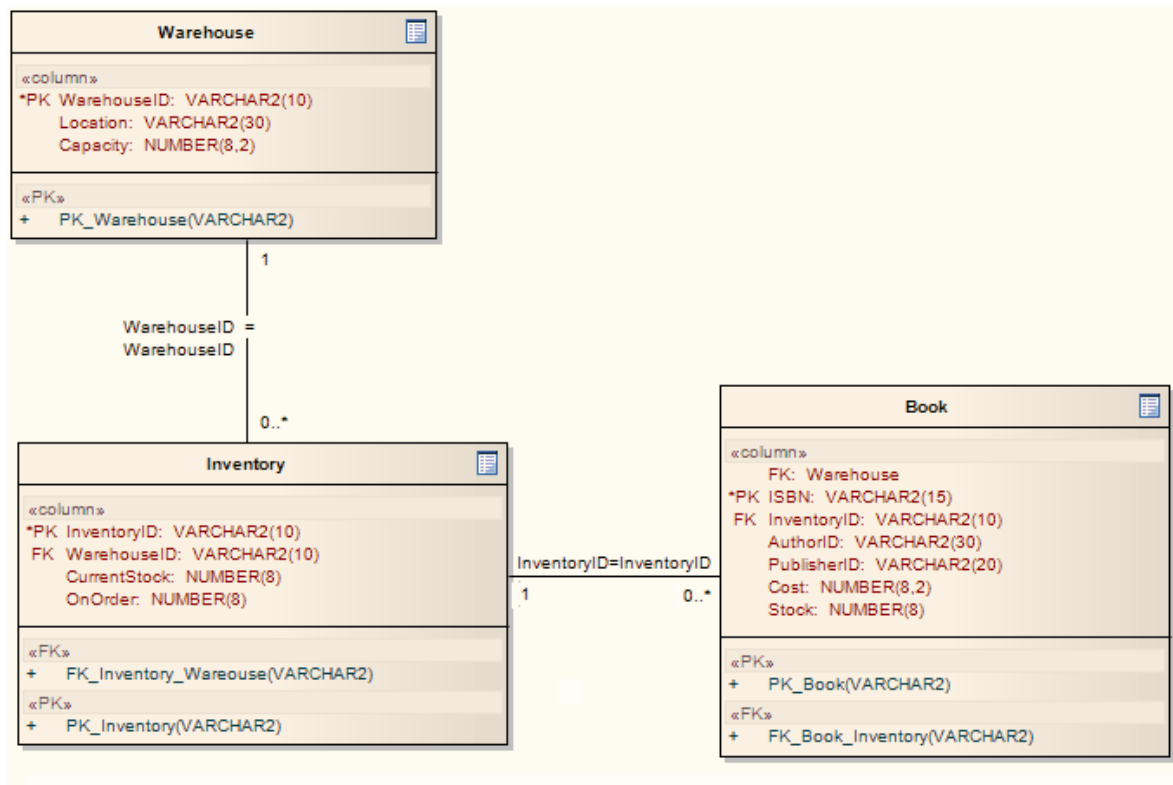


#### Learn more

- [Data Modeling Diagram](#)<sup>[1943]</sup>
- [Set the Default Database Type](#)<sup>[2340]</sup>

### 9.4.5 Database Schema

The following is an example Database Schema represented by a Physical Data Model.



#### Learn more

- [Physical Data Model](#)<sup>[2335]</sup>
- [Data Modeling Toolbox](#)<sup>[818]</sup>

## 9.5 Geodatabase Design for ArcGIS

The **ArcGIS** system, developed by **Esri**, supports the development and management of geodatabases. As for other databases, it is useful to **model the design of a geodatabase** using a standard notation such as UML. You can perform such modeling in Enterprise Architect, using the UML profile for ArcGIS, which is part of the built-in MDG Technology for ArcGIS. Once you have modeled an ArcGIS schema in Enterprise Architect, you can **export the model to ArcGIS** as an XML Workspace document. You can also visualize an existing ArcGIS geodatabase schema, by **importing the ArcGIS XML Workspace document** into Enterprise Architect.

### Notice of Acknowledgement:

*Support for modeling ArcGIS databases in Enterprise Architect was developed in collaboration with the Commonwealth Scientific and Industrial Research Organization (**CSIRO**), who defined mappings between UML 2 and ArcGIS concepts, and prototyped an automated import and export capability for ArcGIS geodatabase schemas represented in UML.*

### Notes

- The MDG Technology for ArcGIS is available in the Professional, Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect

### Learn more

- [ArcGIS UML Profile](#)<sup>[1944]</sup>
- [Export ArcGIS XML Workspace](#)<sup>[1961]</sup>
- [Import ArcGIS XML Workspace](#)<sup>[1968]</sup>
- [Validate an ArcGIS Workspace](#)<sup>[1970]</sup>

### 9.5.1 ArcGIS UML Profile

The MDG Technology for ArcGIS is built in to the Enterprise Architect installer. A key component of the technology is the UML Profile for ArcGIS.

### Access   **Extensions | ArcGIS**

### Features

Feature	Detail	See also
<b>Profile Support</b>	<p>The built-in MDG Technology for ArcGIS provides:</p> <ul style="list-style-type: none"> <li>• ArcGIS Toolbox pages that map ArcGIS concepts to appropriately stereotyped UML elements</li> <li>• A Model Pattern that helps you to start designing geodatabases quickly and to use the required package structure in Enterprise Architect</li> <li>• Datatypes that are specific to the ArcGIS platform</li> <li>• Quick-linker capabilities that help you make valid connections</li> </ul>	<p><a href="#">New Diagram</a><sup>[822]</sup></p> <p><a href="#">Toolbox Shortcut</a><sup>[799]</sup></p> <p><a href="#">Quick Linker</a><sup>[896]</sup></p>



Feature	Detail	See also
	between elements	
<b>ArcGIS Toolbox Pages</b>	<p>The ArcGIS Toolbox contains five core pages:</p> <ul style="list-style-type: none"> <li>• Domains - for coded value and range domains</li> <li>• Features and Tables - for custom feature types and tables</li> <li>• Network Features - for geometric network and topology packages</li> <li>• Raster - for raster datasets</li> <li>• Workspace - for ArcGIS workspace and spatial reference information</li> </ul> <p>Two additional Toolboxes group the objects used specifically in creating Geometric Network and Topology diagrams.</p>	<a href="#">ArcGIS Toolbox Pages</a> <sup>[1946]</sup> <a href="#">Diagram Toolbox</a> <sup>[792]</sup>
<b>Spatial References</b>	Enterprise Architect helps you to model Spatial Reference information for your ArcGIS schema, including the selection of predefined coordinate systems and associated values.	<a href="#">Setting ArcGIS Coordinate Systems</a> <sup>[1958]</sup>
<b>Show/Hide System Attribute Fields</b>	<p>The ArcGIS elements provided through the Toolbox pages contain a number of system-assigned attributes that define the «<i>AttributeIndex</i>», «<i>SpatialIndex</i>» and «<i>RequiredField</i>» stereotypes. When you drag an element onto a diagram from the Toolbox, these attributes are not visible in the newly-created structure.</p> <p>If you want to show these system attributes, right-click on the element (s) and select the <b>Extensions   ArcGIS   Show or Hide ArcGIS System Fields</b> menu option. Similarly, if you have exposed the attributes and want to hide them, select the elements and select the menu option again.</p> <p>This option does not operate on attributes or stereotypes <b>you</b> have added to the selected elements, nor on elements that you have not selected.</p> <p>If you do not select any elements, the option is grayed out.</p>	
<b>Disable MDG Technology for ArcGIS</b>	If you prefer not to use the MDG Technology for ArcGIS in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog ( <b>Settings   MDG Technologies</b> ).	<a href="#">MDG Technologies</a> <sup>[1477]</sup>

**Notes**

- The MDG Technology for ArcGIS is available in the Professional, Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect

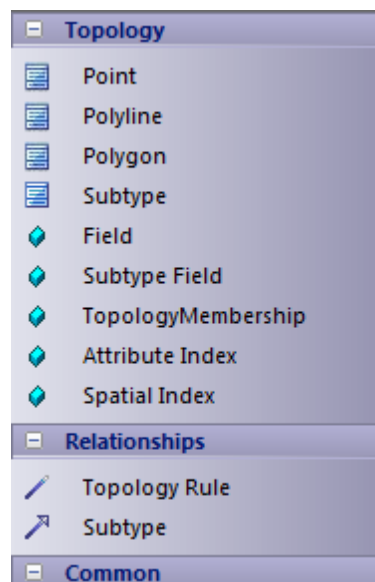
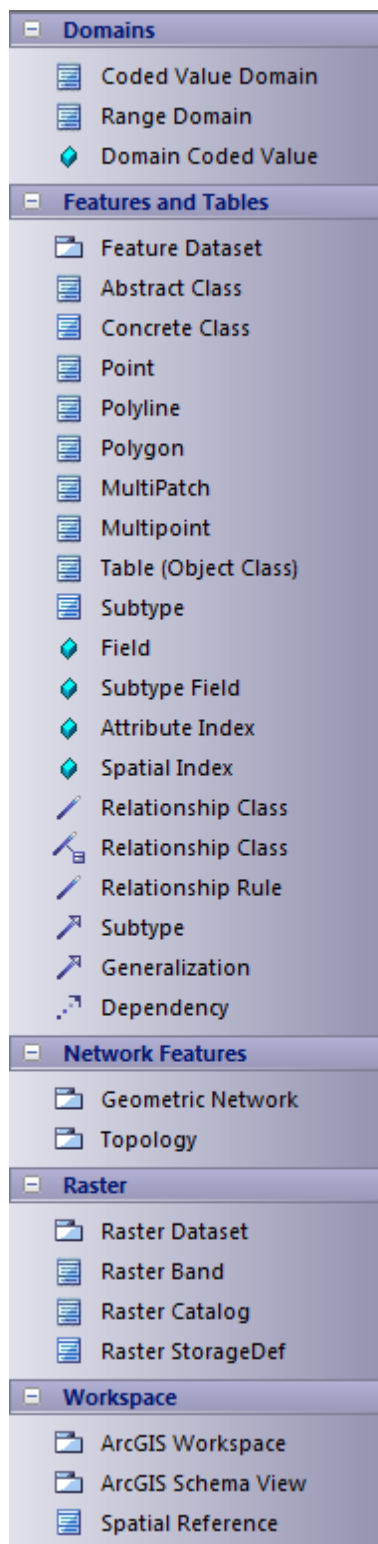
### 9.5.1.1 ArcGIS Toolbox Pages

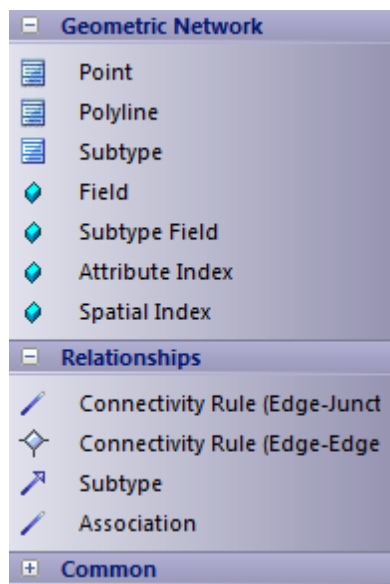
The ArcGIS Toolbox pages provide elements and connectors that you can use to model ArcGIS geodatabase concepts and relationships. The ArcGIS Toolbox consists of five core pages:

- Domains - for coded value and range domains
- Features and Tables - for custom feature types and tables
- Network Features - to identify geometric network and topology packages
- Raster - for raster datasets
- Workspace - for ArcGIS workspace and spatial reference information

Two additional Toolboxes group the objects used specifically in creating Geometric Network and Topology diagrams.

**Access**   [Diagram | Diagram Toolbox: More Tools | ArcGIS | Core](#)  
[Diagram | Diagram Toolbox: More Tools | ArcGIS | Geometric Network](#)  
[Diagram | Diagram Toolbox: More Tools | ArcGIS | Topology](#)





#### Diagram toolbox icons

Toolbox Icon	Description
<b>Packages</b>	
<b>ArcGIS Workspace</b>	The geodatabase workspace Package, which holds all the ArcGIS modeling elements. Export the contents of this Package to produce the Geodatabase XML Workspace Document, which can be imported to <b>Esri ArcCatalog</b> .
<b>ArcGIS Schema View</b>	<p>A stereotyped Package that represents a subset of the geodatabase schema defined within the ArcGIS Workspace Package. ArcGIS Schema View Packages are useful if you need to create partial or modular schemas based on your complete geodatabase design. You can create any number of ArcGIS Schema View Packages under your ArcGIS Workspace Package.</p> <p>Add this element to a diagram under your workspace, then create a UML Dependency connector from it to each Package to include in the generated XML Workspace Document. For example, you could include only a subset of your Feature Datasets and Domains, by drawing UML Dependency connectors to the appropriate Packages.</p> <p>To export your ArcGIS Schema View for use with ArcCatalog, right-click on it and select the <b>Extensions   ArcGIS   Export to ArcGIS Workspace XML</b> option. The system generates a Workspace XML document containing only the elements associated with the ArcGIS Schema View Package.</p> <p>See the <i>Export Modular ArcGIS Schemas</i> topic.</p>
<b>Feature Dataset</b>	<p>A stereotyped Package that holds or organizes Point, Polyline, Polygon or Multipatch elements with the same spatial reference, geometry type and attribute fields (that is, Feature Classes).</p> <p>The Feature Dataset is only created under the ArcGIS Workspace Package (above); it can not be created under another Feature Dataset Package. Feature Dataset</p>

Toolbox Icon	Description
	<p>Packages can contain other types of subpackages, however, which can be useful for organizing large Feature Datasets. When exported to an XML Workspace document, <b>elements</b> of any subPackages are included while the subPackages themselves are ignored, resulting in a 'flattened' model hierarchy.</p> <p>Although ArcGIS forbids Tables (ObjectClasses) being defined under Feature Datasets, Enterprise Architect lets you model Tables under Feature Datasets for convenience. On export, Tables are placed at the root level to create a valid schema.</p>
<b>Geometric Network</b>	An extended UML Package that represents the logical relationships between features in a network system – implemented in ArcGIS as a geometric network.
<b>Raster Dataset</b>	A stereotyped package that holds or organizes the raster data (as Raster Band elements; see below).
<b>Topology</b>	An extended UML Package that represents the shared geometry of a set of Feature Classes from a Feature Dataset.
<b>Elements (in alphabetical order)</b>	
<b>Abstract Class</b>	A standard UML Abstract Class, representing a concept and set of fields, that can be shared by multiple Feature Classes. Feature Classes that connect to an Abstract Class via an Inheritance relationship gain all of its fields. Since the geodatabase does not directly support Abstract Classes, inherited fields are exported into the definition of each child Feature Class when generating a schema from the model.
<b>Coded Value Domain</b>	An extended UML Class, representing a set of valid values that might apply to any type of attribute.
<b>MultiPatch</b>	An extended UML Class, representing the ArcGIS MultiPatch.
<b>Multipoint</b>	An extended UML Class, representing the ArcGIS Multipoint.
<b>Point</b>	An extended UML Class, representing the ArcGIS Point.
<b>Polygon</b>	An extended UML Class, representing the ArcGIS Polygon.
<b>Polyline</b>	An extended UML Class, representing the ArcGIS Polyline.
<b>Range Domain</b>	An extended UML Class, representing a valid range of numeric values that might apply

Toolbox Icon	Description
	to a numeric type of attribute.
<b>Raster Band</b>	An extended UML Class, representing one layer of a matrix of cell values. Every Raster Dataset (above) contains one or more Raster Bands.
<b>Raster Catalog</b>	An extended UML Class, representing a collection of Raster Datasets in the geodatabase.
<b>Raster StorageDef</b>	An extended UML Class, representing the storage properties for a Raster value in the geodatabase; this information is required when a Raster Dataset package element is created (see above).
<b>Spatial Reference</b>	An extended UML Class that defines the spatial reference information of your schema, such as a coordinate system and XYTolerance. You can define one or more Spatial Reference elements, which you link to other ArcGIS elements via their <b>Spatial Reference</b> Tagged Value.
<b>Subtype</b>	An extended UML Class, holding a subset of the attributes of an element in the Feature Dataset (see above).
<b>Table (Object Class)</b>	An extended UML Class, representing a collection of nonspatial data of the same type or Class.
<b>Relationships (in alphabetical order)</b>	
<b>Association</b>	A normal UML Association connector.
<b>Connectivity Rule (Edge-Edge)</b>	An extended UML N-ary Association that models the valid relationships between edge elements in a Geometric Network. For an example, see <i>Connectivity Rule Examples</i> .
<b>Connectivity Rule (Edge-Junction)</b>	An extended UML Association that models the valid relationships between edge and junction elements in a Geometric Network. For an example, see <i>Connectivity Rule Examples</i> .
<b>Dependency</b>	A normal UML Dependency connector.
<b>Generalization</b>	Indicates inheritance from the specific classifier to a general classifier.

Toolbox Icon	Description
<b>Relationship Class</b>	An extended UML <i>Association</i> , providing the relationship between: <ul style="list-style-type: none"> <li>Two elements in the Feature Dataset, or</li> <li>An element in the Feature Dataset and an Object Class element</li> </ul>
<b>Relationship Class</b>	An extended UML <i>Association Class</i> , providing the attributed relationship between: <ul style="list-style-type: none"> <li>Two elements in the Feature Dataset, or</li> <li>An element in the Feature Dataset and an Object Class element</li> </ul>
<b>Relationship Rule</b>	An extended UML Association that determines which subtypes can be related in the geodatabase.
<b>Subtype</b>	An extended UML Association, providing the relationship between a Feature Class element and a Subtype element.
<b>Topology Rule</b>	An extended UML Association that connects Feature Class and Subtype elements in the geodatabase.
<b>Attributes (in alphabetical order)</b>	
<b>Attribute Index</b>	An extended UML Attribute that represents the ArcGIS Attribute Index.
<b>Domain Coded Value</b>	An extended UML Attribute that specifies the value of an ArcGIS Coded Value Domain.
<b>Field</b>	An extended UML Attribute that represents an ArcGIS field of the geodatabase, in a Table or Feature Class.
<b>Spatial Index</b>	An extended UML Attribute that represents the ArcGIS Spatial Index.
<b>Subtype Field</b>	An extended UML Attribute that represents the subtype field of an ArcGIS Table or Feature Class.
<b>TopologyMembership</b>	An extended UML Attribute that represents the <b>accuracy ranks</b> of a Feature Class.

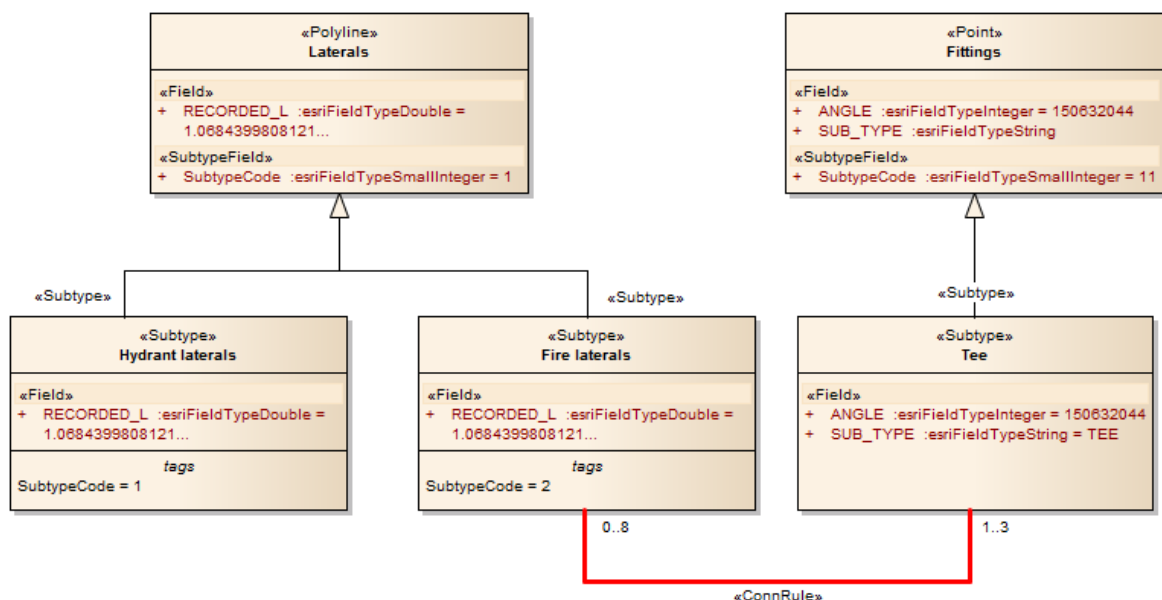
[Learn more](#)

- [Export Modular ArcGIS Schemas](#) <sup>[1962]</sup>
- [Connectivity Rule Examples](#) <sup>[1952]</sup>
- [Topology Example](#) <sup>[1954]</sup>
- [Relationship Rule Example](#) <sup>[1956]</sup>
- [Dependency](#) <sup>[1404]</sup>
- [Association](#) <sup>[1393]</sup>

#### 9.5.1.1.1 Connectivity Rule Examples

In an ArcGIS **Geometric Network** diagram, you can use one or other of the two **Connectivity Rule** relationships - **Edge-Junction** or **Edge-Edge**. These examples illustrate the use of each type.

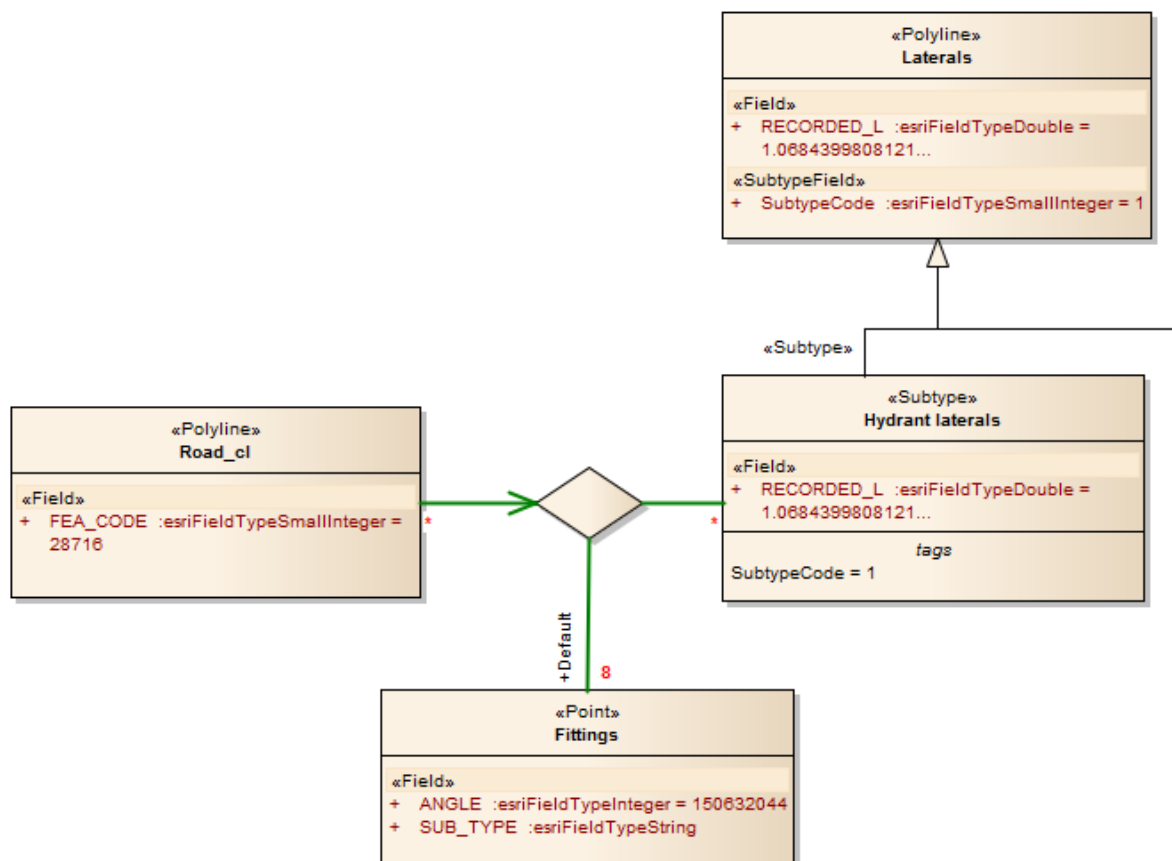
##### Edge-Junction Connectivity Rule



- The Connectivity-Rule (Edge-Junction) connector is a UML binary Association connector
- The connection includes one **edge** element ( «Point», or «Subtype» with «Point» as parent) and one **junction** element ( «Polyline» , or «Subtype» with «Polyline» as parent )
- Cardinality can be set from the source and target **Multiplicity** fields on the connector Properties dialog
- You can set the **Source Role** or **Target Role** fields to **Default** on the connector Properties dialog
- All the elements within this Edge-Junction rule must be held in the «GeometricNetwork» package

##### Edge-Edge Connectivity Rule





- The Connectivity-Rule (Edge-Edge) connector is a UML N-ary Association connector
- The connection should include two **edge** elements («Polyline», or «Subtype» with «Polyline» as parent) and any number of **junction** elements («Point», or «Subtype» with «Point» as parent)
- It is recommended that you use a **Direct Association** connector, drawn from one of the edge elements to the **N-ary** element, to indicate the 'from' Class - in the above diagram, **Road\_cl** is the edge element that is set as the 'from' Class; for the rest of connection, you can use Association connectors to connect the edge or junction element and the N-ary element, drawn either **from** the edge or junction elements **to** the N-ary element, or **from** the N-ary element **to** the edge or junction elements
- Cardinality can be set from the source or target element **Multiplicity** fields on the connector Properties dialog; you only need to set the multiplicity of **one** end of the connector - if **both** ends are set, only the multiplicity of the **target** end is used
- You must mark **one** of the Junction-N-ary connections as **Default**, using the **Source Role** or **Target Role** field on the connector Properties dialog
- All the elements within this Edge-Edge rule must be held in the «GeometricNetwork» package

#### Learn more

- [Connector Properties](#) <sup>[1126]</sup>
- [Source Role](#) <sup>[1130]</sup>
- [Target Role](#) <sup>[1132]</sup>

### 9.5.1.1.2 Topology Example

In geodatabases, **topology** defines the spatial relationship between geographic features; that is, how Point, Polyline, and Polygon features share coincident geometry. Topology is fundamental to **data integrity** in a GIS database. In the Enterprise Architect ArcGIS profile, you use a «*Topology*» Package to model data integrity among the Feature Classes.

Modeling topology in the Enterprise Architect ArcGIS model is simple:

1. Select a «*FeatureDataset*» Package in which to create topology relationships.
2. Open the **diagram** under the «*FeatureDataset*» Package.
3. From the Diagram Toolbox ArcGIS **Network Features** page, drag and drop a «*Topology*» Package icon onto the diagram; this creates a Package that will contain all the elements and relationships that are required for topology definition.

A Topology defined in Enterprise Architect has these characteristics:

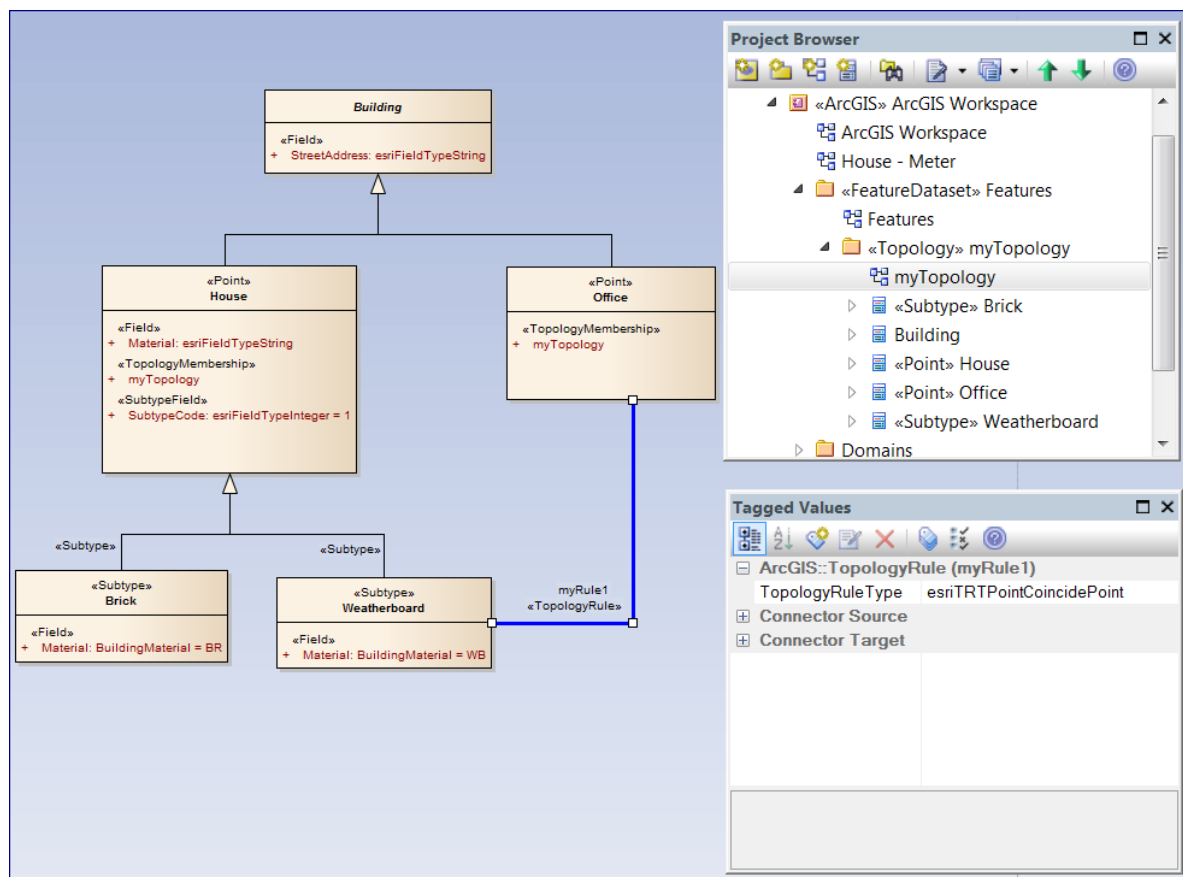
- The «*Topology*» Package cannot be created outside a «*FeatureDataset*» Package
- Within one «*FeatureDataset*» Package, multiple «*Topology*» Packages can be created
- A Feature Class (Point, Polyline or Polygon) can only participate in one «*Topology*» Package
- A Feature Class cannot participate in both a «*Topology*» Package and a «*GeometricNetwork*» Package

#### Elements of Topology

Element	Description	See also
<b>Name</b>	You can define the name for the Topology as the « <i>Topology</i> » Package name.	<a href="#">ArcGIS Toolbox Pages</a> <sup>[1946]</sup>
<b>List of Feature Classes</b>	Either: <ul style="list-style-type: none"> <li>• Create new Feature Classes from the Diagram Toolbox or</li> <li>• Drag existing Feature Classes from the Project Browser into the «<i>Topology</i>» Package</li> </ul>	
<b>X,Y Cluster Tolerance and Z Cluster Tolerance</b>	You define the cluster tolerance values using the <b>ClusterTolerance</b> and <b>ZClusterTolerance</b> Tagged Values of the « <i>Topology</i> » Package.	
<b>Accuracy ranks</b>	Accuracy ranks are defined using the Tagged Values of the <b>TopologyMembership</b> attribute, which you can create using the <b>TopologyMembership</b> icon on the Topology page of the Diagram Toolbox.  Add this stereotyped attribute to each Feature Class element and then set a value for each rank. <ul style="list-style-type: none"> <li>• The <b>name</b> of the attribute should be the name of the «<i>Topology</i>» Package</li> <li>• You do not need to set the <b>type</b> of the attribute</li> </ul>	

Element	Description	See also
	<p>Each Feature Class only has one TopologyMembership attribute. If you do not add a TopologyMembership attribute to a Feature Class, the ArcGIS exporter will generate a set of default ranking values for you. The values for <b>XYRank</b> and <b>ZRank</b> are between 1 and 50.</p>	
<b>Topology Rules</b>	<p>Topology Rules are represented by a UML Association connector that has the «<i>TopologyRule</i>» stereotype. You can create the connector using the <b>Topology Rule</b> icon on the Topology page of the Diagram Toolbox.</p> <p>Use this connector to link:</p> <ul style="list-style-type: none"><li>• Two Feature Class («<i>Point</i>», «<i>Polyline</i>» or «<i>Polygon</i>») elements</li><li>• Two «<i>Subtype</i>» elements</li><li>• A Feature Class («<i>Point</i>», «<i>Polyline</i>» or «<i>Polygon</i>») element to a «<i>Subtype</i>» element</li><li>• A Feature Class («<i>Point</i>», «<i>Polyline</i>» or «<i>Polygon</i>») itself, or</li><li>• A «<i>Subtype</i>» element itself</li></ul> <p>The <b>TopologyRuleType</b> tag is used to define the type of Topology Rule.</p>	

#### Example Topology Rule connection



### 9.5.1.1.3 Relationship Rule Example

In ArcGIS modeling, you can use **relationship rules** to **refine** the cardinality of a **«RelationshipClass»** connector between a source Feature Class or Table and a destination Feature Class or Table; a Relationship Class connector only defines the initial cardinality, such as one-to-many or many-to-many.

A relationship rule in Enterprise Architect is represented by a **«RelationshipRule»** connector, a stereotyped UML Association connector, which you can create using the **Relationship Rule** icon on the ArcGIS Core page of the Diagram Toolbox. You set the cardinality from the source and target **Multiplicity** fields on the connector Properties dialog.

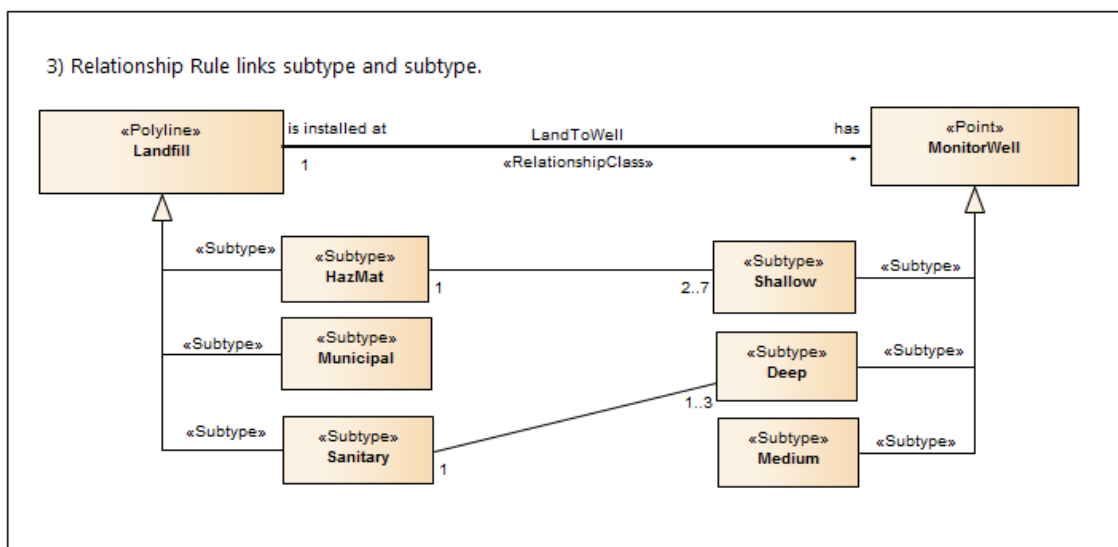
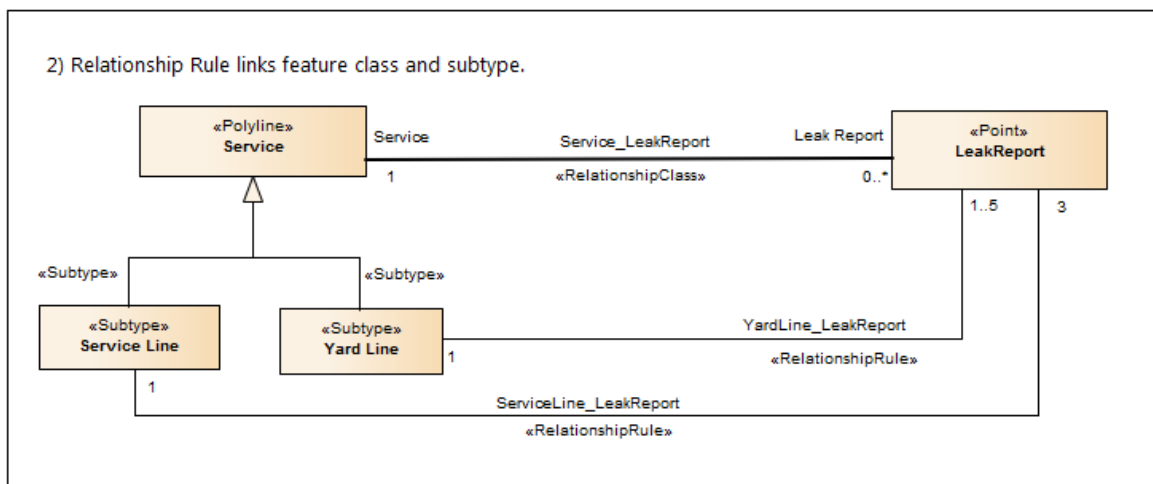
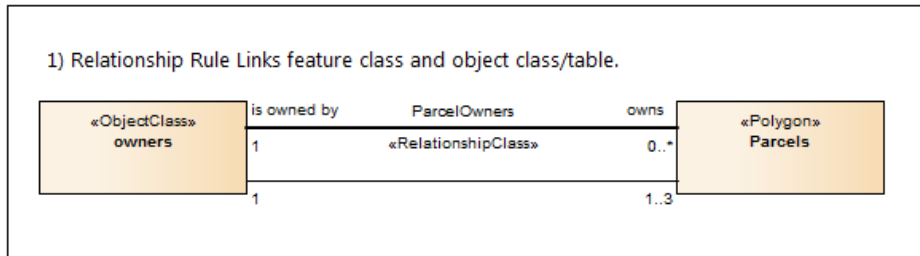
When creating a **«RelationshipRule»** connector between two objects, you must have:

- An existing **«RelationshipClass»** connector between the two objects that you want to define the relationship rule for; if there is no connector, the **«RelationshipRule»** you create is ignored during ArcGIS schema generation
- A cardinality range at each end that is compatible with the cardinality of the parent **«RelationshipClass»**; for example, if you define a cardinality of **1-M** in a **«RelationshipClass»** connector, the source end of the **«RelationshipRule»** connector must be **1**, while you can set the target end of the **«RelationshipRule»** to a specific number such as **3** (see the example diagrams in this topic)

Relationship rules can also restrict the type of object in the source Feature Class or Table that can be related to a certain kind of object in the destination Feature Class or Table. For example, if the source Class has **no** subtype elements, the relationship rule applies to **all** features. If the source Class **has** subtype elements and the **«RelationshipRule»** is linked to one of the subtype elements, this means only the associated subtype element is related to the **«RelationshipRule»**. The same restriction is also applied to the destination Feature Class or Table.

## Examples

This diagram provides three examples of possible «RelationshipRule» connections in an ArcGIS model. A custom Line Thickness has been applied to highlight the Relationship Class connectors, and the «RelationshipRule» stereotype label has been hidden where appropriate:



### 9.5.1.2 Setting ArcGIS Coordinate Systems

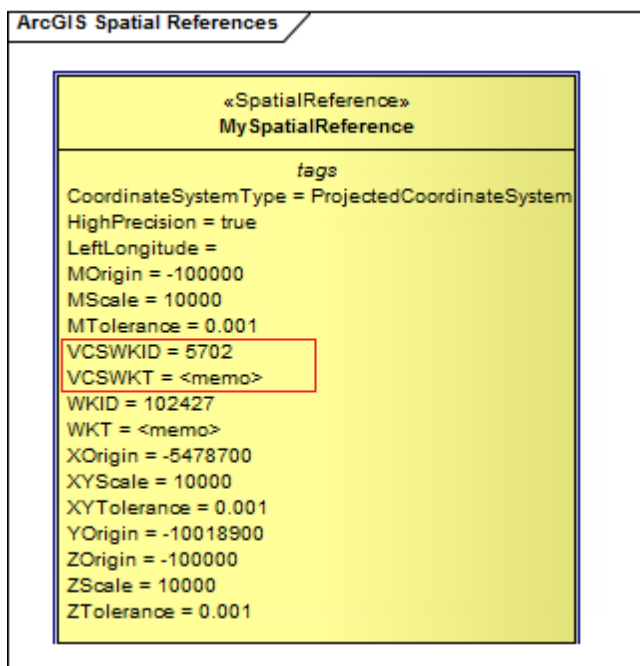
ArcGIS Feature Classes and Feature Datasets use **spatial references**, which consist of a coordinate system and associated values such as XY resolution and various tolerance values.

You can capture spatial reference properties using a Class stereotyped as «*SpatialReference*», which is available from the ArcGIS Toolbox pages. The ArcGIS model pattern includes a Package named **Spatial References**, as a placeholder for creating such elements.

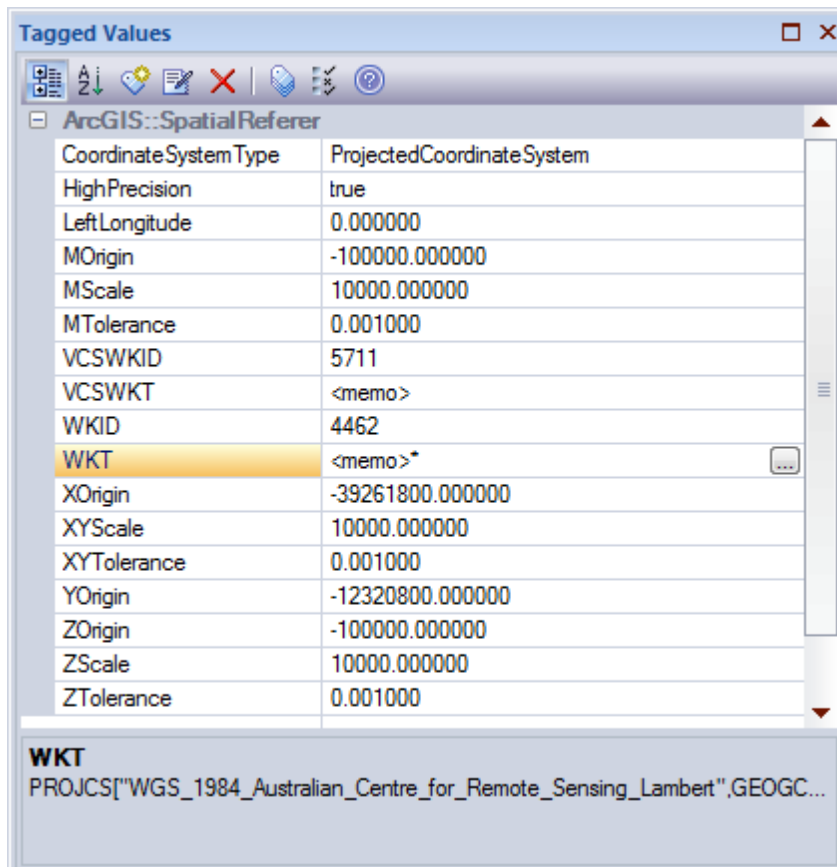
To help you model spatial reference properties, Enterprise Architect provides a dialog for selecting one of the predefined coordinate systems supported by ArcGIS. When you select a Geographic or Projected coordinate system, Enterprise Architect automatically inserts default values for the associated properties, such as **Well Known Text (WKT)**, resolution, precision or tolerances. These values are held as Tagged Values on the «*SpatialReference*» element.

You can also add vertical coordinates to a selected Geographic or Projected coordinate system; the vertical coordinate is loaded to the **VCSWKID** and **VCSWKID** Tagged Values on the «*SpatialReference*» element.

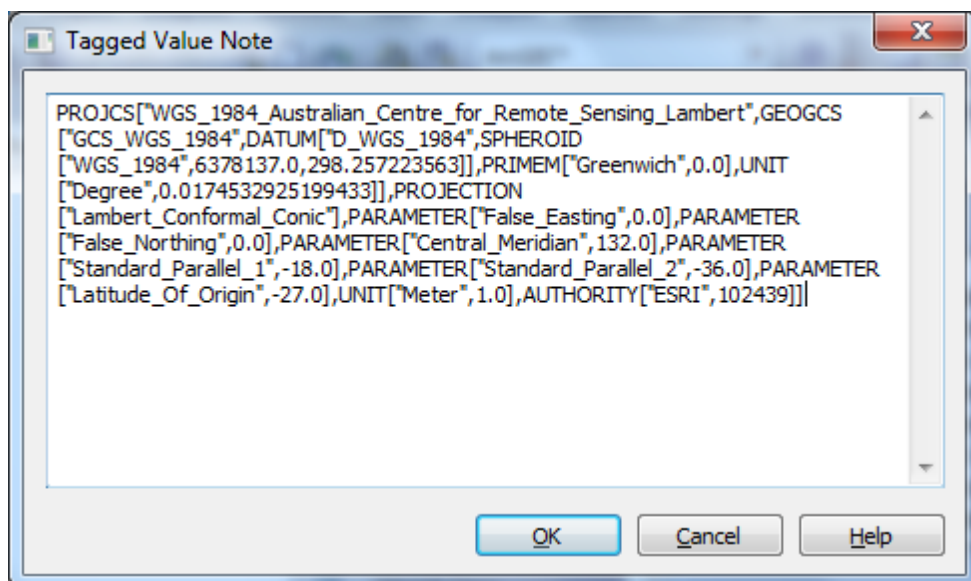
This is an example «*SpatialReference*» element:



Looking at the **WKT** Tagged Value in the Tagged Value window for this element, you can see that the *WGS 1984 Australian Centre for Remote Sensing Lambert* Projected Coordinate system has been selected.



You can expand the information held in this Tagged Value by viewing its Tagged Value Note.



#### Define a Spatial Reference element

Step	Action	See also
1	Open the diagram under the <i>Spatial References</i> package of your ArcGIS model.  (You can actually use any ArcGIS diagram in your model to define Spatial Reference elements; however, this diagram is a convenient placeholder created by Enterprise Architect's model pattern for ArcGIS.)	
2	Drag a Spatial Reference element from the Workspace page of the ArcGIS Core Toolbox onto the diagram.	
3	Right-click on the Spatial Reference element, and select the <b>Extensions   ArcGIS   Set Coordinate System</b> menu option.  The Set Coordinate System dialog displays.	
4	Expand the Geographic or Projected Coordinate Systems hierarchy as appropriate and click on the required coordinate system in the list.	
5	If you want to also apply a vertical coordinate system, click on the [ ... ] button at the right of the <b>Vertical Coordinate</b> field.  The Set Vertical Coordinate System dialog displays, containing a hierarchy that you again expand and from which you select a listed vertical coordinate system.  Click on the <b>OK</b> button to return to the Set Coordinate System dialog; the <b>Vertical Coordinate</b> field now displays the system you selected.	
6	Click on the <b>OK</b> button to close the dialog and return to the diagram.  The Tagged Values for the Spatial Reference element are updated with the Coordinate System information you have selected..	

### Notes

- You can refer to a «*SpatialReference*» Class from any other Feature Dataset or Feature Class in your model, using the **SpatialReference** Tagged Value; the «*SpatialReference*» Class thus provides a single point of control, should you need to change the Spatial Reference information later
- If a Feature Class element references a «*SpatialReference*» Class that contains a vertical coordinate, set the **HasZ** Tagged Value on that Feature Class element to **true** if you want this Feature Class element to store three-dimensional data
- If you do not refer to a «*SpatialReference*» Class from any Feature Dataset or Feature Class in your ArcGIS model, the system will generate an XML schema with the *Unknown* Spatial Reference type for these elements



## 9.5.2 Export ArcGIS XML Workspace

When you have modeled your Geodatabase Workspace XML Document (containing the ArcGIS schema), you can export it to an external directory (using the *Publish Model Package* facility), from which you can then import it to the **Esri ArcCatalog**.

**Access** Click on an ArcGIS stereotyped package (your ArcGIS Workspace Package) in the Project Browser, then either:

**Extensions | Publish | ArcGIS**

**Extensions | ArcGIS | Export to ArcGIS Workspace XML**

**Project | Model Publisher**

**Project | Model Import/Export | Export Package to XMI File: Publish**, or

**Right-click | Import/Export | Export Package to XMI: Publish**

**Right-click | Extensions | ArcGIS | Export to ArcGIS Workspace XML**

### Export the Workspace

Field/Option/ Button	Action	See also
<b>Root Package</b>	Display the name of the selected ArcGIS Workspace package.	
<b>Filename</b>	Type in or browse for the file path into which the XML file is to be generated.	
<b>XML Type</b>	Select ArcGIS as the XML/XMI version to export the package to.	
<b>Format XML Output</b>	Format the output into readable XML (this takes a few more seconds at the end of the run).	
<b>Write Log File</b>	Write a log of the export activity (recommended). The log file is saved to the directory into which the XML file is exported.	
<b>View XML</b>	View the exported XML file.	
<b>Export</b>	Initiate XML export.	
<b>Close</b>	Close this dialog.	
<b>Progress</b>	Observe the progress of the XML export.	

### Notes

- The MDG Technology for ArcGIS is available in the Professional, Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect
- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Export XML** permission to export to XML
- Before exporting your model to an ArcGIS schema, you must define at least one Spatial Reference element; Spatial Reference elements are referred to by other schema elements via a dynamically linked Tagged Value, named *SpatialReference*
- The *DefaultSpatialReference* tag on an ArcGIS **package** is used to specify a Spatial Reference that can be applied to all Feature Datasets and Feature Classes in the workspace; therefore, you do not need to apply a Spatial Reference element to **each** Feature Dataset or Feature Class
- If you do not reference a Spatial Reference Class from any Feature Dataset or Feature Class in your ArcGIS model, Enterprise Architect by default will generate an XML schema with *Unknown type of Spatial Reference* for these elements

### Learn more

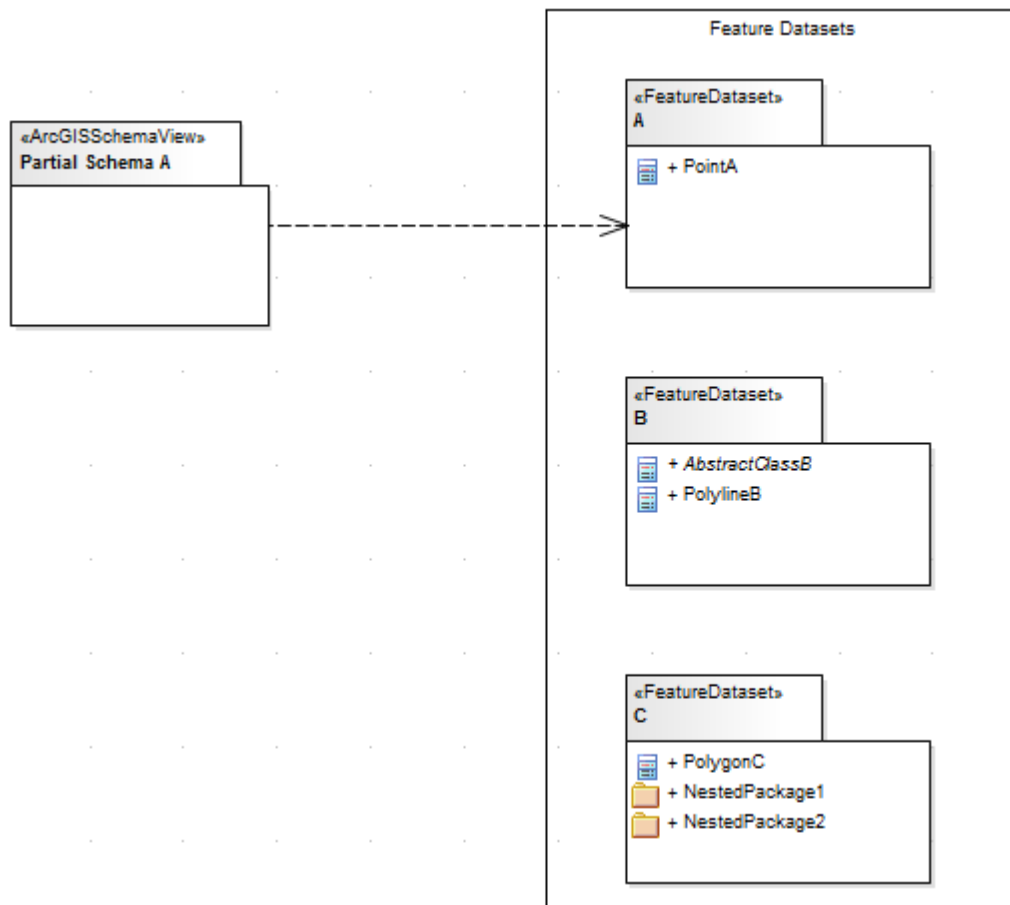
- [Geodatabase Design for ArcGIS](#)<sup>[1944]</sup>
- [Publish Model Package](#)<sup>[476]</sup>
- [List of Available Permissions](#)<sup>[329]</sup>
- [Import ArcGIS XML Workspace](#)<sup>[1968]</sup>
- [Setting ArcGIS Coordinate Systems](#)<sup>[1958]</sup>
- [Validate an ArcGIS Workspace](#)<sup>[1970]</sup>

## 9.5.3 Export Modular ArcGIS Schemas

In Enterprise Architect, in addition to exporting your **entire** ArcGIS workspace, you can also export **partial schemas**. This is useful if you have a large geodatabase schema, as might be defined in an industry reference model. You might require the entire schema in some situations, but only require small parts of it for particular spatial applications, such as field data collection. In such a scenario, you would want to export a schema that contains only the Feature Classes, Tables and Domains that your field data application uses – without duplicating parts of your original schema model. You use the «**ArcGISSchemaView**» stereotyped Package for this purpose.

An «ArcGISSchemaView» Package is modeled as a subPackage of an ArcGIS **Workspace** Package. You can define any number of «ArcGISSchemaView» Packages - each representing a different subset of the **geodatabase schema**. You specify which parts of the schema are included by drawing a UML **Dependency** connector from the «ArcGISSchemaView» Package to each included Package. When you **export** the «ArcGISSchemaView» Package, the system includes any other Packages that your included Packages depend on (via Dependency connectors).

This figure shows a partial schema that includes only one of the three Feature Datasets from the complete schema.



### Create an ArcGISSchema Package

Step	Action	See also
1	Create or open an ArcGIS diagram within your ArcGIS Workspace.	
2	Drag the <b>ArcGIS Schema View</b> icon from the <b>Core</b> Diagram Toolbox onto your diagram.  A prompt displays to enter the Package name.	
3	Type in a meaningful Package name and click on the <b>OK</b> button.	
4	Drag onto the diagram any other Packages that you want to include in the exported schema.  (You could achieve the same result using the child diagram of the «ArcGISSchema View» Package to draw the included Packages).	

Step	Action	See also
5	Draw a Dependency connector from the «ArcGISSchemaView» Package to each of the other Packages.	

### Notes

- Defining the Dependency relationships on a **diagram** is convenient, but not necessary; as long as the dependencies are defined in the **model** – irrespective of whether they exist on a diagram – the ArcGIS schema exporter will use them
- You can arrange your dependency diagrams in whatever part of the ArcGIS Workspace seems appropriate – the diagrams can reside under the «ArcGISSchemaView» Package itself or under any other element within the ArcGIS Workspace

### Export an ArcGIS Schema View for use with ArcCatalog

Step	Action	See also
1	Select the <b>ArcGIS Schema View</b> Package in a diagram or in the Project Browser.	
2	Right-click and select <b>Extensions   ArcGIS   Export to ArcGIS Workspace XML</b> .	
3	Identify the target file and click on the <b>Export</b> button.  The system generates a Workspace XML document containing only the elements associated with the ArcGIS Schema View package.	

### Which related elements are included when you export an ArcGIS Schema View Package?

The following rules apply when you export an ArcGIS Schema View Package:

- Dependencies are modeled using the UML **Dependency** connector
- All elements of a Package that the ArcGIS Schema View depends on (directly or indirectly) are included in the generated schema
- All fields **inherited from** Abstract Classes by included elements are exported, regardless of the Package in which the Abstract Classes reside
- All **Coded Value Domain** elements to which included elements refer are exported, regardless of the Package in which the Coded Value Domain elements reside
- If an ArcGIS Schema View Package depends on one or more subPackages of a **Feature Dataset** Package, the Feature Dataset is exported with only those elements contained in the linked subPackages
- If a field of an included element refers to a **Coded Value Domain** element, that Coded Value Domain element is exported, irrespective of whether the ArcGIS Schema View Package has an explicit

dependency on the Coded Value Domain element's Package

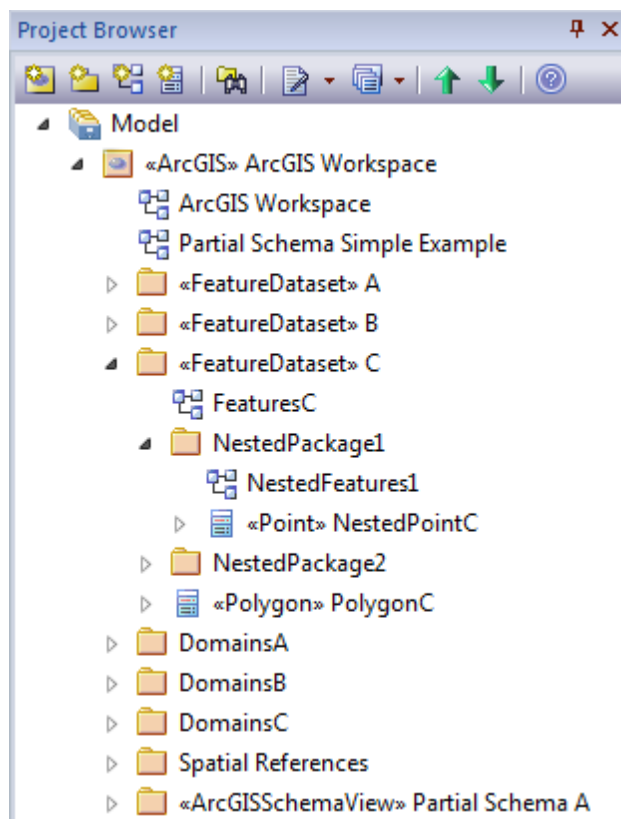
- If an included element has a **Relationship Class** connector to another element X AND element X is not already included by the ArcGIS Schema View, neither element X nor the Relationship Class connector are exported; the log file will hold a list of names of the Relationship Class connectors that, for this reason, are not exported

### Examples of modeling partial schemas

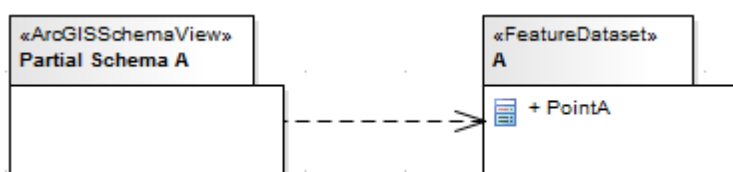
Consider this complete Workspace, which includes three Feature Datasets named **A**, **B** and **C**, and three Packages of Coded Value Domains named **DomainsA**, **DomainsB** and **DomainsC**:



The corresponding model hierarchy in the Project Browser looks like this:

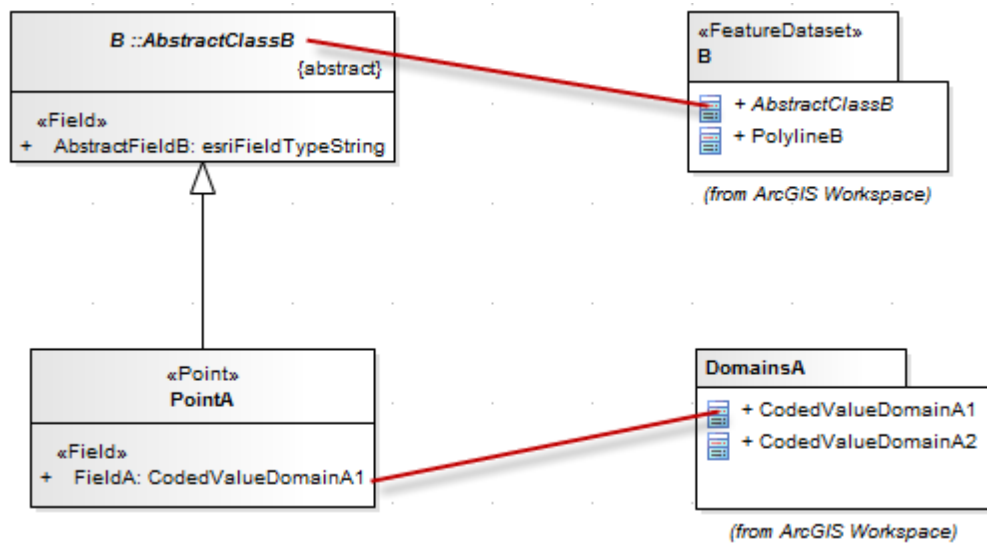


Assume that we want to export only Feature Dataset A and its required elements. We could model the Schema as a partial schema that includes a single Feature Dataset, as shown:

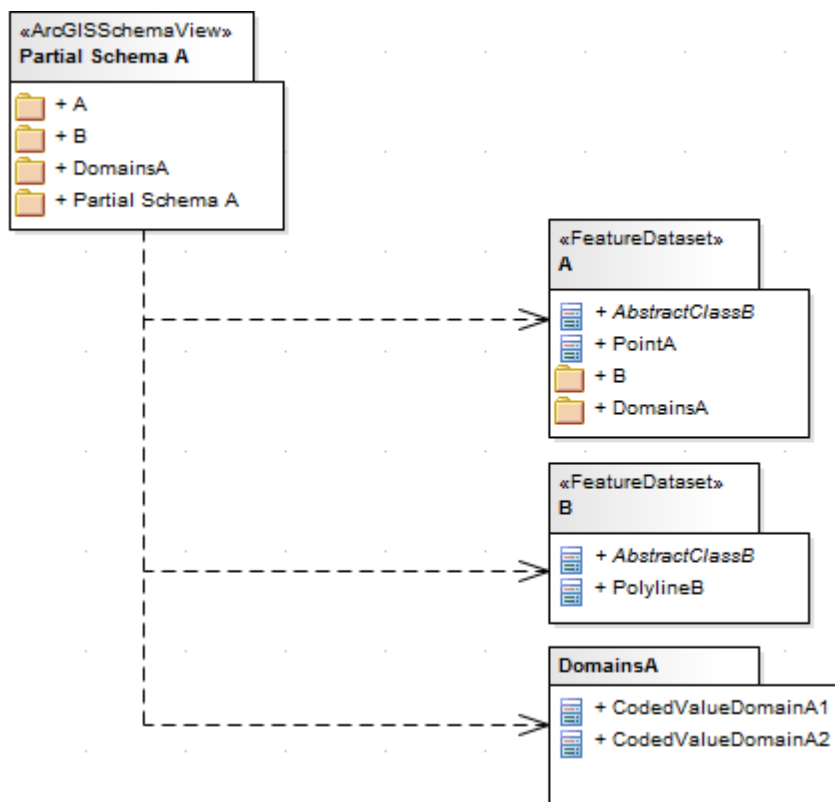


(This diagram is equivalent to the first diagram provided at the start of the topic.) Assuming that **Point A** depends on no other elements, the resulting schema would include only **FeatureDataset A** with its **Feature Class**, Point A.

Now assume that Point A inherits from the Abstract Class **AbstractClassB** (defined in **FeatureDataset Package B**) and that one of A's fields has type **CodedValueDomainA1** defined in the **DomainsA** Package (as in the model diagram below). Now, the same Partial Schema model above would result in an exported schema that included the fields of AbstractClassB and CodedValueDomainA1, even though Partial Schema A does not explicitly depend on Package B or Package DomainsA, because partial schemas automatically include elements that are related by inheritance or are referred to by field types. The exporter thus helps you to generate valid ArcGIS schemas by including such required elements.



If we wanted to include all **CodedValueDomains** in **DomainsA** and all **Feature Classes** in **FeatureDataset B** (including any domains they depended on), we could model this situation to include entire Packages of elements in a partial schema via direct and indirect UML Dependency connectors, as shown.



Finally, let's assume we want to create a partial schema that includes only the elements in **NestedPackage1**. We model the scenario as a partial schema Package that refers to nested Packages within a Feature Dataset.



The resulting schema would include a Feature Dataset named **C** that contained all elements within **NestedPackage1**. The elements in **NestedPackage2** would be excluded as would **PolygonC** (assuming no explicit relationships existed with the elements of NestedPackage1).

### 9.5.4 Import ArcGIS XML Workspace

If you have a Geodatabase Workspace XML Document (containing the ArcGIS schema) you can import it into your Enterprise Architect project as a UML model.

**Access** Click on the target Package in the Project Browser, then either:

**Extensions | Import | ArcGIS**

**Extensions | ArcGIS | Import ArcGIS Workspace XML**

**Project | Model Import/Export | Import Package From XMI File: Other XML File Formats | ArcGIS**

**Right-click | Import/Export | Import Package From XMI File: Other XML File Formats | ArcGIS**

**Right-click | Extensions | ArcGIS | Import ArcGIS Workspace XML**

#### Import a Geodatabase Workspace XML document

Field/Option/ Button	Action	See also
<b>Filename</b>	Type in or browse for the name of the ArcGIS XML file to import.	
<b>Create Diagrams</b>	Select the checkbox to create Class diagrams under the imported Packages.	
<b>Hide System-Level ArcGIS Fields on Diagrams</b>	<p>Select the checkbox to hide these stereotyped attributes:</p> <ul style="list-style-type: none"> <li>• RequiredField</li> <li>• AttributeIndex</li> <li>• SpatialIndex</li> </ul> <p>on these stereotyped Classes:</p> <ul style="list-style-type: none"> <li>• Point</li> <li>• Polyline</li> <li>• Polygon</li> </ul>	



Field/Option/ Button	Action	See also
	<ul style="list-style-type: none"> <li>MultiPatch</li> </ul> <p>The RequiredField and AttributeIndex attributes are also hidden for the Table (Object Class) Class.</p> <p>This option is enabled only when the <b>Create Diagrams</b> checkbox is selected.</p>	
<b>Strip GUIDs</b>	The Strip GUIDs feature is currently mandatory for ArcGIS imports, which means that elements are created 'as new' each time an ArcGIS schema is imported.	
<b>Write Log File</b>	<p>Select the check box to write a log of import activity (recommended).</p> <p>The log file is saved in the directory from which the file is being imported, with the same name as the imported file plus the suffix <i>_import.log</i>.</p>	
<b>View XML</b>	Click on this button to view the XML before import.	
<b>Import</b>	Click on this button to import the ArcGIS XML file.	
<b>Close</b>	Click on this button to close this dialog.	
<b>Help</b>	Click on this button to display this Help page.	
<b>Import Progress</b>	This field indicates the progress of the import.	

**Notes**

- The MDG Technology for ArcGIS is available in the Professional, Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect

**Learn more**

- [Geodatabase Design For ArcGIS](#)<sup>[1944]</sup>
- [ArcGIS UML Profile](#)<sup>[1944]</sup>
- [Export ArcGIS XML Workspace](#)<sup>[1967]</sup>
- [Validate an ArcGIS Workspace](#)<sup>[1970]</sup>

### 9.5.5 Validate an ArcGIS Workspace

When you have developed or imported an ArcGIS model, you can validate it against a set of rules in a system-provided ArcGIS validation table.

**Access**   **Project Browser | Right-click «ArcGIS» Workspace Package | Extensions | ArcGIS | Validate ArcGIS Model**

The menu option launches a validation script on the workspace. While running, the script logs information to the ArcGIS Model Validation tab of the System Output window. Check the script output for errors and warnings.

There are two ways to investigate the errors reported by the model validation script:

- Expand the System Output window and review the errors and warnings directly; you can double-click on a warning or error line to highlight the element or attribute the message relates to, in the Project Browser

**or**

- Copy all of the output to a text file and open the file using your preferred text editor; this is likely to provide cleaner formatting of the script's output

#### **Learn more**

- [Validating ArcGIS Models in Enterprise Architect 10](#) - Online Resource
- [ArcGIS Model Validation Rules table](#) - Online Resource

## 9.6 MDG Technology for ODM

**MDG Technology for ODM** is the implementation of the Object Management Group's **Ontology Definition Metamodel** for Enterprise Architect. It provides ontology modeling capabilities within Enterprise Architect (9.2 or later), enabling you to develop large-scale ontologies within the fully-integrated modeling environment, for your project domain.

Through the MDG Technology for ODM, you have access to the following features:

- A **UML Profile for the Resource Description Framework (RDF)**
- A **UML Profile for Web Ontology Language (OWL)**
- Customized **diagram types** and **toolbox pages**, for convenient access to elements and relationships to model ontologies and resources effectively
- **Model Templates** to get started quickly with ontology modeling
- Commands to **import and export** RDF and OWL files, and to define new **namespaces** for ODM packages and new **labels** for OWL and RDF elements

### Topics

Topic	Detail	See also
<b>Ontologies</b>	<p>Ontologies define a common vocabulary for sharing information in a domain. They provide a standardized, machine-interpretable definition of basic concepts within the domain, and of relations among them. Formally defined ontologies give stakeholders the ability to:</p> <ul style="list-style-type: none"> <li>• Share common understanding of the structure of information among themselves or among software agents</li> <li>• Reuse domain knowledge</li> <li>• Explicitly codify domain assumptions</li> <li>• Separate domain knowledge from operational knowledge</li> <li>• Analyze domain knowledge effectively</li> </ul>	
<b>The OMG ODM</b>	<p>The Ontology Definition Metamodel has been formalized by the OMG as a standard.</p> <p>It defines an important set of enabling capabilities for Model Driven Architecture (MDA) for a formalized representation of business semantics and taxonomies, and knowledge representation based on those formalizations. It defines extensions to the Unified Modeling Language (UML) to provide a custom modeling notation for ontology definition. These extensions are:</p> <ul style="list-style-type: none"> <li>• The Resource Description Framework (RDF), used to represent data, properties and formal semantics of information in the web</li> <li>• Web Ontology Language (OWL), used to represent terms in vocabularies, and their interrelationships; OWL extends the RDF to define both the domain information and the relevant domain meaning</li> </ul>	
<b>Disable MDG Technology for ODM</b>	<p>If you prefer not to use the MDG Technology for ODM in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog (<b>Settings   MDG Technologies</b>).</p>	

Topic	Detail	See also

**Learn more**

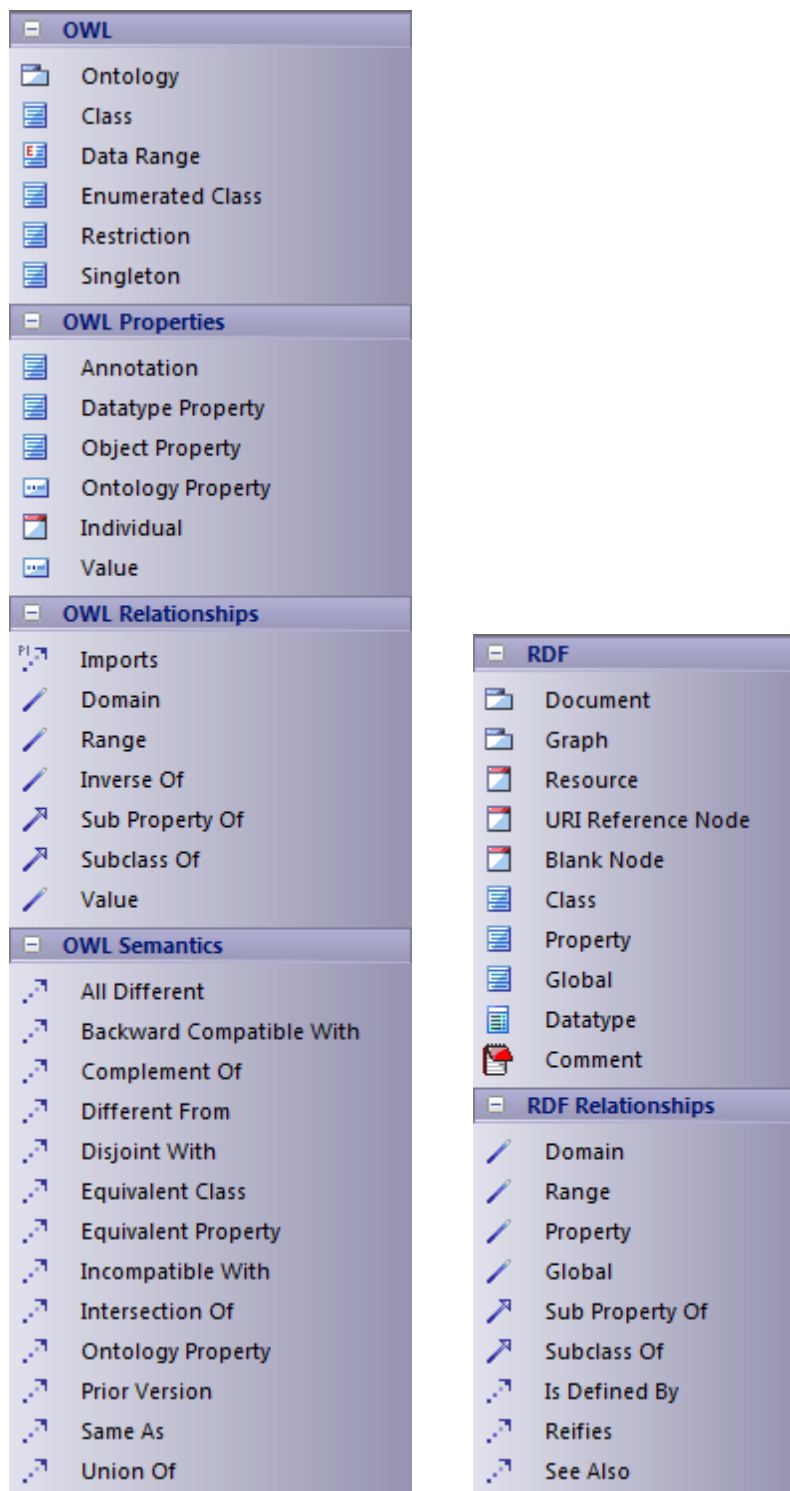
- [Example ODM Diagrams](#) <sup>[1975]</sup>
- [ODM Toolbox Pages](#) <sup>[1972]</sup>
- [ODM Commands](#) <sup>[1980]</sup>
- [New Diagram](#) <sup>[822]</sup>
- [Quick Linker](#) <sup>[896]</sup>
- [Model Wizard](#) <sup>[753]</sup>
- [Toolbox Shortcut](#) <sup>[799]</sup>

**9.6.1 ODM Toolbox Pages**

Enterprise Architect provides two sets of toolbox pages for ODM:

- Web Ontology Language (OWL) pages for Ontology Definition diagrams and Ontology Facts diagrams, and
- Resource Description Framework (RDF) pages for Resource Definition diagrams

**Access**   **Diagram | Diagram Toolbox: More tools | ODM | Web Ontology Language, and Diagram | Diagram Toolbox: More tools | ODM | Resource Description Framework**



### Learn more

- [OWL Elements & Relationships](#) <sup>[1974]</sup>
- [RDF Elements & Relationships](#) <sup>[1977]</sup>
- [Example ODM Diagrams](#) <sup>[1979]</sup>

### 9.6.1.1 OWL Elements & Relationships

This topic explains each of the icons on the Web Ontology Language (OWL) Toolbox pages for Ontology Definition diagrams and Ontology Facts diagrams.

Access [Diagram](#) | [Diagram Toolbox: More tools](#) | [ODM 1.0](#) | [Web Ontology Language](#)

#### OWL Elements:

Toolbox Icon	Description
<b>Ontology</b>	The OWL ontology package, which holds all the OWL modeling elements. You can export the contents of this package to produce the Ontology XML document.
<b>Class</b>	An extended UML Class that represents an OWL Class that describes an instantiable entity with properties and semantic meaning.
<b>Data Range</b>	An extended UML Enumeration that defines a collection of values for an OWL Property.
<b>Enumerated Class</b>	An extended UML Class that defines an OWL Class extension defined by any one of the range of the allowed OWL Individuals.
<b>Restriction</b>	An extended UML Class that defines an OWL Class extension as restricted by the specified property and its allowable values.
<b>Singleton</b>	An extended UML Class, representing an OWL Class for a specific OWL Individual.

#### OWL Properties (Elements):

Toolbox Icon	Description
<b>Annotation</b>	An extended UML Class, representing an OWL Annotation Property definition.
<b>Datatype Property</b>	An extended UML Class, representing an OWL Datatype Property definition.
<b>Object Property</b>	An extended UML Class, representing a semantic OWL Property definition.
<b>Ontology Property</b>	An extended UML Part, representing a property defined on the OWL Ontology.

Toolbox Icon	Description
<b>Individual</b>	An extended UML Object, representing an instance of an OWL Class which defines an individual fact.
<b>Value</b>	An extended UML Part, holding a value defined in an OWL Property or OWL Individual.

**OWL Relationships:**

Toolbox Icon	Description
<b>Imports</b>	An extended UML PackageImports, that enables an OWL ontology to reference another OWL Ontology.
<b>Domain</b>	An extended UML Association, that specifies the OWL Classes that apply the specified OWL Property (Annotation, Datatype or Object Property).
<b>Range</b>	An extended UML Association, that specifies the OWL Class with the value type applicable to the specified OWL Property (Annotation, Datatype or Object Property).
<b>Inverse Of</b>	An extended UML Association, between two opposing, but related OWL Property elements.
<b>Sub Property Of</b>	An extended UML Generalization between two OWL Property elements.
<b>Subclass Of</b>	An extended UML Generalization between two OWL Class elements.
<b>Value</b>	An extended UML Association, defining an OWL Property and value between OWL Classes.

**OWL Semantics (Relationships):**

Toolbox Icon	Description
<b>All Different</b>	An extended UML Dependency between different (ie: unique) OWL Individuals typed by the same OWL Class.

Toolbox Icon	Description
<b>Backward Compatible With</b>	An extended UML Dependency between an OWL Ontology and another that it's backward compatible with.
<b>Complement Of</b>	An extended UML Dependency between an OWL Class and its complement (or opposite).
<b>Different From</b>	An extended UML Dependency between two semantically different OWL Individuals typed by the same OWL Class.
<b>Disjoint With</b>	An extended UML Dependency between two OWL Classes that have no common individuals.
<b>Equivalent Class</b>	An extended UML Dependency between two equivalent OWL Classes.
<b>Equivalent Property</b>	An extended UML Dependency between two equivalent OWL Property elements.
<b>Incompatible With</b>	An extended UML Dependency between an OWL Ontology and another that it is incompatible with.
<b>Intersection Of</b>	An extended UML Dependency between an OWL Class and others it also specializes.
<b>Ontology Property</b>	An extended UML Dependency, representing a property defined on the OWL Ontology.
<b>Prior Version</b>	An extended UML Dependency between an OWL Ontology and its predecessor.
<b>Same As</b>	An extended UML Dependency between two semantically identical OWL Individuals typed by the same OWL Class.
<b>Union Of</b>	An extended UML Dependency between a general OWL Class and others that distinctly specialize it.

[Learn more](#)

- [ODM Toolbox Pages](#)  1972



### 9.6.1.2 RDF Elements & Relationships

This topic explains each of the icons on the Resource Description Framework (RDF) pages for Resource Definition diagrams.

[Access](#) [Diagram](#) | [Diagram Toolbox: More tools](#) | [ODM 1.0](#) | [Resource Description Framework](#)

#### RDF Elements:

Toolbox Icon	Description
<b>Document</b>	The RDF Document package, which holds all the RDF modeling elements. You can export the contents of this package to produce the Resource Description XML document.
<b>Graph</b>	An extended UML Package that represents a set of RDF subject and object triples within the RDF Document.
<b>Resource</b>	An extended UML Object that represents a uniquely identifiable general resource.
<b>URI Reference Node</b>	An extended UML Object that represents a uniquely identifiable external resource.
<b>Blank Node</b>	An extended UML Object that represents a uniquely identifiable internal resource.
<b>Class</b>	An extended UML Class representing an RDF Class, which describes an instantiable resource with properties.
<b>Property</b>	An extended UML Class, representing an RDF Property definition.
<b>Global</b>	An extended UML Class, representing a global RDF Property definition.
<b>Datatype</b>	An extended UML Datatype, representing an RDF Datatype definition.
<b>Comment</b>	A UML Comment element.

#### RDF Relationships:

Toolbox Icon	Description
<b>Domain</b>	An extended UML Association that specifies the RDF Classes that apply the

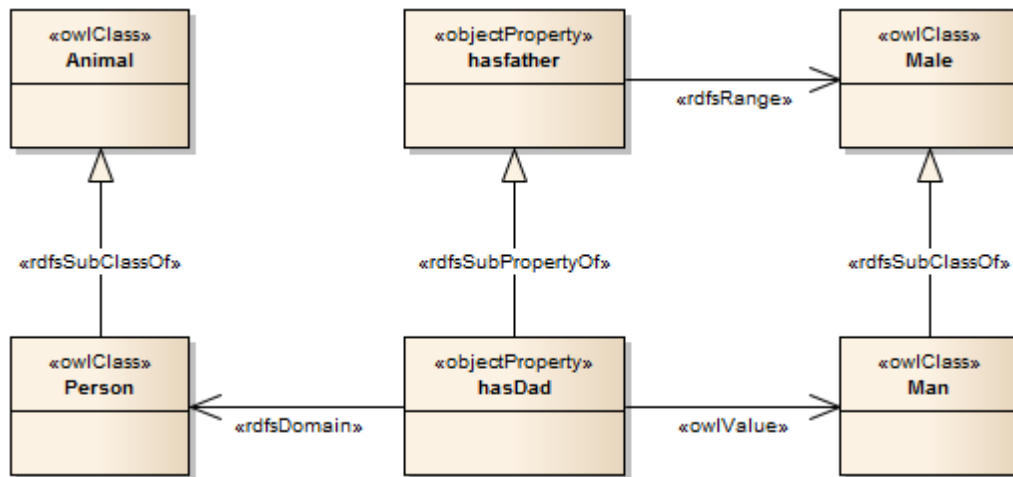
Toolbox Icon	Description
	specified RDF Property.
<b>Range</b>	An extended UML Association that specifies the RDF Class with the value type applicable to the specified RDF Property.
<b>Property</b>	An extended UML Association that defines a RDF property between two RDF Classes.
<b>Global</b>	An extended UML Association that defines a global RDF property between two RDF Classes.
<b>Sub Property Of</b>	An extended UML Generalization between two RDF Property elements.
<b>Subclass Of</b>	An extended UML Generalization between two RDF Class elements.
<b>Is Defined By</b>	An extended UML Dependency between a RDF Resource and another that defines it.
<b>Reifies</b>	An extended UML Dependency between a RDF Resource and another that it reifies.
<b>See Also</b>	An extended UML Dependency between a RDF Resource and another that contains more information about it.

[Learn more](#)

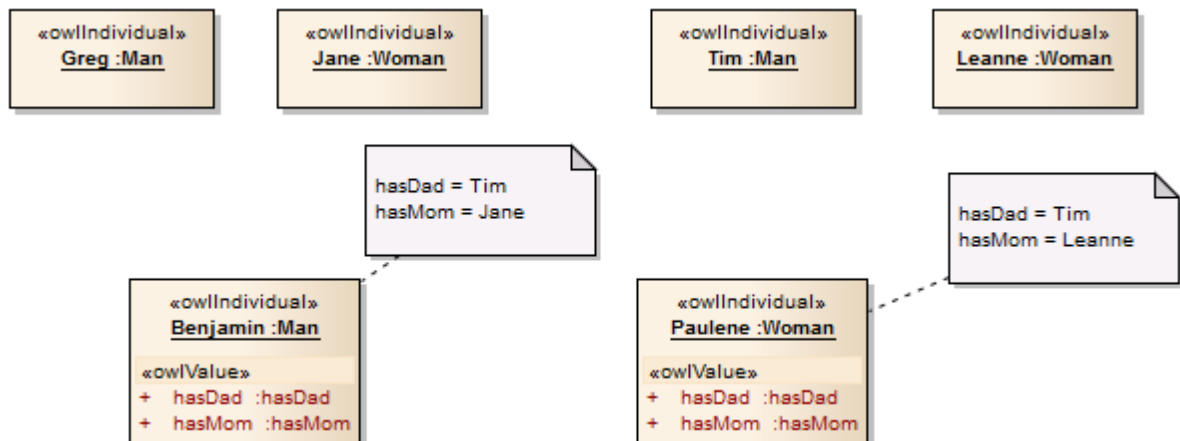
- [ODM Toolbox Pages](#) 

## 9.6.2 Example ODM Diagrams

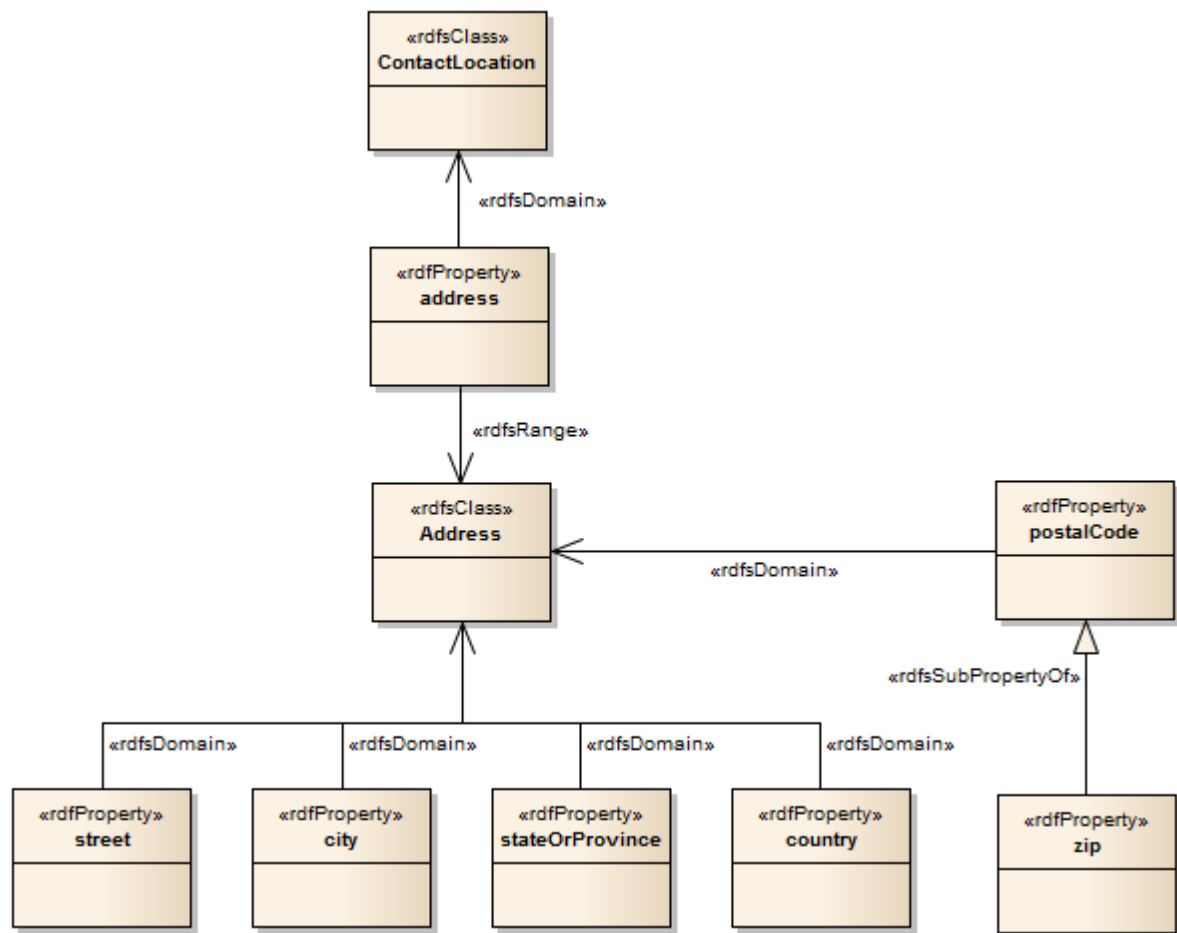
Example OWL Definition Diagram:



Example OWL Facts Diagram:



Example RDF Definition Diagram:



#### Learn more

- [MDG Technology for ODM](#)<sup>[1971]</sup>
- [ODM Toolbox Pages](#)<sup>[1972]</sup>








### 9.6.3 ODM Commands

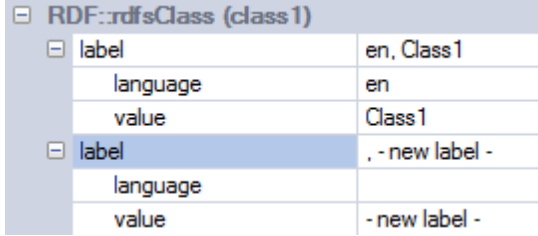

The MDG Technology for ODM provides four management commands to help you maintain your ODM models:

- **Export** OWL/RDF models as an .rdf or .xml file
- **Import** OWL or RDF content (in .owl or .rdf xml file format) as an ODM model package
- Create a new **namespace definition** for an rdfDocument or owlOntology package
- Create a new **label definition** for a valid RDF or OWL element (such as an owlClass, objectProperty, rdfClass or rdfProperty)

**Access**    **Extensions | ODM | <command>**  
**Extensions | Import | OWL/RDF**, or  
**Extensions | Publish | OWL/RDF**

#### Commands:

Command	Detail	See also																		
Export OWL/RDF	<p>In the Project Browser, select the <code>rdfDocument</code> or <code>owlOntology</code> package to export.</p> <p>Select either:</p> <ul style="list-style-type: none"><li>• <b>Extensions   ODM   Export OWL/RDF</b>, or</li><li>• <b>Extensions   Publish   OWL/RDF</b></li></ul> <p>A browser dialog displays, on which you specify the file location to export to, and the <code>.owl</code> or <code>.rdf</code> filename.</p> <p>Click on the <b>Save</b> button.</p> <p>The package content is exported to the specified file.</p>																			
Import OWL/RDF	<p>In the Project Browser, select the <code>rdfDocument</code> or <code>owlOntology</code> package into which to import the <code>.owl</code> or <code>.rdf</code> file.</p> <p>Select either:</p> <ul style="list-style-type: none"><li>• <b>Extensions   ODM   Import OWL/RDF</b>, or</li><li>• <b>Extensions   Import   OWL/RDF</b></li></ul> <p>A browser dialog displays, through which you locate and select the file to import.</p> <p>Click on the <b>Open</b> button.</p> <p>The file content is imported into the selected package.</p>																			
Define New Namespace	<p>Ensure that the <b>Show Duplicate Tags</b> checkbox is selected in the Options dialog (<b>Tools   Options   Objects</b>).</p> <p>In the Project Browser, select the RDF or OWL package to be allocated a new namespace.</p> <p>Select the <b>Extensions   ODM   Define New Namespace</b> command.</p> <p>Double-click on the selected package to display the package Properties dialog, and select the <b>RDF</b> page. Notice the additional <code>namespaceDefinition</code> tag.</p> <table><tr><td></td><td><code>namespaceDefinition</code></td><td><code>ns0, http://myontologies/newOntology</code></td></tr><tr><td></td><td><code>namespacePrefix</code></td><td><code>ns0</code></td></tr><tr><td></td><td><code>namespaceURI</code></td><td><code>http://myontologies/newOntology</code></td></tr><tr><td></td><td><code>namespaceDefinition</code></td><td><code>- new prefix -, - new URI -</code></td></tr><tr><td></td><td><code>namespacePrefix</code></td><td><code>- new prefix -</code></td></tr><tr><td></td><td><code>namespaceURI</code></td><td><code>- new URI -</code></td></tr></table> <p>Click on the <code>namespacePrefix</code> and <code>namespaceURI</code> value fields for the new namespace definition, and type in the appropriate new values.</p> <p>If necessary, click on the previous <code>namespaceDefinition</code> tag and on the  (<b>Delete</b>) icon in the RDF page toolbar.</p>		<code>namespaceDefinition</code>	<code>ns0, http://myontologies/newOntology</code>		<code>namespacePrefix</code>	<code>ns0</code>		<code>namespaceURI</code>	<code>http://myontologies/newOntology</code>		<code>namespaceDefinition</code>	<code>- new prefix -, - new URI -</code>		<code>namespacePrefix</code>	<code>- new prefix -</code>		<code>namespaceURI</code>	<code>- new URI -</code>	
	<code>namespaceDefinition</code>	<code>ns0, http://myontologies/newOntology</code>																		
	<code>namespacePrefix</code>	<code>ns0</code>																		
	<code>namespaceURI</code>	<code>http://myontologies/newOntology</code>																		
	<code>namespaceDefinition</code>	<code>- new prefix -, - new URI -</code>																		
	<code>namespacePrefix</code>	<code>- new prefix -</code>																		
	<code>namespaceURI</code>	<code>- new URI -</code>																		

Command	Detail	See also
	Click on the <b>OK</b> button.	
<b>Define New Label</b>	<p>Ensure that the <b>Show Duplicate Tags</b> checkbox is selected in the Options dialog (<b>Tools   Options   Objects</b>).</p> <p>In the Project Browser, select the RDF or OWL element to be allocated a new label.</p> <p>Select the <b>Extensions   ODM   Define New Label</b> command.</p> <p>Double-click on the selected element to display the element Properties dialog, and select the <b>RDF</b> page. Notice the additional <i>label</i> tag.</p>  <p>Click on the <b>language</b> and <b>value</b> value fields for the new label definition, and type in the appropriate new values.</p> <p>If necessary, click on the previous <i>label</i> tag and on the  (<b>Delete</b>) icon in the RDF page toolbar.</p> <p>Click on the <b>OK</b> button.</p>	

#### Learn more

- [MDG Technology for ODM](#)

## 9.7 MDG Technology For GML

MDG Technology for GML is the implementation of the Open Geospatial Consortium's **Geography Markup Language (GML) 3.3** for Enterprise Architect, which provides an XML grammar for geographical feature modeling capabilities within Enterprise Architect at or later than Release 10.

Through the MDG Technology for GML, you can:

- Apply a **UML Profile for the Geography Markup Language (GML) 3.3**
- Make use of **customized diagram types** and **toolbox pages**, for convenient access to elements and relationships to model **geographical features** effectively
- Generate **GML Application Schema** files

### \*Beta Implementation\*

This is a BETA release of the GML profile and associated tools. As such, it is supplied without warranty of any kind and is subject to change without notice during the Beta period. Sparx Systems welcomes any feedback, issues, suggestions and comments on this implementation. As with all Beta software, please take due care when using the GML profile in a production environment.

### Notes

- The MDG Technology for GML is available in the Professional, Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect

### Learn more

- [UML Profile for GML](#)<sup>[1983]</sup>
- [Generate GML Application Schema](#)<sup>[1987]</sup>
- [The Open Geospatial Consortium GML Specification](#) (Online Resource)

### 9.7.1 UML Profile for GML

The MDG Technology for GML is built into the Enterprise Architect installer. A key component of the technology is the UML Profile for GML.

### Access   **Extensions | GML**

### Features

Feature	Detail	See also
<b>Profile Support</b>	<p>You can develop GML constructs quickly and simply, through use of the built-in MDG Technology for GML facilities provided in the form of:</p> <ul style="list-style-type: none"> <li>• A GML diagram type, accessed through the New Diagram dialog</li> <li>• GML pages in the Diagram Toolbox that map GML concepts to appropriately stereotyped UML elements</li> <li>• GML element and relationship entries in the Toolbox Shortcut Menu and Quick Linker</li> </ul>	<p><a href="#">New Diagram</a><sup>[822]</sup></p> <p><a href="#">Diagram Toolbox</a><sup>[792]</sup></p> <p><a href="#">Toolbox Shortcut</a></p>

Feature	Detail	See also
		<a href="#">Quick Linker</a> <sup>[799]</sup> <sup>[896]</sup>
<b>GML Toolbox Page</b>	The GML Toolbox pages contain elements and connectors to model geographical features effectively.	<a href="#">GML Toolbox Pages</a> <sup>[1984]</sup>
<b>UML Classes from ISO</b>	<p>(Optional) You can download the UML Classes implemented in ISO/TC 211 as an XMI file, then import the XMI file into Enterprise Architect as a package containing diagrams and standard UML Classes, which you can reuse in your model.</p> <ul style="list-style-type: none"> <li>Not all UML Classes implemented in ISO/TC 211 have a corresponding mapping in GML; the Classes that have a mapping (as specified in the <i>GML 3.2.1 specification</i>) are specified in the configurable file <b>GMLClassMapping.xml</b> in the Sparx Systems &gt; EA &gt; Config &gt; GML folder</li> <li>The Namespace information for these Classes is specified in the configurable file <b>GMLNamespaces.xml</b> in the Sparx Systems &gt; EA &gt; Config &gt; GML folder</li> </ul>	<a href="#">ISO/TC 211</a> (Online Resource) <a href="#">XMI Import</a> <sup>[478]</sup>
<b>GML Application Schema Generation</b>	<p>Any model you create using the built-in MDG Technology for GML can be exported as a GML Application Schema.</p> <p>Using the configurable file <b>GMLStereotypes.xml</b> in the Sparx Systems &gt; EA &gt; Config &gt; GML folder, you can specify aliases for the standard GML stereotypes. The GML Application Schema Generator will also consider these aliases during Schema generation.</p>	<a href="#">Generate GML Application Schema</a> <sup>[1987]</sup>
<b>Disable MDG Technology for GML</b>	If you prefer not to use the MDG Technology for GML in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog ( <b>Settings   MDG Technologies</b> ).	<a href="#">MDG Technologies</a> <sup>[1477]</sup>

**Notes**

- The MDG Technology for GML is available in the Professional, Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect

**Learn more**

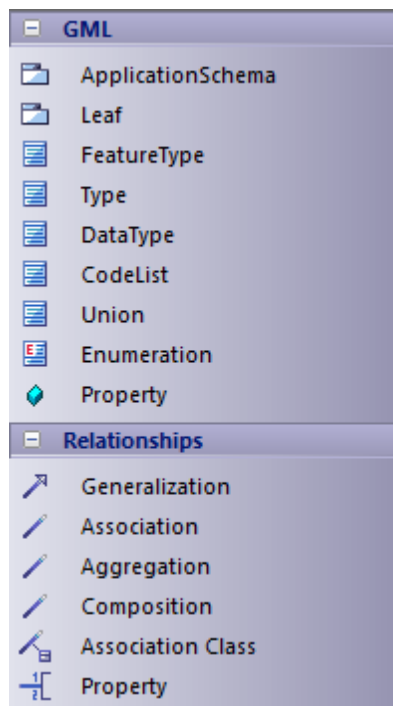
- [MDG Technology For GML](#) <sup>[1983]</sup>

**9.7.1.1 GML Toolbox Pages**

When you begin to model GML concepts and relationships, you can represent them on a GML diagram using the elements and connectors provided in the GML Toolbox pages.

**Access** [Diagram | Diagram Toolbox: More Tools | GML](#)



GML Toolbox pagesGML Toolbox page icons

Toolbox Icon	Description
<b>Packages</b>	
<b>ApplicationSchema</b>	The GML workspace package, which holds all the GML modeling elements. You can export the contents of this package to produce the GML Application Schema.
<b>Leaf</b>	A stereotyped package that can be created under an Application Schema. <i>Leaf</i> is a convenient package for grouping elements within an Application Schema.  If the value of the tag <b>xsdDocument</b> on this package is empty, the contents of this package will be exported as a part of the parent package whose tag <b>xsdDocument</b> is <b>not</b> empty.
<b>Elements</b>	
<b>FeatureType</b>	An extended UML Class that maps to an XSD Global element, whose content model is a globally scoped XML Schema type derived by direct/indirect extension of <b>gml:AbstractFeatureType</b> .
<b>Type</b>	An extended UML Class that maps to an XSD element, whose content model is a

Toolbox Icon	Description
	globally scoped XML Schema type derived by direct/indirect extension of <b>gml:AbstractGMLType</b> .
<b>DataType</b>	An extended UML Class that maps to an XSD Global element, whose content model is a globally scoped XML Schema <b>complexType</b> .
<b>CodeList</b>	An extended UML Class, that maps to: <ul style="list-style-type: none"> <li>• A union of an enumeration and a pattern, or</li> <li>• an external Dictionary</li> </ul>
<b>Union</b>	An extended UML Class that maps to an XSD Choice group, whose members are GML objects or features, or objects corresponding to DataTypes.
<b>Enumeration</b>	An extended UML Class that is a restriction of XSD string with enumeration values.
<b>Relationships</b>	
<b>Generalization</b>	Identifies an element as a specialized descendant of another element, containing additional properties. This relationship is allowed between: <ul style="list-style-type: none"> <li>• Feature Types</li> <li>• DataTypes and Unions</li> <li>• Types</li> </ul> <p>An element can have a maximum of one Generalization connector.</p>
<b>Association</b>	Identifies the connection between two elements. Only navigable Association Ends will be mapped to GML Application Schema.
<b>Aggregation</b>	An extended UML Association that shows that an element is composed of other elements.
<b>Composition</b>	An extended UML Association that is a stronger form of Aggregation, indicating ownership of the whole over its parts.
<b>Association Class</b>	Identifies the attributed connection between two elements. Only navigable Association Ends can be mapped to GML Application Schema.
<b>Attribute</b>	

Toolbox Icon	Description
<b>Property</b>	An extended UML Attribute, whose type is either a property type (if the type is a complex type) or a simple type.
<b>Association Role</b>	
<b>Property</b>	An extended UML Association Role, whose type is always a property type.

### Notes

- In GML, **Association Class** is specific to GML 3.3; when generating GML 3.2.1 Application Schema, the Association and Class of an Association Class are treated as separate entities
- The values for tags on the «Application Schema» and «Leaf» stereotyped packages can be entered using either:
  - [The Tagged Values window](#)<sup>[1134]</sup> or
  - [The Generate GML Application Schema dialog](#)<sup>[1987]</sup>

### Learn more

- [MDG Technology For GML](#)<sup>[1983]</sup>
- [UML Profile for GML](#)<sup>[1983]</sup>

## 9.7.2 Generate GML Application Schema

After you have created a GML model using the built-in MDG Technology for GML, you can generate GML Application Schema from the model Package.

**Access**    **Extensions | GML | Generate GML Application Schema**

### Generate GML Application Schema

Field/Option/ Button	Action	See also
<b>Source Package</b>	Displays the name of the selected «Application Schema» stereotyped package.	
<b>Directory</b>	Type in or browse for the directory into which the application schema file is to be generated.	
<b>GML Version</b>	Click on the drop-down arrow and select the version of the GML Application Schema. Currently versions <b>3.3</b> and <b>3.2.1</b> are supported.	

Field/Option/ Button	Action	See also										
XML Encoding	Click on the drop-down arrow and select the XML encoding scheme to be used.											
Ignore Element and Property Visibility	Click on this option to include all <b>Classes</b> , <b>Attributes</b> and <b>Association End</b> roles in the GML Application Schema, irrespective of their visibility (or Scope). Otherwise, only those with <b>Public</b> visibility will be considered for generation.	(Element) <a href="#">Details</a> <sup>[959]</sup> <a href="#">General Properties of Attributes</a> <sup>[100]</sup> <a href="#">Source Role</a> <sup>[1130]</sup>										
Stylesheet	(Optional) Select an XSL stylesheet to post-process the Schema before saving it to file.  Import the required stylesheet into the project through the Resources window before selecting it in this field.	<a href="#">Resources</a> <sup>[1173]</sup>										
Application Schema(s)	Lists all the Application Schema packages under the selected source package.  Click on a package to display its details, as well as any of its child «Leaf» stereotyped package details, in the Package Details panel.  GML Application Schema will not be generated for any package in the list that is unchecked.											
Package Details	Against each of the child «Application Schema» and (optionally) «Leaf» stereotyped packages (if any), enter a filename. To do this, double-click on the package name in the list and, on the Package Details dialog, review or complete the fields: <table><tr><th>Field/Option/Button</th><th>Action</th></tr><tr><td>Package</td><td>The selected package name.</td></tr><tr><td>Stereotype</td><td>The stereotype of the selected package.</td></tr><tr><td>Filename</td><td>Optional for a «Leaf» stereotyped package.  Click on the [ ... ] button and type the name of the .x GML Application Schema is to be generated.  The filename must be in the same file path as selected field on the Generate GML Application Schema dialog.</td></tr><tr><td>Namespace</td><td>Not required for a «Leaf» stereotyped package.</td></tr></table>	Field/Option/Button	Action	Package	The selected package name.	Stereotype	The stereotype of the selected package.	Filename	Optional for a «Leaf» stereotyped package.  Click on the [ ... ] button and type the name of the .x GML Application Schema is to be generated.  The filename must be in the same file path as selected field on the Generate GML Application Schema dialog.	Namespace	Not required for a «Leaf» stereotyped package.	
Field/Option/Button	Action											
Package	The selected package name.											
Stereotype	The stereotype of the selected package.											
Filename	Optional for a «Leaf» stereotyped package.  Click on the [ ... ] button and type the name of the .x GML Application Schema is to be generated.  The filename must be in the same file path as selected field on the Generate GML Application Schema dialog.											
Namespace	Not required for a «Leaf» stereotyped package.											

Field/Option/ Button	Action	See also						
	<table><tr><th>Field/Option/Button</th><th>Action</th></tr><tr><td></td><td>Type in the namespace for the selected package.</td></tr><tr><td><b>Namespace Prefix</b></td><td>Not required for a «Leaf» stereotyped package. Type the abbreviated value that represents the Nar</td></tr></table>	Field/Option/Button	Action		Type in the namespace for the selected package.	<b>Namespace Prefix</b>	Not required for a «Leaf» stereotyped package. Type the abbreviated value that represents the Nar	
	Field/Option/Button	Action						
		Type in the namespace for the selected package.						
	<b>Namespace Prefix</b>	Not required for a «Leaf» stereotyped package. Type the abbreviated value that represents the Nar						
	<ul style="list-style-type: none"><li>• The value of the <b>Filename</b> field will be saved as the value of the tag <b>xsdDocument</b> on the package</li><li>• The value of the <b>Namespace</b> field will be saved as the value of the tag <b>targetNamespace</b> on the package</li><li>• The value of the field <b>Namespace Prefix</b> will be saved as the value of the tag <b>xmlns</b> on the package</li></ul>							
<b>Generate</b>	Generate the GML Application Schema for each of the checked packages in the Package Details panel.							
<b>View Schema</b>	Display the generated GML Application Schema for the selected package in the Package Details panel.							
<b>Close</b>	Close the Generate GML Application Schema dialog.							
<b>Help</b>	Display this Help topic.							

### Notes

- The MDG Technology for GML is available in the Professional, Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect
- A «CodeList» stereotyped Class with the tag **asDictionary** set to **true** will be exported as a separate XML file to the directory specified in the **Directory** field, with the filename *CodeList\_<classname>.xml* (where <classname> is the name of the «CodeList» stereotyped Class)
- Using the configurable file **GMLStereotypes.xml** in the Sparx Systems > EA > Config > GML folder, you can specify aliases for the standard GML stereotypes; the GML Application Schema Generator will also consider these aliases during Schema generation
- Association Class** is specific to **GML 3.3**; when generating **GML 3.2.1** Application Schema, the Association and Class of an Association Class are treated as separate entities

### Learn more

- [MDG Technology For GML](#) 
- [The Open Geospatial Consortium GML Specification](#) (Online Resource)









## 9.8 User Interface Models

**User Interface** diagrams are **customized** UML diagrams on which you can visually represent a system's user interface using forms, controls and labels.

**Example Diagram** [Example User Interface Diagram](#) 

### Tools

Select User Interface diagram elements and connectors from the **User Interface - Simple** pages of the Toolbox.

User Interface Diagram Elements	User Interface Diagram Connectors
 Package	 Associate
 Screen	 Aggregate
 UI Control	 Generalize
 Object	 Realize

### Notes

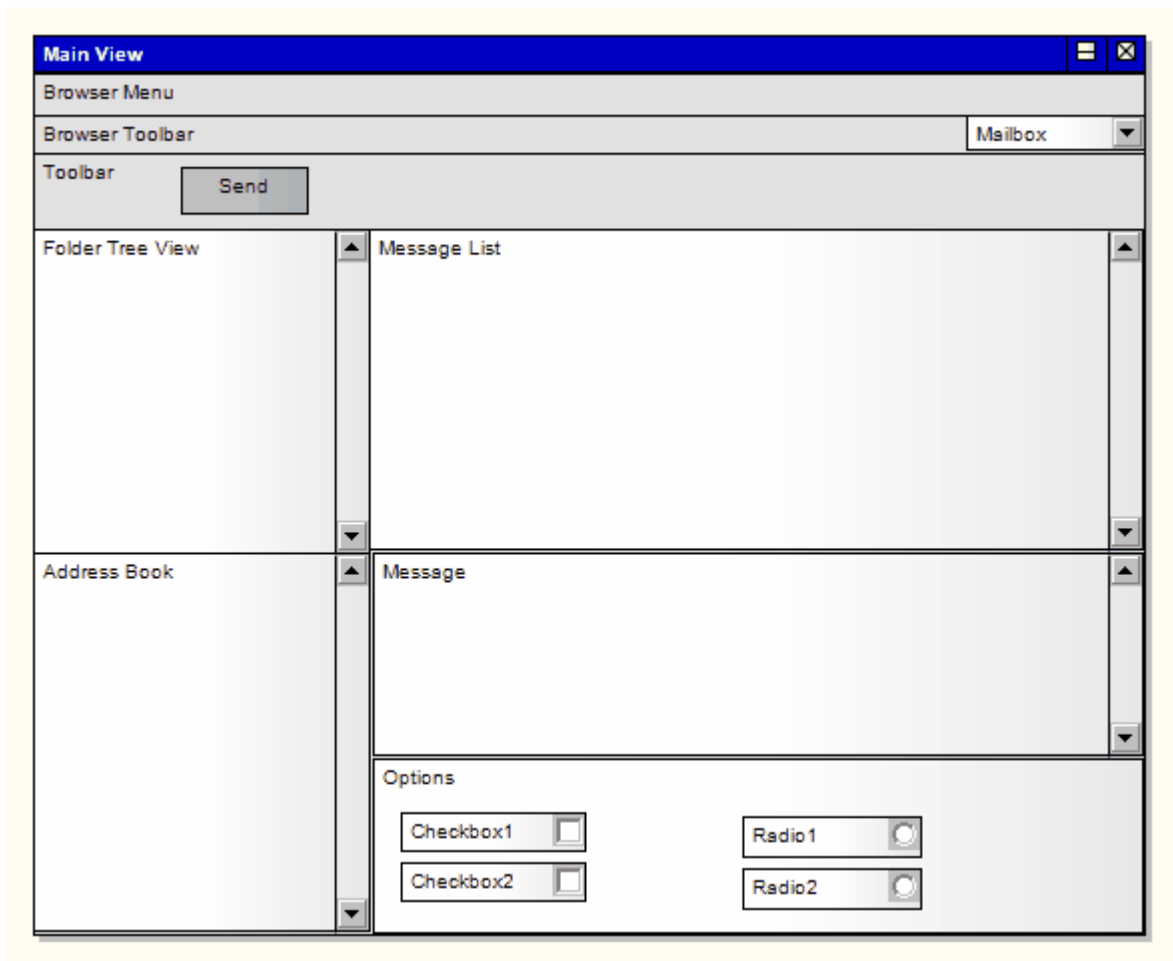
- Using stereotyped Classes, you can model the design of a web page user interface
- The Enterprise Architect Professional, Corporate and Suite editions also include the **MDG Win32 UI Technology**, with which you can design user interface components that render more precisely as Win32® User Interface elements

### Learn more

- [Web Stereotypes](#) 
- [Win32 UI Technology](#) 

### 9.8.1 Example User Interface Diagram

In this example User Interface diagram, forms, controls and labels are arranged on the diagram to define the appearance of a user interface screen and controls. UI Control elements can also be traced to other model elements linking the UI with the underlying implementation.



#### Learn more

- [Screen](#) [1991]
- [UI Control Elements](#) [1992]

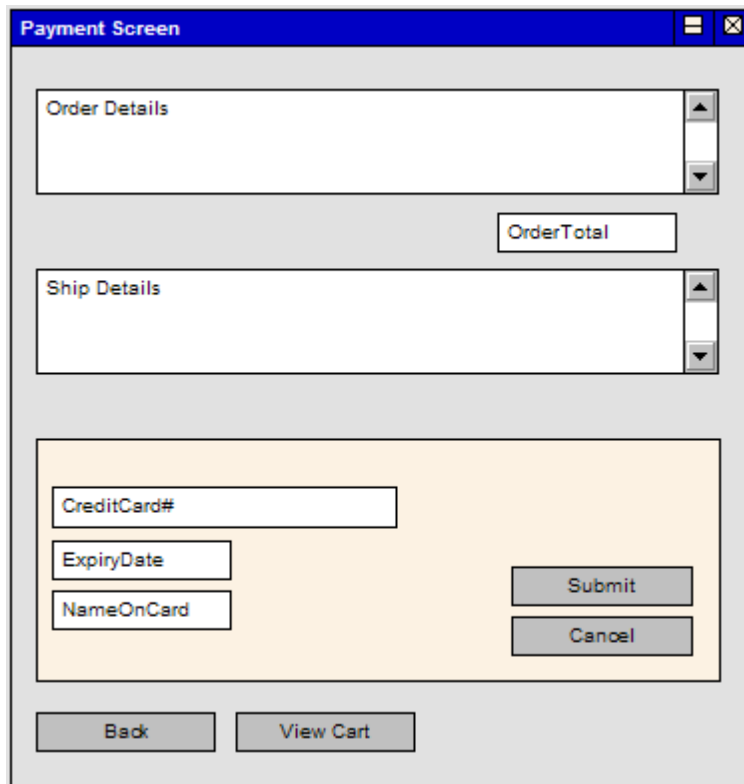
### 9.8.2 Screen

A **Screen** is used to prototype User Interface screen flow. By using UML features such as requirements, constraints and scenarios against User Interface diagram elements, you can build up a solid and detailed understanding of user interface behavior without having to use code. This provides an excellent means of establishing the precise behavior of the system from a user's perspective, and in conjunction with the Use Case model, defines exactly how a user gets work done.

Web pages can also be prototyped and specified rigorously using Enterprise Architect's custom interface extensions.

#### Example

This example diagram illustrates some features of Enterprise Architect's screen modeling extensions that support web page prototyping. By adding requirements, rules, scenarios and notes to each element, a detailed model is built up of the form or web page, without having to resort to GUI builders or HTML.



Enterprise Architect displays UI Controls as a range of special icons, depending on the stereotype used; for example, a Control stereotyped as a «list» displays with a vertical scroll bar.

#### Toolbox icon



#### Learn more

- [User Interface Models](#) <sup>[1990]</sup>
- [UI Controls](#) <sup>[1992]</sup>

### 9.8.3 UI Control Elements

A **UI Control** element represents a user interface control element (such as an edit box). It is used for capturing the components of a screen layout and requirements in a Custom or User Interface diagram.

There are a number of UI Control elements available in the User Interface page of the Toolbox. These include:

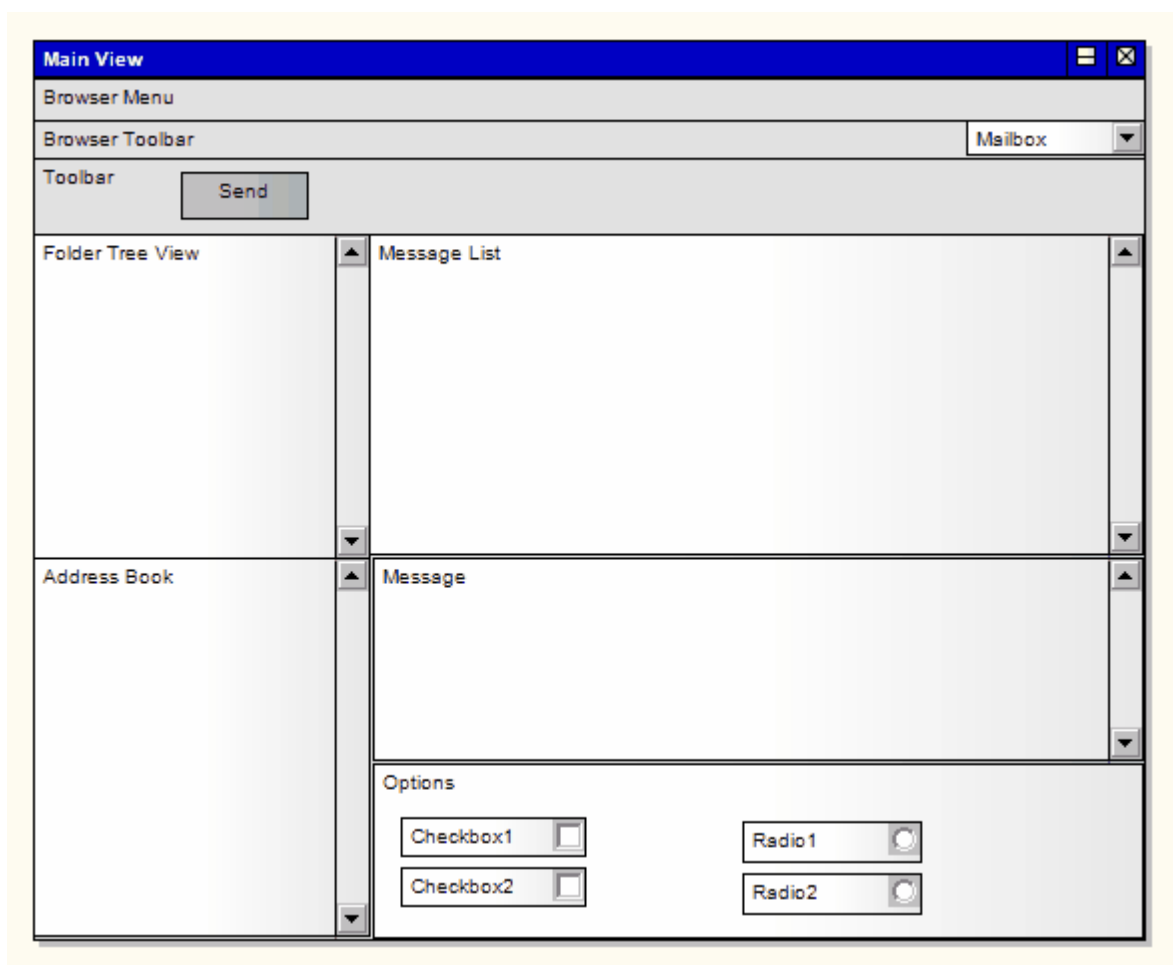
- List
- Table



- Text Box
- Label
- Form
- Panel
- Button
- Combobox
- Checkbox
- Checkbox (left hand side)
- Radio button
- Radio button (left hand side)
- Vertical Line
- Horizontal Line

### Example

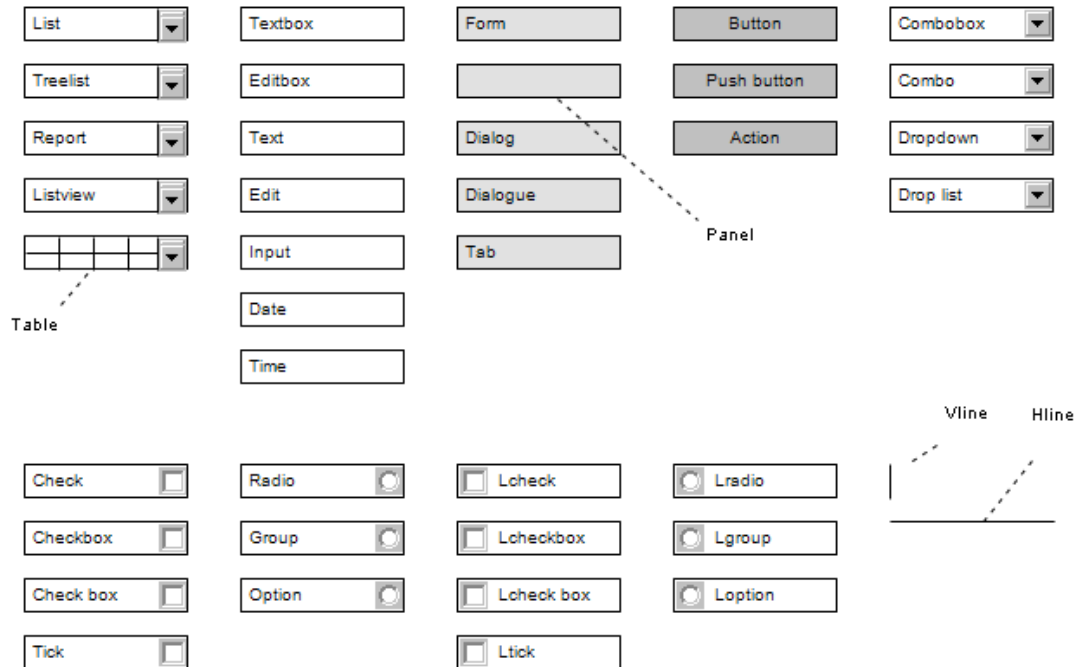
The icons can be combined on a Screen icon to represent the appearance of a user interface screen, as shown:



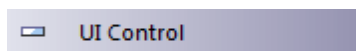
You can also extend the available icons by selecting other stereotypes in the UI Control Element Properties

dialog. The full set of available stereotypes is shown below; type or select the text in the **Stereotype** field to create the corresponding icon.

#### ui User Interface



#### Toolbox icon



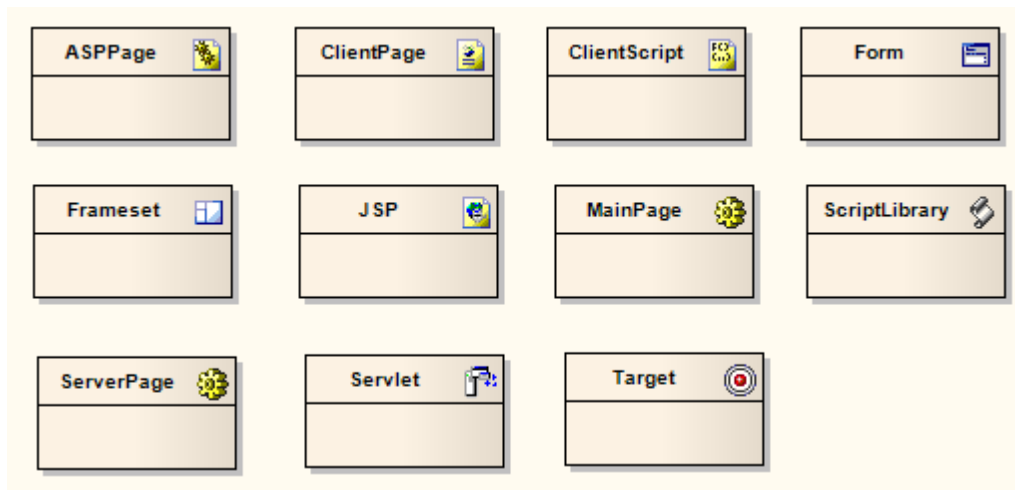
(where **UI Control** is the name of the user interface element type)

#### Learn more

- [User Interface Models](#)<sup>[1990]</sup>
- [Screen](#)<sup>[1991]</sup>

### 9.8.4 Web Stereotypes

Enterprise Architect supports a number of stereotypes for web page modeling, the graphical elements for which display with a **graphical icon** instead of the usual «*stereotype*» format. These stereotypes are only supported for Class elements. These are the various graphical icons and their associated stereotypes:



A similar set of web modeling elements and their relationships are also available through dedicated Web Modeling pages in the Diagram Toolbox.

#### To set a web icon

Step	Action
1	Create a new Class element in a diagram.
2	Display the Class Properties dialog.
3	In the <b>Stereotype</b> field, either type in the required stereotype name or click on the drop-down arrow and select the required stereotype (as named above).
4	Click on the <b>OK</b> button. The Class displays as in one of the examples shown.

#### Learn more

- [User Interface Models](#)<sup>[1990]</sup>
- [Diagram Toolbox](#)<sup>[792]</sup>

### 9.8.5 Win32 UI Technology

Using the **MDG Win32 UI Technology**, you can design user interface screens that render as Win32® user interface elements, so that you can create and maintain User Interface diagrams in the Win32® User Interface format.

**Access** The Win32 UI facilities are provided in the form of:

- A Win32 User Interface diagram type, accessed through the New Diagram dialog
- Win32 User Interface pages in the Toolbox
- Win32 User Interface element and relationship entries in the Toolbox Shortcut menu and Quick Linker

#### Win32 User Interface Toolbox Pages

You can access the Win32 User Interface Toolbox pages through the **Diagram | Diagram Toolbox: More tools... User Interface - Win32** option.

You can also set Win32® User Interface Modeling as the active default technology to access the Toolbox pages directly.

#### Disable Win32 User Interface Technology

If you prefer not to use the Win32® UI Technology in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog (**Settings | MDG Technologies**).

#### Notes

- The MDG Win32® User Interface Technology is available in the Enterprise Architect Professional, Corporate and Suite editions
- When you initially drag the **Combo Box** icon onto a diagram, of **Type Drop Down** or **Drop Down List**, the middle 'tracking handle' on each side of the element is white, indicating that you can only adjust the **width** of the element

To adjust the **height** of the element as well as the width, click on the drop-down arrow part of the image; the middle 'tracking handle' on the bottom edge is now white, indicating that you can drag the base down to set the **virtual** height (the height of the element when it is expanded to show all possible values in the drop-down list)

#### Learn more

- [User Interface Models](#) <sup>[1990]</sup>
- [Manage MDG Technologies](#) <sup>[1477]</sup>

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Modeling Languages | Standard Profiles | Win32 User Interface**
- (Alt+F1) | **Enterprise Architect | Software Engineering | User Interface | Win32 Dialog Modeling**

## 9.9 Other Stereotypes

There are many other UML elements that you can also work with in Enterprise Architect, most of which are basic elements extended by the use of stereotypes. This topic gives a brief introduction to some of these elements.

- [Analysis Stereotypes](#) <sup>[1800]</sup>
- [Boundary](#) <sup>[1997]</sup>
- [Composite Elements](#) <sup>[936]</sup>
- [Control](#) <sup>[1999]</sup>
- [Entity](#) <sup>[2000]</sup>
- [Event](#) <sup>[2001]</sup>
- [Feature](#) <sup>[1769]</sup>
- [Hyperlinks](#) <sup>[2002]</sup>
- [Image](#) <sup>[2006]</sup>
- [N-Ary Association](#) <sup>[2007]</sup>
- [Packaging Component](#) <sup>[2008]</sup>
- [Process](#) <sup>[2008]</sup>
- [Requirement](#) <sup>[1763]</sup>
- [Risk](#) <sup>[2009]</sup>
- [Screen](#) <sup>[1991]</sup>
- [Task](#) <sup>[2010]</sup>
- [Test Case](#) <sup>[2010]</sup>
- [Table](#) <sup>[1942]</sup>
- [UI Control Elements](#) <sup>[1992]</sup>
- [Web Stereotypes](#) <sup>[1995]</sup>

### Learn more

- [UML Stereotypes](#) <sup>[1452]</sup>

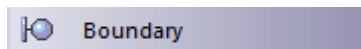
### 9.9.1 Boundary



#### Description

A Boundary is a stereotyped Object that models some system boundary, typically a user interface screen. You can also create a Boundary as a stereotyped Class. Boundary elements are used in analysis to capture user interactions, screen flows and element interactions (or 'collaborations').

A Boundary is used in the conceptual phase to capture users interacting with the system at a screen level (or some other boundary interface type). It is often used in Sequence and Robustness (Analysis) diagrams. It is the View in the Model-View-Controller pattern.

Toolbox iconLearn more

- [Create a Boundary](#) <sup>[1998]</sup>
- [Analysis Stereotypes](#) <sup>[1800]</sup>
- [Sequence Diagram](#) <sup>[1249]</sup>
- [Communication Diagram](#) <sup>[1259]</sup>
- [Object Diagram](#) <sup>[1186]</sup>
- [Analysis Diagram](#) <sup>[1801]</sup>

**9.9.1.1 Create a Boundary**Create a Boundary element on a diagram as an Object

Step	Action
1	In the Toolbox, select the <b>More tools   Extended   Analysis</b> menu option.
2	From the Analysis Elements page, drag the Boundary element onto the diagram.

Create a Boundary element as a stereotyped Class

Step	Action
1	Insert a new Class.
2	Right-click on the element and select the <b>Properties</b> context menu option; the Properties dialog displays.
3	In the <b>Stereotype</b> field, type the value <b>boundary</b> .
4	Click on the <b>Apply</b> and <b>OK</b> buttons.
5	Save the diagram ( <b>Ctrl+S</b> ).

## 9.9.2 Control



### Description

A Control is a stereotyped Object that models a controlling entity or manager. A Control organizes and schedules other activities and elements, typically in Analysis (including Robustness), Sequence and Communication diagrams. It is the controller of the Model-View-Controller pattern.

You can also create a Control as a stereotyped Class.

### Toolbox icon



### Learn more

- [Create a Control Element](#) <sup>[1999]</sup>
- [Analysis Stereotypes](#) <sup>[1800]</sup>
  - [Sequence Diagram](#) <sup>[1249]</sup>
  - [Communication Diagram](#) <sup>[1259]</sup>
  - [Object Diagram](#) <sup>[1186]</sup>
  - [Analysis Diagram](#) <sup>[1801]</sup>

### 9.9.2.1 Create a Control Element

#### How to

To create a Control element on a diagram as an Object, follow the steps below

Step	Action
1	In the Toolbox, select the <b>More tools   Analysis</b> menu option.
2	From the Analysis Elements page, drag the Control element onto the diagram.

To create a Control element as a stereotyped Class, using the Class Properties dialog, follow the steps below

Step	Action
1	Insert a new Class.
2	Right-click on the element and select the <b>Properties</b> context menu option. The Properties dialog displays.
3	In the <b>Stereotype</b> field, type the value <b>control</b> .
4	Click on the <b>Apply</b> and <b>OK</b> buttons.
5	Save the diagram ( <b>Ctrl+S</b> ).

### 9.9.3 Entity



#### Description

An Entity is a stereotyped Object that models a store or persistence mechanism that captures the information or knowledge in a system. It is the Model in the Model-View-Controller pattern.

You can also create an Entity as a stereotyped Class. See the *Create an Entity* topic.

#### Toolbox icon



#### Learn more

- [Create an Entity](#) <sup>[2001]</sup>
- [Analysis Stereotypes](#) <sup>[1800]</sup>
- [Sequence Diagram](#) <sup>[1249]</sup>
- [Communication Diagram](#) <sup>[1259]</sup>
- [Object Diagram](#) <sup>[1186]</sup>
- [Analysis Diagram](#) <sup>[1801]</sup>



### 9.9.3.1 Create an Entity

#### How to

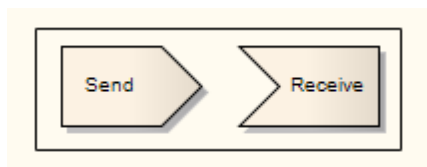
To create an Entity element on a diagram as an Object

Step	Action
1	In the Toolbox, select the <b>More tools   Analysis</b> menu option.
2	From the Analysis Elements page, drag the <i>Entity</i> element onto the diagram.

To create an Entity element as a stereotyped Class, using the Class Properties dialog

Step	Action
1	Insert a new Class.
2	Right-click on the element and select the <b>Properties</b> context menu option; the Properties dialog displays.
3	In the <b>Stereotype</b> field, type the value <b>entity</b> .
4	Click on the <b>Apply</b> and <b>OK</b> buttons.
5	Save the diagram ( <b>Ctrl+S</b> ).

### 9.9.4 Event



#### Description

Two elements are used to model *Events*; the:

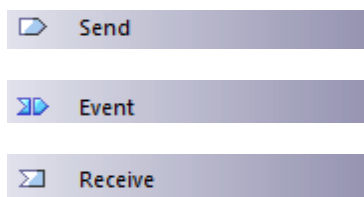
- *Send Event* which models the generation of a stimulus in the system and the passing of that stimulus to other elements, either within the system or external to the system

- *Receive Event*, depicted as a rectangle with a recessed 'V' on the left side, which indicates that an event occurs in the system due to some external or internal stimulus; typically this invokes further activities and processing

Send and Receive Events can be added from the Analysis, State and Activity Element pages of the Toolbox.

If you should select the wrong type of event, or otherwise want to change the type, right-click on the Event and select the **Advanced | Make Sender** or **Advanced | Make Receiver** context menu option, as appropriate.

#### Toolbox icon



#### Learn more

- [Analysis Toolbox](#)<sup>[812]</sup>
- [State Toolbox](#)<sup>[807]</sup>
- [Activity Toolbox](#)<sup>[808]</sup>

## 9.9.5 Hyperlinks

### Description

You can place a *Hyperlink* element onto a diagram. This element is a type of text element, but one that can contain a pointer to a range of objects such as associated document files, web pages, Help, model features and even other Enterprise Architect model files. When you double-click on the element, Enterprise Architect executes the link.

To add a Hyperlink element, drag the Hyperlink icon from the Common page of the Toolbox onto the diagram.

(Alternatively, click on the **Hyperlink** icon in the UML Elements toolbar and then click on the diagram.)



### Configure the Hyperlink

When you add the Hyperlink to the diagram, the Hyperlink Details dialog displays. If you want to display the information in a more readable layout, you can resize the dialog.

You first select the type of object to link to; click on the drop-down arrow in the **Type** field and select a type. The Hyperlink Details dialog then displays the appropriate fields, prompts or dialog to enable you to specify the object to link to. For example, if you intend to hyperlink to:

- An attribute, the Set Attribute dialog displays to enable you to select that attribute

- A file, the **Action** field displays to enable you to specify whether to Open the file in read only mode, or Edit the file; in either case the file is opened within Enterprise Architect if possible or, if not possible, with the Windows default viewer/editor for the file type

For example, if you hyperlink to a .rtf file, you can *view* it in whichever viewer is appropriate; however, you cannot *edit* .rtf files in Enterprise Architect, so the file always opens in the *Windows* default .rtf editor

- A diagram, the Select a Diagram dialog displays, which enables you to select the diagram from anywhere in the project; you can filter the selection to diagrams of certain types

In most cases you define the location of the hyperlinked object in the **Address** field, either by overtyping the field or by clicking on the ( ... ) (Browse) button. You can use full paths or local (path substitution) paths.

If you select **EA Command** as the link type, the **Address** field is replaced by a drop-down list of Enterprise Architect commands. You can select **LocalPath** and click on the ( ... ) (Browse) button to display the Local Paths dialog, which you complete as required. Subsequently, when you click on the hyperlink the Local Paths dialog immediately displays and you can apply, switch, expand or update the current path.

If you select to link to a **Learning Center** topic, you define the link through two drop-down lists; the first to specify the Learning Center **category** and the second to specify the Learning Center **page** or **topic**.

In the **Alias** field, type the text to display in the hyperlink. If you do not provide an alias, either the text defaults to the link itself, or (for certain link targets such as a matrix profile) the dialog generates a simple text instruction.

If you prefer to display only the hyperlink text, without the icon, select the **Hide Icon** checkbox.

### Notes

- If required, you can create a number of empty hyperlinks to complete later; if you then double-click on an empty hyperlink, the Hyperlink Details dialog displays and you can enter the details
- Once you have created the hyperlink, you can also edit the hyperlink text by clicking once on the field and once on the text, then right-clicking and selecting the **Edit Selected** context menu option
- You can add notes to the hyperlink, which display in the Hyperlink Details dialog when you right-click on the hyperlink and select the **Properties** context menu option; you can format these notes using the Notes toolbar

### Learn more

- [Hyperlinks to Files](#)<sup>[2004]</sup>
- [Hyperlinks to Scripts](#)<sup>[2004]</sup>
- [Add Action as a Hyperlink](#)<sup>[2004]</sup>
- [Hyperlinks between Diagrams](#)<sup>[2005]</sup>
- [Notes Toolbar](#)<sup>[1143]</sup> (to create hyperlinks in Notes text)
- [Hyperlink From Linked Document](#)<sup>[1095]</sup>
- [Insert Reference Links](#)<sup>[1083]</sup> (to create hyperlinks in document Reports)
- [Add Object Links](#)<sup>[359]</sup> (to create hyperlinks in Team Review posts)

### 9.9.5.1 Hyperlinks To Files

To create a hyperlink on a diagram to an external file, simply click on the file in a file list (such as Windows Explorer) or on your Desktop and drag it onto the diagram.

A short context menu displays with two options - **Hyperlink** and **Artifact**. Click on the **Hyperlink** option to create the hyperlink on the diagram.


The link is effective immediately, and you can right-click on it to add or change properties as necessary.

Files of most types - including .sql and .ddl - are opened within the appropriate Enterprise Architect code editor.

#### Learn more


- [Hyperlinks](#) 

### 9.9.5.2 Script Hyperlinks

You can create a hyperlink on a diagram to execute a script. Simply drag the required script from the [Execution Analyzer](#)  window onto the diagram. A context menu displays, from which you select whether the script to be executed is a Build, Test, Run, Debug or Deploy script. The hyperlink is effective immediately; when you click on it, the script executes.

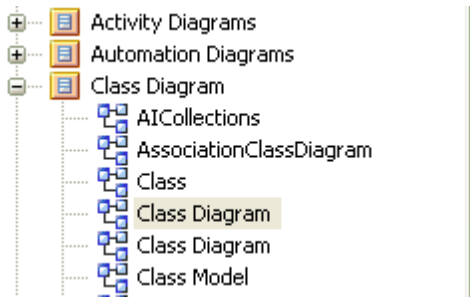
### 9.9.5.3 Add Action As Hyperlink

#### Create an Action element to represent a hyperlink

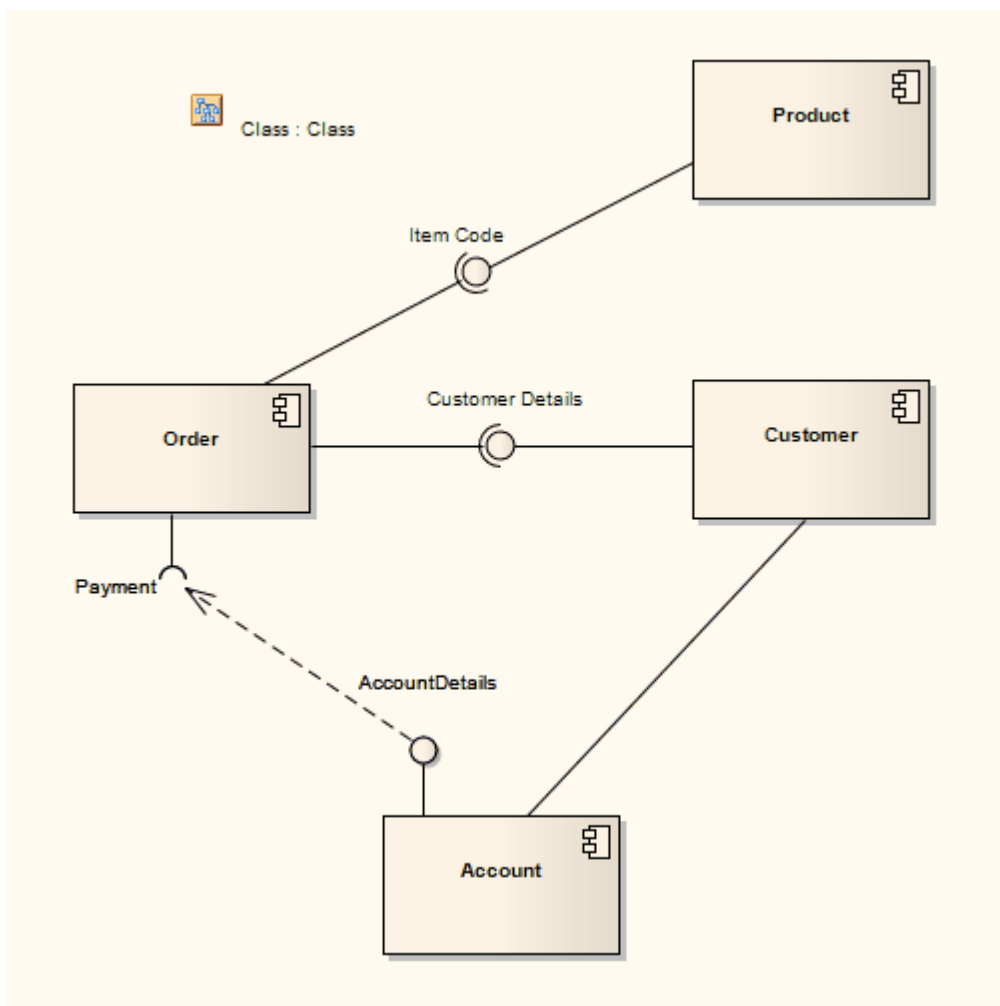
Step	Action
1	Drag an Action element from the Activity page of the Toolbox onto the diagram. A context menu immediately displays.
2	Select the <b>Other</b> menu option. The New Action dialog displays, with the <b>Other</b> radio button selected.
3	Click on the drop down arrow on the field in the Select Kind panel, and click on the <b>Hyperlink</b> option.
4	Click on the <b>OK</b> button. The <i>Hyperlink Action</i> element displays on the diagram.
5	Right click on the element and select the <b>Advanced   Set Hyperlink</b> menu option. The Hyperlink Details dialog displays.
6	<a href="#">Set the hyperlink's properties</a> 

### 9.9.5.4 Hyperlinks Between Diagrams

#### Create a hyperlink between diagrams

Step	Action
1	<p>Open the diagram in which to display the hyperlink to another diagram.</p> <p>From the Project Browser select the diagram you want to create a hyperlink to:</p> 
2	<p>Drag the diagram on to the current diagram.</p> <p>The Select Type dialog displays.</p>
3	<p>Select the <b>Hyperlink</b> option and click on the <b>OK</b> button.</p> <p>The final hyperlinked diagram should resemble the diagram below, where the Class diagram is the diagram to which the Product Order diagram hyperlinks (notice that the hyperlink icon is different).</p>

#### Example



#### Notes

- If the hyperlink appears as a Sub Activity, select the **Tools | Options | Diagram | Behavior** menu option and deselect the **Use Automatic SubActivities** checkbox

### 9.9.6 Image



#### Description

An *Image* is a System Boundary element that automatically displays first the Boundary Properties dialog and then the Select Alternate Image dialog to change its representation to an imported image. You can use it as

an icon for an element or group of elements, or as a diagram background.

Image elements are available from the Common page of the Toolbox.

#### Toolbox icon



#### Learn more

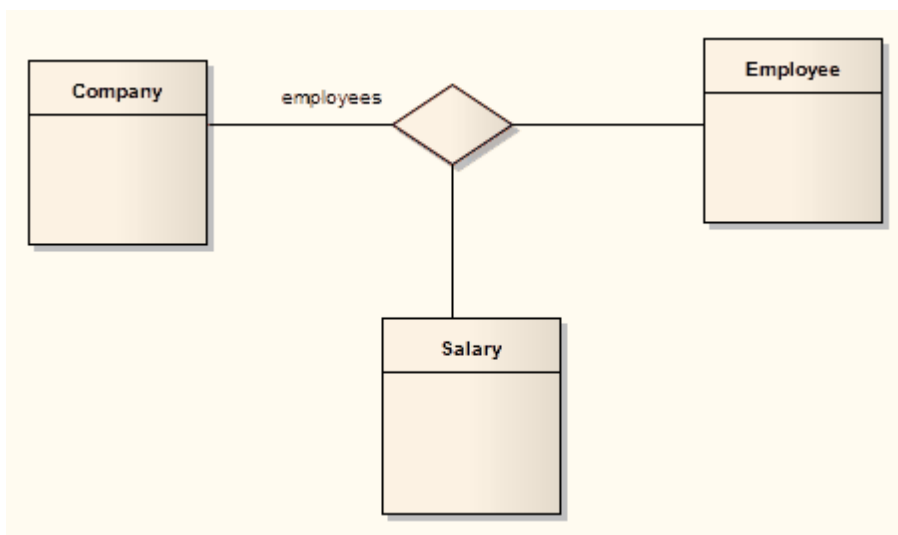
- [System Boundary](#)<sup>[1347]</sup>

### 9.9.7 N-Ary Association



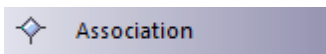
#### Description

An *n-Ary Association* element is used to model complex relationships between three or more elements, typically in a Class diagram. It is not a commonly-employed device, but can be used to good effect where there is a dependant relationship between several elements. It is generally used with the Association connector, but the relationships can include other types of connector.



In the example above there is a relationship between a *Company*, an *Employee* and a *Salary*.

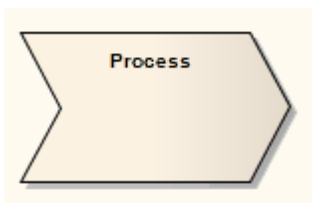
#### Toolbox icon



#### Learn more

- [Class Diagram](#) [1184]
- [Association](#) [1393]

### 9.9.8 Process



#### Description

A Process is an Activity element with the stereotype *process*, which expresses the concept of a business process. Typically this involves inputs, outputs, work flows, goals and connections with other Processes. The Process element is typically used in Analysis diagrams.

Business processes typically range across many parts of the organization and span one or more systems.

#### Toolbox icon



#### Learn more

- [Analysis diagram](#) [1801]

### 9.9.9 Packaging Component

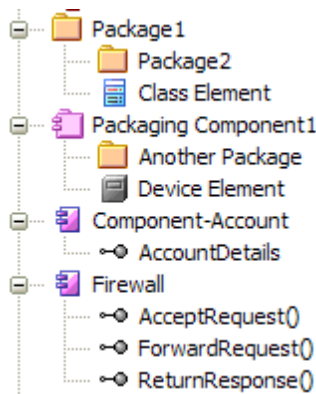


#### Description

A *Packaging Component* is an element that appears very similar to a Component in a diagram but behaves as a Package in the Project Browser (that is, it can be version controlled and can contain other Packages and elements). It is typically used in Component diagrams.

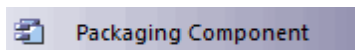
In the Project Browser, the three elements display as shown below:





The Component element cannot contain child Packages or Packaging Components.

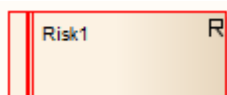
#### Toolbox icon



#### Learn more

- [Component diagrams](#)<sup>[1194]</sup>
- [Component](#)<sup>[1370]</sup>
- [Package](#)<sup>[1382]</sup>

### 9.9.10 Risk



#### Description

A Risk is defined as the effect of uncertainty on objectives. In Project Management, it is necessary to try to identify risks and assess:

- The likelihood that they have a negative effect on a project and
- How large that effect is likely to be

Those risks with a high probability of occurrence and/or a large impact on the project can be mitigated.

A Risk Management process might consist of the following five steps:

1. Identify risks and represent each with a Risk element.
2. Identify which elements (such as Components, Use Cases or Features) are vulnerable to each risk; you might decide to create «trace» dependencies from these elements to the Risk elements.
3. Assess the likelihood and magnitude of the risks.
4. Identify ways to mitigate the risks.

5. Prioritize the risk reduction measures based on their likelihood, magnitude and ease of mitigation.

Risk *elements* are not the same as the risks that you assign *to* an element through the Project Management window. Such risks are internal to the selected element, whilst a Risk element can be associated with a number of elements, either in a logical group or totally separate.

Risk elements are available from the Requirements page of the Toolbox.

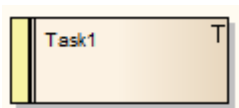
#### Notes

- Risk elements can be displayed with or without an identifying **R** in the top right corner of the element; to toggle the display of this letter, select or deselect the **Show stereotype icon for requirements** checkbox on the Options dialog, Objects page

#### Toolbox icon



### 9.9.11 Task



#### Description

A Task element represents a task that must be performed in relation to an element. The Task element enables you to assign resources to the task itself, rather than just to the parent element.

You can create a hierarchy or tree structure of Task elements to break a large task into separate parts and assign different resources to each part.

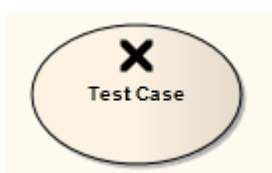
#### Toolbox icon



#### Learn more

- [Resource Management](#)<sup>512</sup>

### 9.9.12 Test Case



### Description

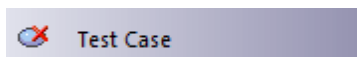
A *Test Case* is a stereotyped Use Case element. You might use it to extend the facilities of the Testing window, by applying element properties and capabilities to the tests of a feature represented by another element or - more appropriately - set of elements. That is, you can define in one go - in the Testing window for the Test Case element - the details of the tests that apply to each of several elements, instead of recording the details separately in each element.

Within the Test Case element properties you can define test requirements and constraints, and associate the test with test files. You can also link the element to Document Artifacts or (in the Corporate, Business and Software Engineering, System Engineering and Ultimate editions) directly to linked documents, such as a Test Plan.

The Test Case element enables you to give greater visibility to tests, in the Project Browser, Diagram List, Package Browser, Model Search, Relationship Matrix, Traceability window and reports.

The Test Case element is available through the Use Case and Maintenance pages of the Toolbox.

### Toolbox icon



### Learn more

- [Working on Test Records](#) 

# Part

---



## 10 Model Transformation

**Model Driven Architecture (MDA) Transformations** provide a fully configurable method of converting model elements and model fragments from one domain to another. This typically involves converting **Platform-Independent Model (PIM)** elements to **Platform-Specific Model (PSM)** elements. You can transform a single element from the PIM into **multiple** PSM elements across multiple domains.

Transformations are a huge productivity boost, and reduce the necessity of manually implementing stock Classes and elements for a particular implementation domain; for example, database tables generated from persistent PIM Classes. For a further productivity boost, Enterprise Architect can automatically generate code for your transformed Classes that target code languages.

### Ready-built Transformations

The Enterprise Architect installer includes a number of basic **built-in** transformations, including:

- PIM to:
  - C#
  - C++
  - DDL table elements
  - EJB Entity Bean
  - EJB Session Bean
  - Java
  - PHP
  - VB.Net
  - XSD
- Data Model to Entity Relationship Diagram (ERD)
- Entity Relationship Diagram (ERD) to Data Model
- Sequence diagram to Communication diagram
- Communication diagram to Sequence diagram
- Java Model to JUnit test model
- .NET model to NUnit test model
- WSDL interface model to WSDL

Further transformations will become available over time, either built in or as downloadable modules from the Sparx Systems website.

### Customized Transformations

You can **modify** the built-in transformations or **define your own**, using Enterprise Architect's simple code generation template language. This involves little more than writing templates to create a simple **intermediary source** file; the system reads the source file and binds that to the new PSM.

### Transformation Dependencies

When you execute a transformation, the system creates internal bindings (**Transformation Dependencies**) between each PSM created and the original PIM. This is essential, providing the ability to forward

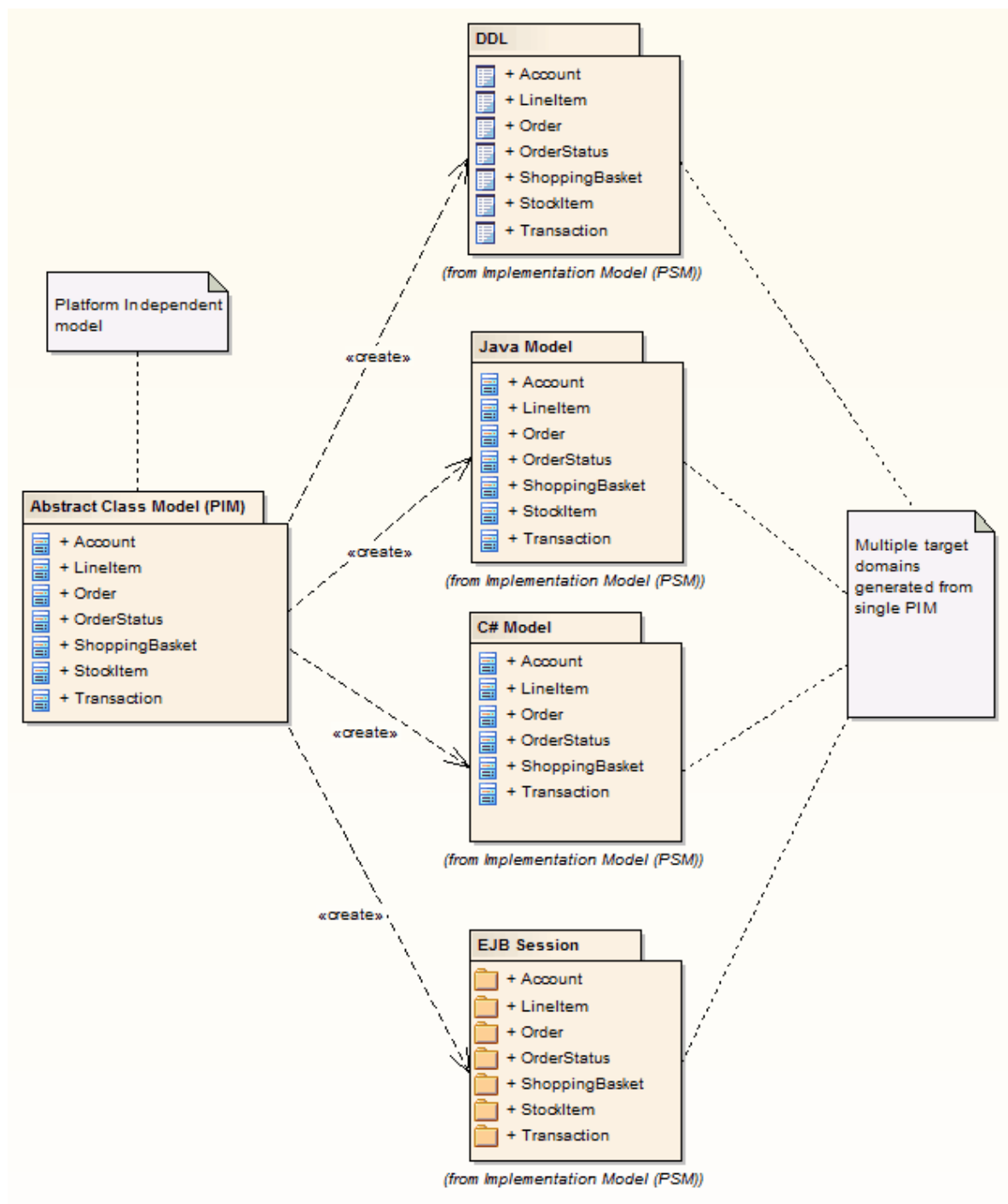
synchronize from the PIM to the PSM many times, adding or deleting features as you go; for example, adding a new attribute to a PIM Class can be forward synchronized to a new column in the Data Model.

You can observe the Transformation Dependencies for a package using the Traceability window, to check the impact of changes to a PIM element on the corresponding elements in each generated PSM, or to verify where a change required in a PSM should be initiated in the PIM (and also to reflect back in other PSMs). The Transformation Dependencies are a valuable tool in managing the traceability of your models.

Enterprise Architect does not delete or overwrite any element features that were not originally generated by the transform; therefore, you can add **new** methods to your elements, and Enterprise Architect does not act on them during the forward generation process.

### **Example of a Transformation**

This diagram highlights how transformations work and how they can significantly boost your productivity.



### Notes

- If you are using the Corporate, Business and Software Engineering, System Engineering or Ultimate edition, if security is enabled you must have **Transform Package** access permission to perform an MDA Transformation on the elements of a package

### Learn more

- [Transform Elements](#) <sup>2017</sup>

- [Built-in Transformations](#) <sup>[2020]</sup>
- [Edit Transformation Templates](#) <sup>[2048]</sup>
- [Write Transformations](#) <sup>[2051]</sup>
- [Traceability](#) <sup>[723]</sup>
- [Import Code Generation and Transformation Templates](#) <sup>[1637]</sup>
- [Export Code Generation and Transformation Templates](#) <sup>[1637]</sup>
- [Permission List](#) <sup>[329]</sup>

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Enterprise Architect | Model Transformations**



## 10.1 Transform Elements

When you perform a Model Transformation of elements of any kind, you can initiate the transformation either from a diagram or from a Package.

### Access

Transform selected elements on a diagram:

**Tools | Model Transformation (MDA) | Transform Selected Elements**, or  
**On diagram | Class context menu | Advanced | Transform (Ctrl+H)**

Transform all elements in the Package currently selected in the Project Browser:

**Tools | Model Transformation (MDA) | Transform Current Package**, or  
**Project Browser package context menu | Advanced | Transform Package (Ctrl+Shift+H)**

### Perform a Transformation

Field/Option/Button	Action	See also
<b>Elements</b>	Lists all of the individual elements selected in the diagram or held in the Package. Either: <ul style="list-style-type: none"> <li>Click on an element to include just that element in the transformation</li> <li>Hold <b>Ctrl</b> and click on several separate elements to include them in the transformation, or</li> <li>Hold <b>Shift</b> and click on the first and last elements in a block to include those elements in the transformation</li> </ul>	
<b>All</b>	Click on this button to select all of the elements in the list to include them in the transformation.	
<b>None</b>	Click on this button to deselect all of the elements in the list.	
<b>Include child packages</b>	(If you have selected to transform elements in a <b>Package</b> .) Select this checkbox to include (in the <b>Elements</b> list and potentially in the transformation) elements from the child Packages of the selected Package.	
<b>Transformations</b>	Select the checkbox against each type of transformation to perform. When you select a checkbox, the Browse Project dialog displays; locate and select the target Package into which to generate the transformed elements.  If you want to change a selected target Package, click on the ( ... ) button to the right of the Package name and select the new Package	

Field/Option/ Button	Action	See also
	from the dialog.	
<b>Generate Code on result</b>	Select this checkbox to specify whether or not to <b>automatically</b> generate code for transformed Classes that target code languages.  If you select this option, the first time you transform to the Class the system prompts you to select a filename to generate code into; subsequent transformations automatically generate code to that filename.	
<b>Perform Transformations on result</b>	Select the checkbox to automatically execute transformations <b>previously</b> done on the target Class or Classes.	<a href="#">Chaining Transformations</a> <sup>[2019]</sup>
<b>Intermediary File</b>	If you want to capture the intermediary language file (for example, to debug it), either type in the file path and name or click on the browse button ( ... ) and locate and select the file.	
<b>Write Always</b>	Select this checkbox to always write the intermediary file to disk.	<a href="#">Intermediary Language</a> <sup>[2053]</sup>  <a href="#">Intermediary Language Debugging</a> <sup>[2053]</sup>
<b>Write Now</b>	Click on this button to generate the intermediary file without performing the full transformation.	
<b>Do Transform</b>	Click on this button to execute the transformation.  When the transformation is complete, the Model Transformation dialog closes.	
<b>Close</b>	Click on this button to close the Model Transformation dialog without performing the transformation.	

**Notes**

- When the dialog displays, all elements are selected and all transformations previously performed from any of these Classes are checked
- This procedure does not apply to the Sequence diagram/Communication diagram transformation, or the Communication diagram/Sequence diagram transformation

**Learn more**

- [Sequence/Communication Diagram Transformations](#) <sup>[2041]</sup>

### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Model Transformations | Transform | Transform Elements**
- (Alt+F1) | **Enterprise Architect | Model Transformations | Transform | Transform Packages**

## **10.1.1 Chaining Transformations**

Chaining transformations provides an extra degree of flexibility and power to performing transformations. For example, if two transformations have a common element; you might separate this element out into its own transformation, and then perform the original transformations from the common point. The separated transform could even produce a useful model itself.

You can chain transformations by setting the **Perform Transformations on result** checkbox in the Model Transformation dialog, so that transformations that have already been performed on target Classes are executed **automatically** next time that Class is transformed to.

### Learn more





- [Transform Elements](#)  2017

## 10.2 Built-in Transformations

Enterprise Architect provides a number of built-in transformations that are designed to be of value to as wide a range of users as possible, to act as a base that you can tailor to the specifics of your custom domain, and to serve as examples of how to write transformations.

### Built-in Transformations

Transformation	Converts	See also
<b>C#</b>	Platform-Independent Model (PIM) elements to language-specific C# Class elements.	<a href="#">C# Transformation</a> <small>2021</small>
<b>C++</b>	PIM elements to language-specific C++ Class elements.	<a href="#">C++ Transformation</a> <small>2023</small>
<b>Data Definition Language</b>	A logical model to a data model targeted at the default database type, ready for DDL generation.	<a href="#">DDL Transformation</a> <small>2025</small>
<b>Entity Relationship Diagram to Data Model</b>	An ERD logical model to a data model targeted at the default database type, ready for DDL generation.	<a href="#">ERD To Data Model Transformation</a> <small>2032</small>
<b>Data Model to Entity Relationship Diagram</b>	A data model to an ERD logical model.	<a href="#">Data Model To ERD Transformation</a> <small>2024</small>
<b>EJB Session Bean</b>	A single Class element to the elements of an EJB <b>session</b> .	<a href="#">EJB Transformations</a> <small>2029</small>
<b>EJB Entity Bean</b>	A single Class element to the elements of an EJB <b>entity</b> .	
<b>Java</b>	PIM elements to language-specific Java Class elements.	<a href="#">Java Transformation</a> <small>2034</small>
<b>JUnit</b>	An existing Java Class element with public methods to a Class with a test method for each public method, plus the methods required to appropriately set up the tests.	<a href="#">JUnit Transformation</a> <small>2037</small>
<b>NUnit</b>	An existing .Net compatible Class with public methods to a Class with a test method for each public method, plus the methods required to appropriately set up the tests.	<a href="#">NUnit Transformation</a> <small>2038</small>
<b>PHP</b>	PIM elements to language-specific PHP Class elements.	<a href="#">PHP Transformation</a> <small>2040</small>

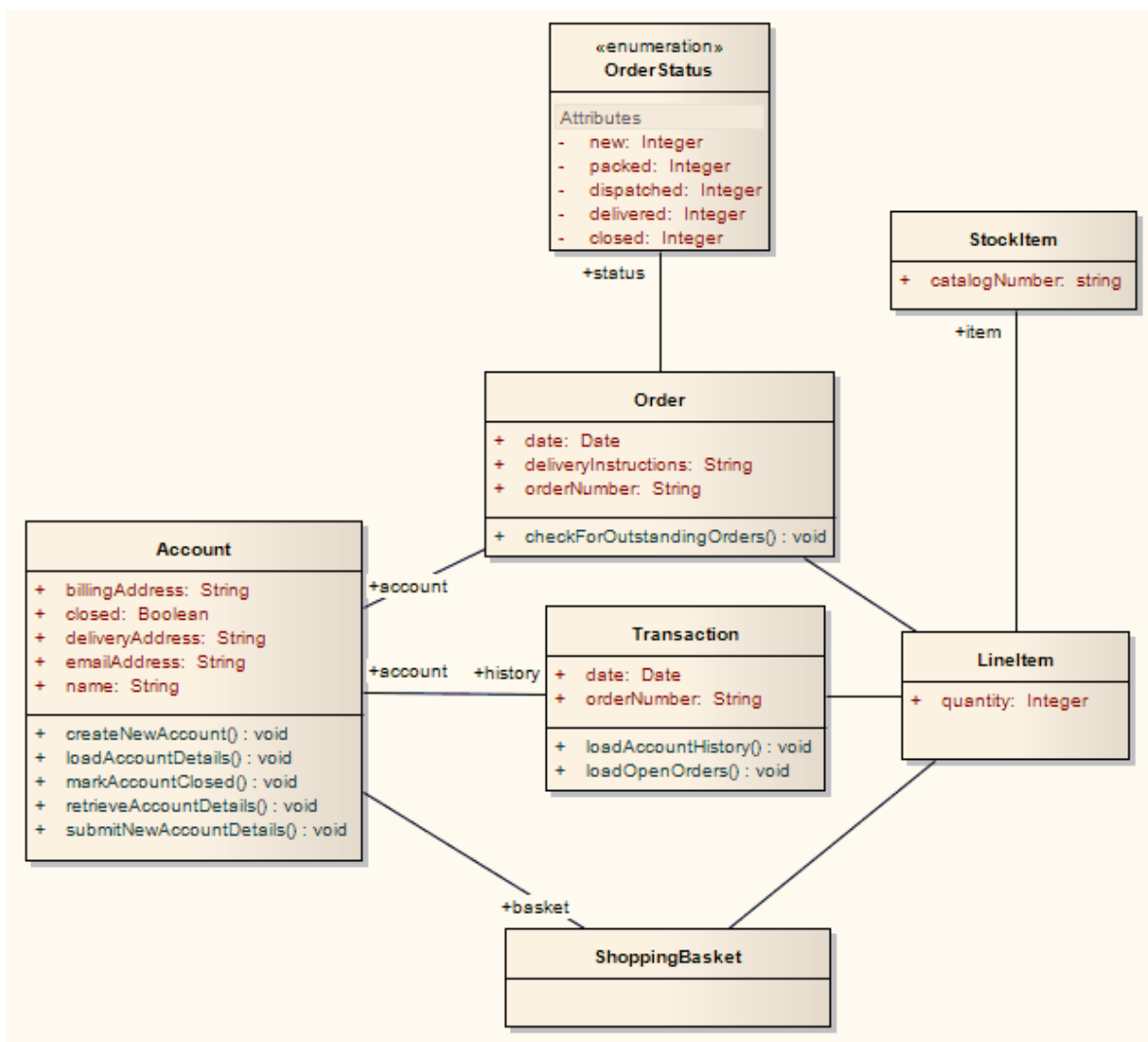
Transformation	Converts	See also
<b>Sequence/ Communication Diagram</b>	All elements and messages in an opened Sequence diagram into matching elements and messages in a Communication diagram, and vice versa.	<a href="#">Sequence/ Communication Diagram Transformations</a>  2041
<b>VB.Net</b>	PIM elements to language-specific VB.Net Class elements.	<a href="#">VB.Net Transformation</a>  2042
<b>WSDL</b>	A simple model to an expanded model of a WSDL interface, suitable for generation.	<a href="#">WSDL Transformation</a>  2043
<b>XSD</b>	PIM elements to UML Profile for XML elements, as an intermediary step in creating an XML Schema.	<a href="#">XSD Transformation</a>  2044

### 10.2.1 C# Transformation

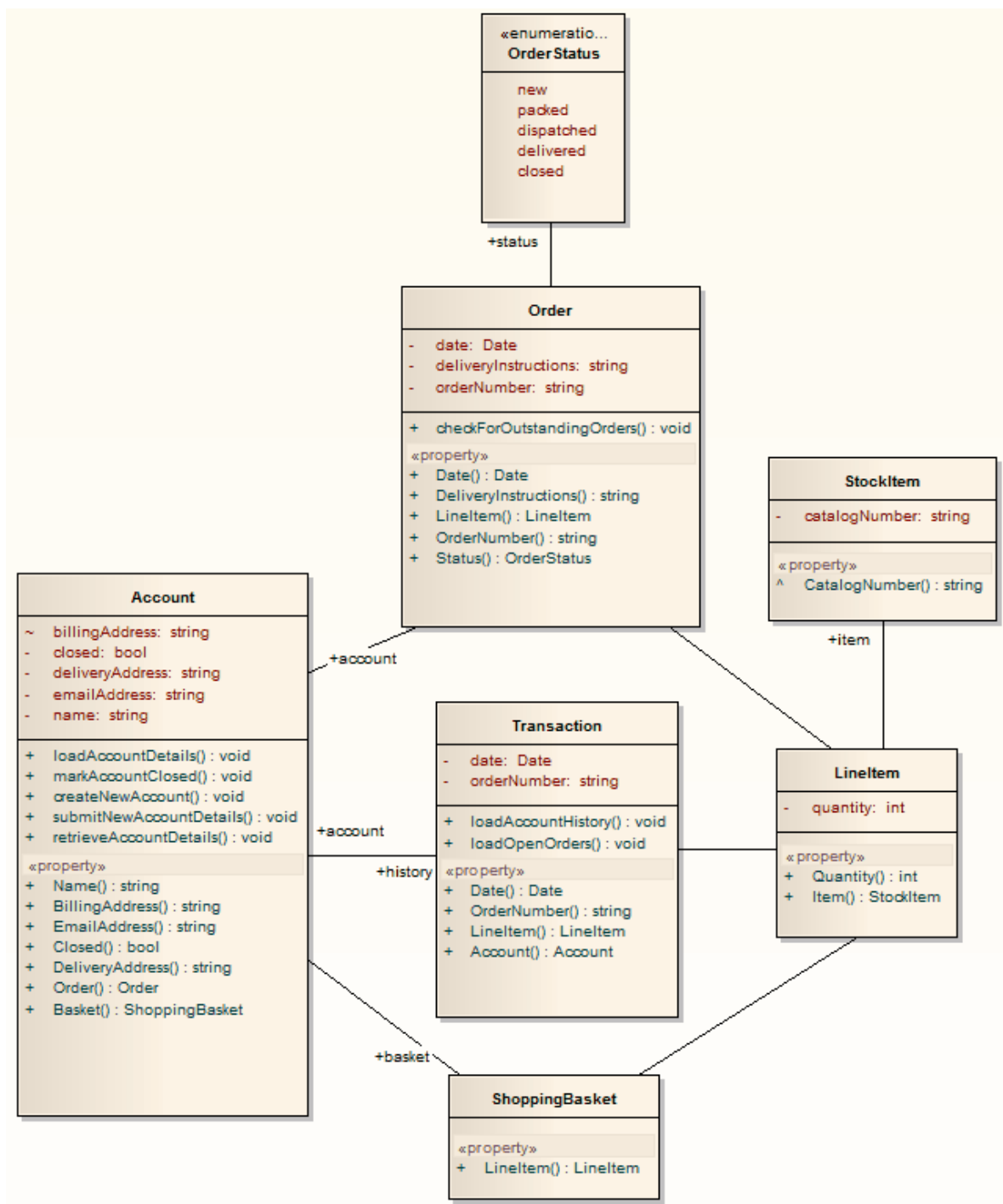
The **C#** transformation converts Platform-Independent Model (PIM) element types to C#-specific Class element types, and creates encapsulation according to the system options you have set for creating properties from C# attributes (on the C# Specifications page of the Options dialog).

#### Example

#### The PIM elements



After transformation, become the PSM elements



### Learn more

- [C# Options Page](#) <sup>2268</sup>

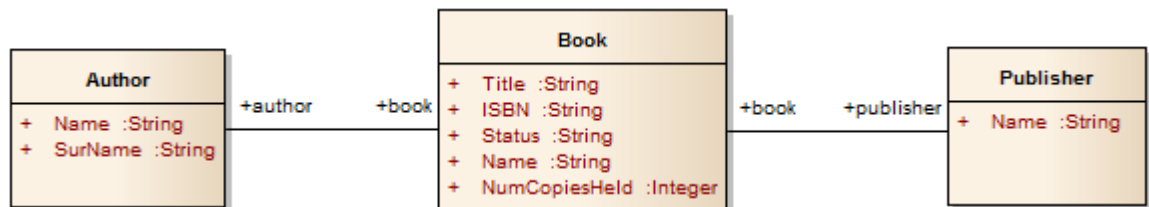
## 10.2.2 C++ Transformation

The **C++** transformation converts Platform-Independent Model (PIM) element types to C++ specific Class element types and creates encapsulation (producing the getters and setters) according to the options you have set for creating properties from C++ attributes (on the C++ Specifications page of the Options dialog). Note that the **public** attributes in the PIM are converted to **private** attributes in the PSM. All operations on an

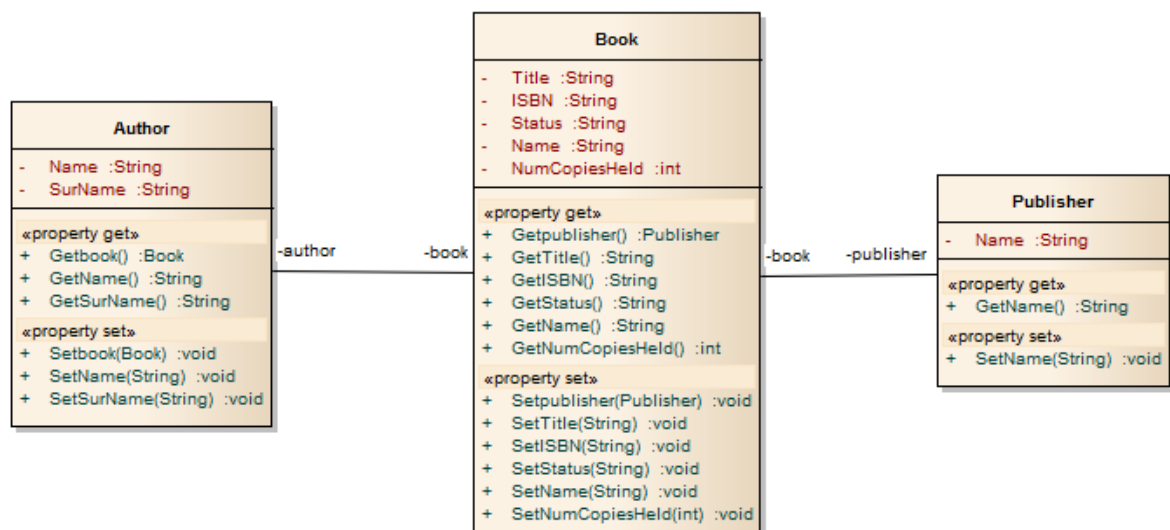
interface are transformed into pure virtual methods on an equivalent class.

### Example

#### The PIM elements



#### After transformation, become the PSM elements



### Learn more

- [C++ Options](#) <sup>[2269]</sup>

## 10.2.3 Data Model To ERD Transformation

The **Data Model to Entity Relationship Diagram (ERD)** transformation creates an ERD logical model from a Data Model. It is the reverse of the **ERD to Data Model** transformation. This transformation uses and demonstrates support in the intermediary language for a number of database-specific concepts.

### Supported Concepts

Concept	Effect	See also
<b>Entity</b>	Mapped one-to-one onto <b>Table</b> elements.	



Concept	Effect	See also
<b>Attribute</b>	Mapped one-to-one onto <b>Columns</b> .	
<b>Primary Key</b>	Derived from the <b>PrimaryKey</b> type of column.	

### Notes

- Sometimes you might want to limit the stretch of the diamond-shaped **Relationship** connectors; simply pick a Relationship connector, right-click to display the context menu, and select the **Bend Line at Cursor** option

### Learn more

- [ERD to Data Model Transformation](#) <sup>[2032]</sup>
- [Data Models](#) <sup>[1937]</sup>

## 10.2.4 DDL Transformation

The **DDL** transformation converts the logical model to a data model that is targeted at the default database type and that is ready for DDL generation. The data model can then be used to automatically generate DDL statements to run in one of the system-supported database products.

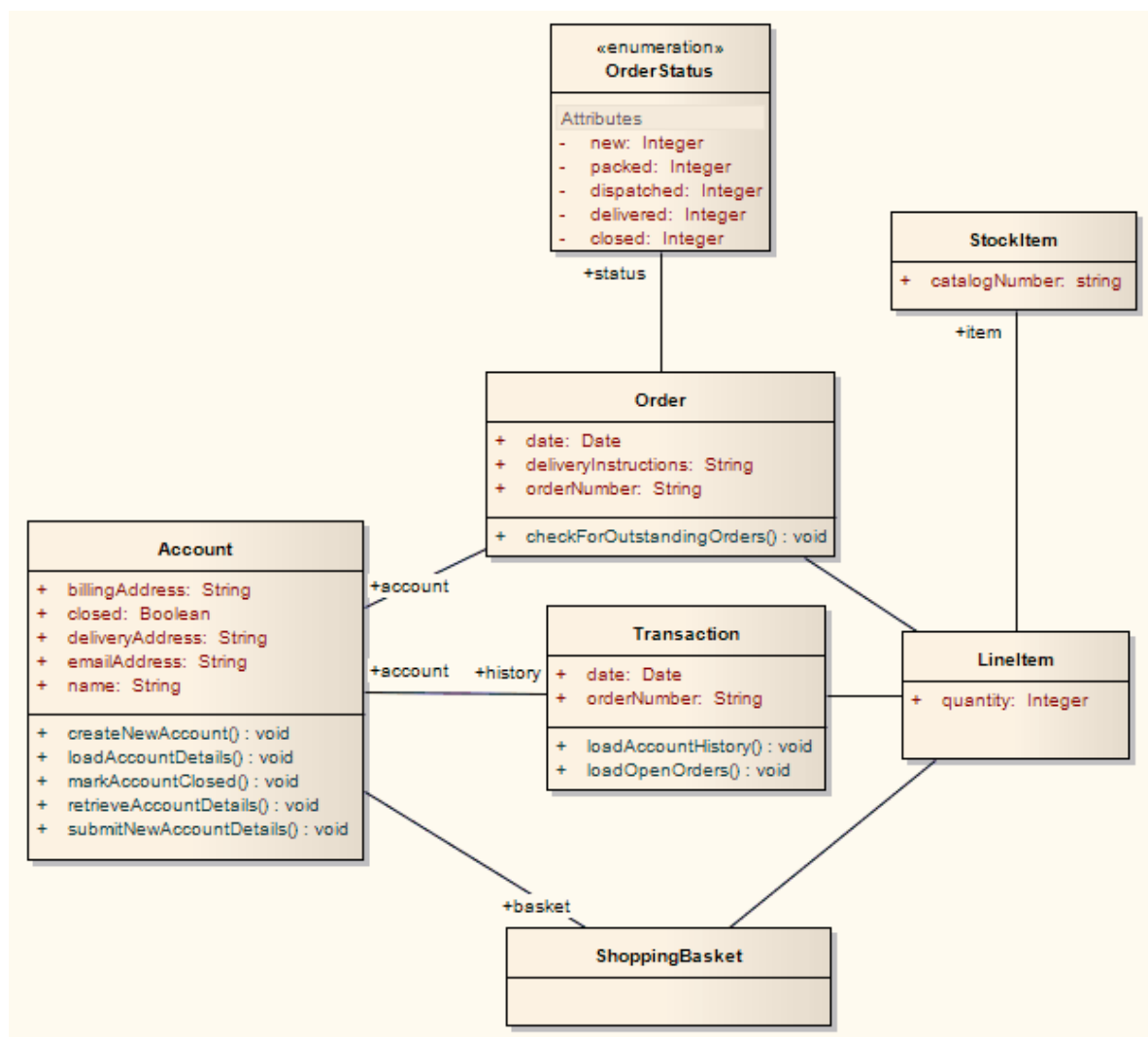
The DDL transformation uses and demonstrates support in the intermediary language for a number of database-specific concepts.

### Concepts

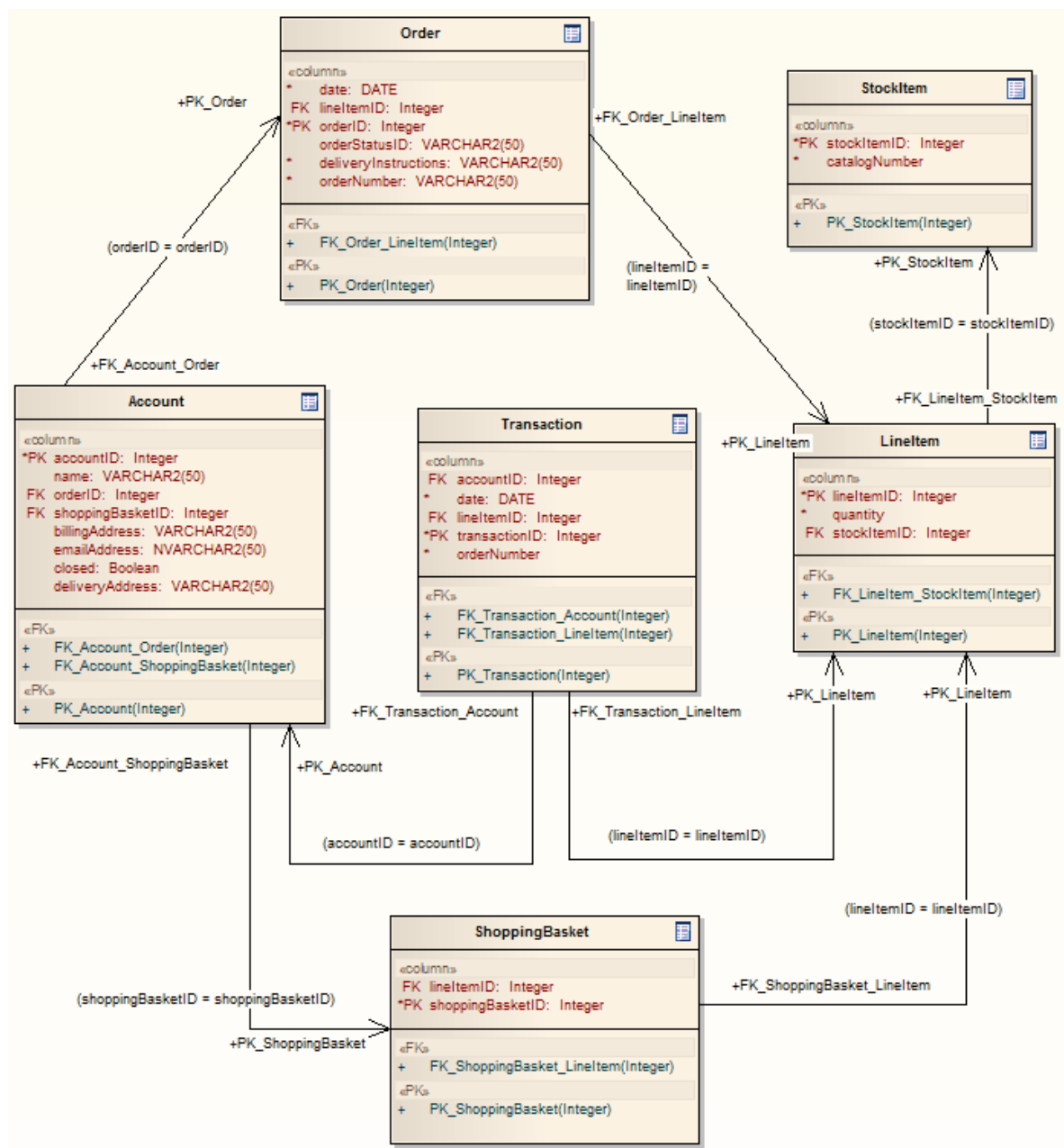
Concept	Effect	See also
<b>Table</b>	Mapped one-to-one onto <b>Class</b> elements.  'Many-to-many' relationships are supported by the transformation, creating <b>Join</b> tables.	
<b>Column</b>	Mapped one-to-one onto <b>attributes</b> .	
<b>Primary Key</b>	Lists all the columns involved so that they exist in the Class, and creates a <b>Primary Key Method</b> for them.	
<b>Foreign Key</b>	A special sort of connector, in which the <b>Source</b> and <b>Target</b> sections list all of the columns involved so that: <ul style="list-style-type: none"> <li>The columns exist</li> <li>A matching Primary Key exists in the destination Class, and</li> <li>The transformation creates the appropriate Foreign Key</li> </ul>	<a href="#">Transform Foreign Keys</a> <sup>[2065]</sup>

**Notes**

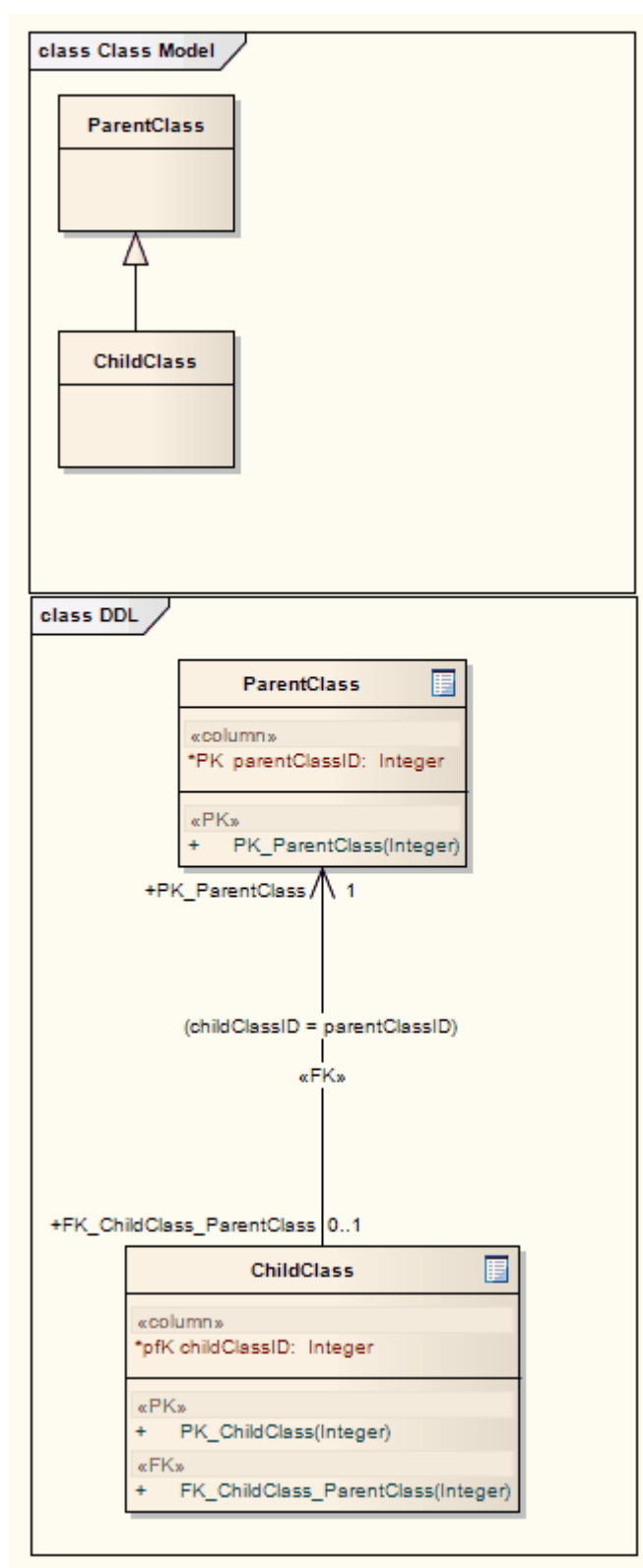
- You can define DBMS-specific aspects not depicted in a Logical model, such as Stored Procedures, Triggers, Views and Check Constraints, after the transformation; see the *Physical Data Model* topic

**Example****The PIM elements**

After transformation, become the PSM elements



**Generalizations** are handled by providing the child element with a Foreign Key to the parent element, as shown. Copy-down inheritance is not supported.



#### Learn more

- [Intermediary Language](#) <sup>[2053]</sup>
- [Data Models](#) <sup>[1937]</sup>

- [Physical Data Model](#) 

## 10.2.5 EJB Transformations

The **EJB Session Bean** and **EJB Entity Bean** transformations reduce the work required to generate the internals of Enterprise Java Beans. You can therefore focus on modeling at a higher level of abstraction.

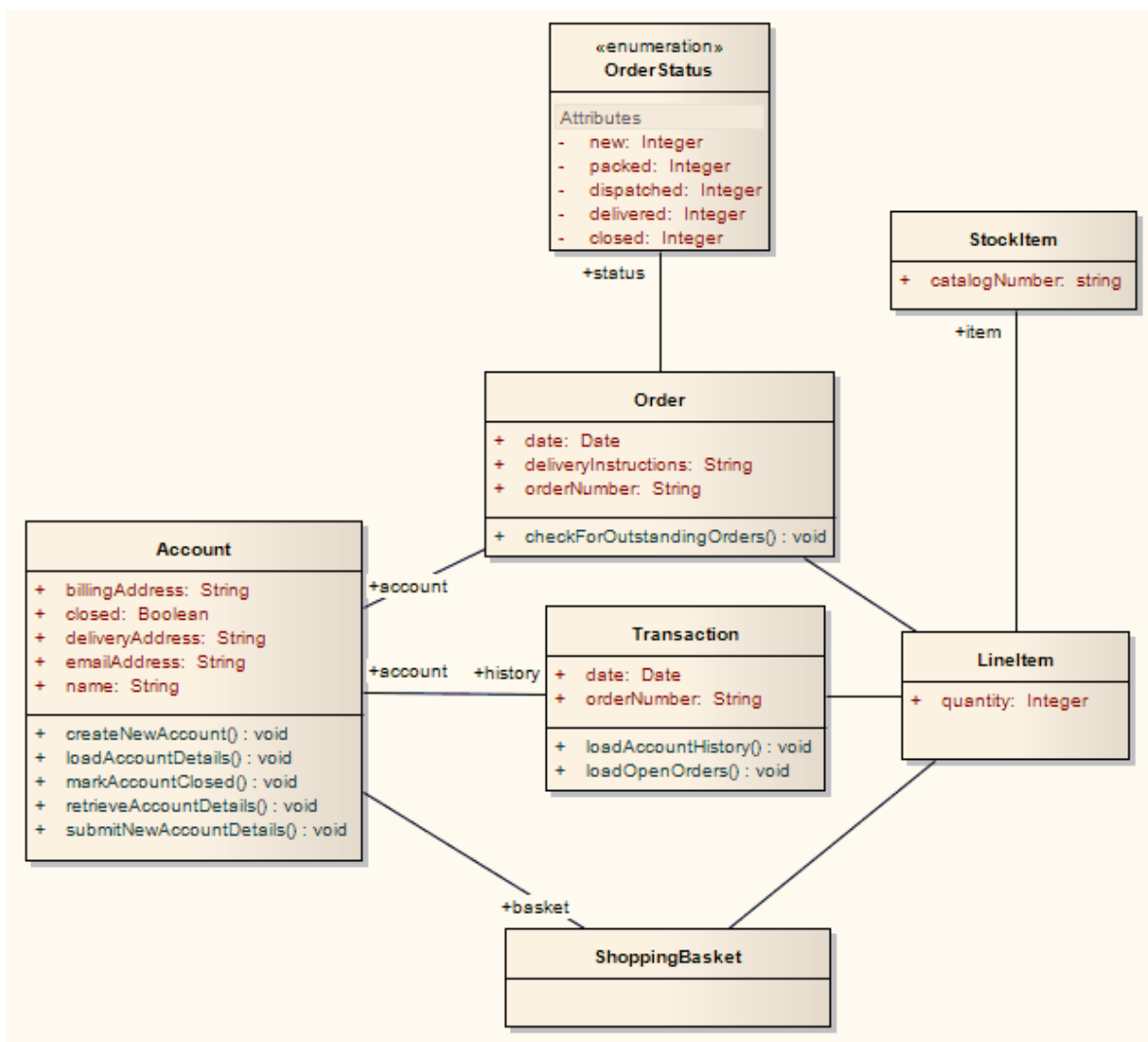
### Transformations

Transformation	Detail	See also
<b>EJB Session Bean</b>	This transformation converts a single Class element (containing the attributes, operations and references required for code generation by the javax.ejb.* package) to <ul style="list-style-type: none"><li>• An <b>implementation Class</b> element</li><li>• A <b>home interface</b> element</li><li>• A <b>remote interface</b> element</li></ul>	
<b>EJB Entity Bean</b>	This transformation converts a single Class element (containing the attributes, operations and references required for code generation by the javax.ejb.* package) to: <ul style="list-style-type: none"><li>• An <b>implementation Class</b> element</li><li>• A <b>home interface</b> element</li><li>• A <b>remote interface</b> element</li><li>• A Primary Key element</li></ul>	

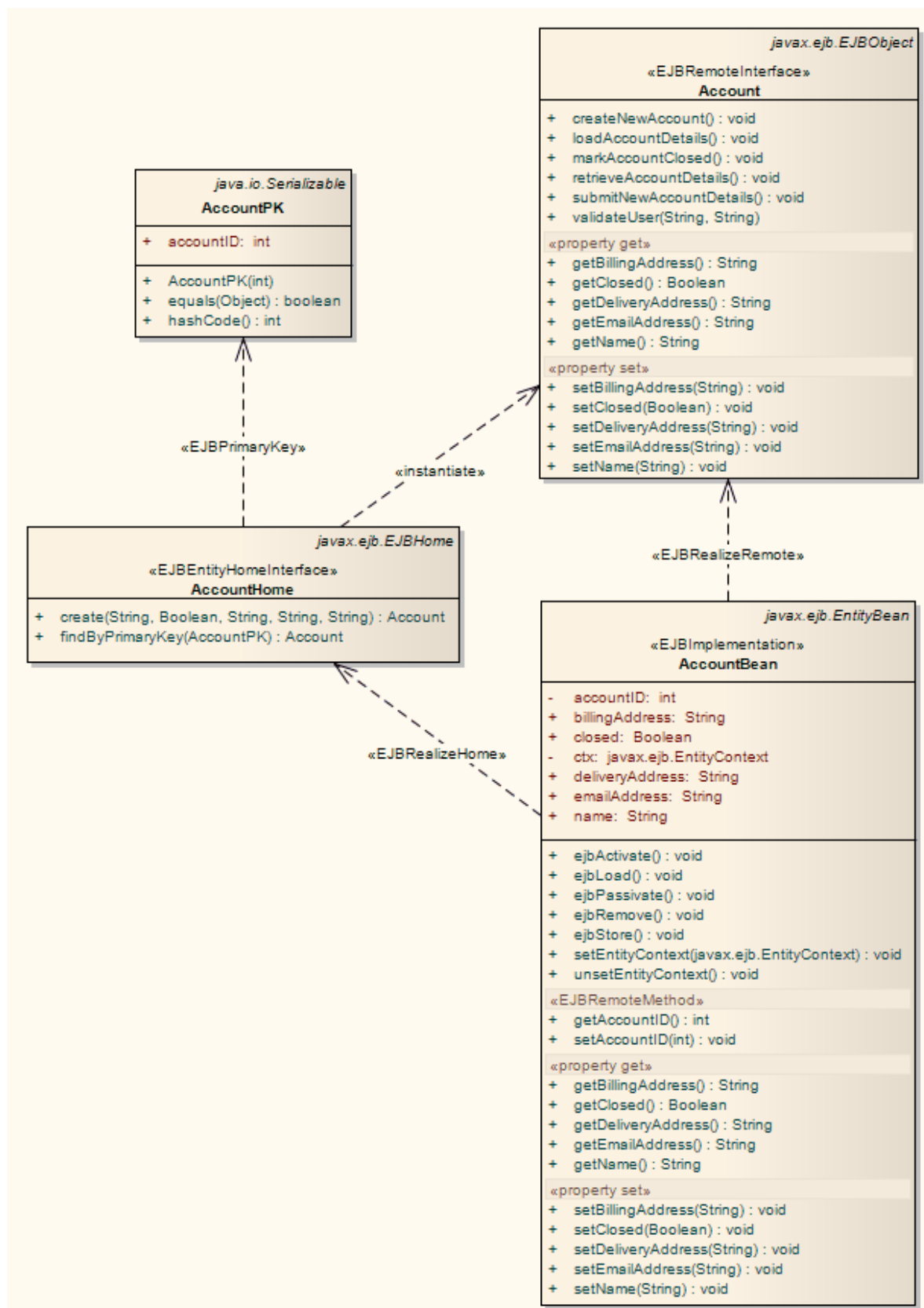
Both transformations also generate a **META-INF** package containing a **deployment descriptor** element.

### Example

#### The PIM elements



After transformation generate a set of Entity Beans, where each one takes the following form (for the Account Class)



[Learn more](#)

- [Software Engineering](#) <sup>[2073]</sup>
- [Generate Source Code](#) <sup>[2111]</sup>

### 10.2.6 ERD To Data Model Transformation

The **Entity Relationship Diagram (ERD) to Data Model** transformation converts an ERD logical model to a data model targeted at the default database type, ready for generating DDL statements to run in one of the system-supported database products. Before doing the transformation, you define the common data type for each attribute and select a database type as the default database. You can then automatically generate the data modeling diagram.

The transformation uses and demonstrates support in the intermediary language for a number of database-specific concepts.

#### Concepts

Concept	Definition	See also
<b>Table</b>	Mapped one-to-one onto <b>Class</b> elements.	
<b>Column</b>	Mapped one-to-one onto <b>attributes</b> .	
<b>Primary Key</b>	Lists all the columns involved so that they exist in the Class, and creates a <b>Primary Key Method</b> for them.	
<b>Foreign Key</b>	<p>A special sort of connector, in which the <b>Source</b> and <b>Target</b> sections list all of the columns involved so that:</p> <ul style="list-style-type: none"> <li>• The columns exist</li> <li>• A matching Primary Key exists in the destination Class, and</li> <li>• The transformation creates the appropriate Foreign Key</li> </ul>	<a href="#">Transform Foreign Keys</a> <sup>[2065]</sup>

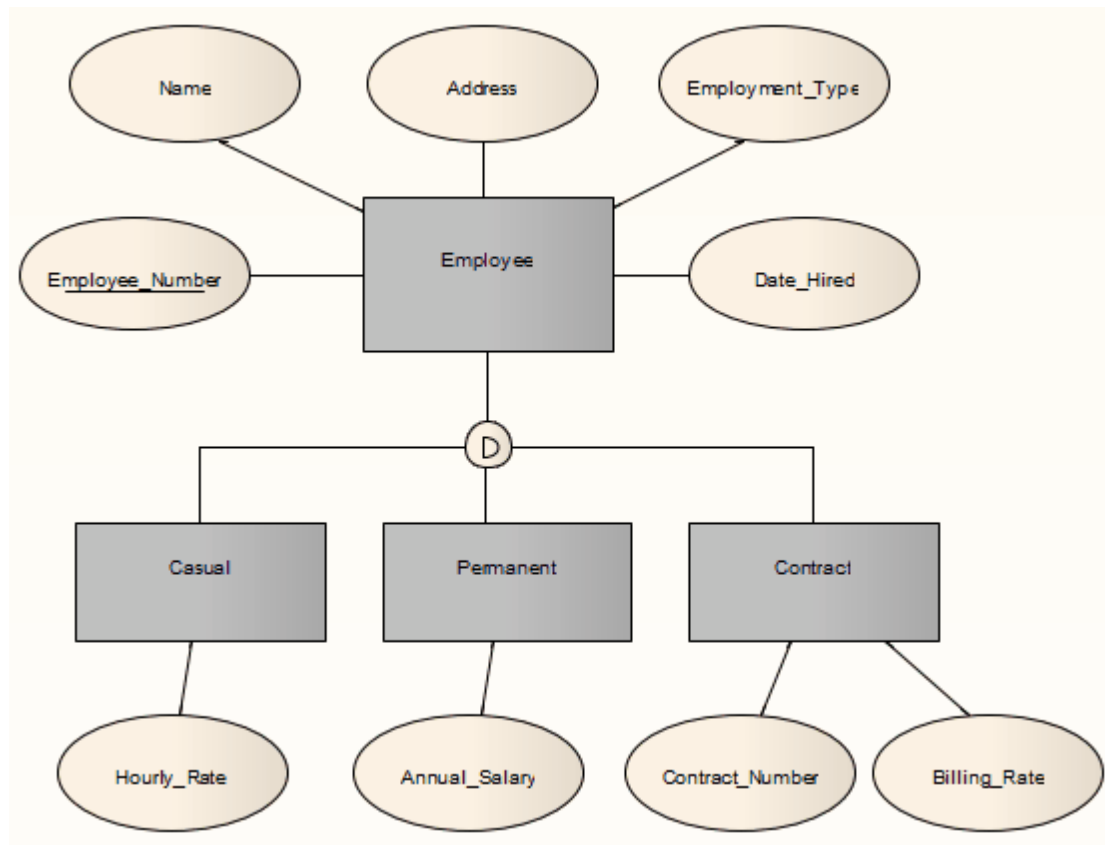
#### Generalization

ERD technology can handle Generalization, as shown. Note that the copy-down inheritance is currently supported with two levels only.

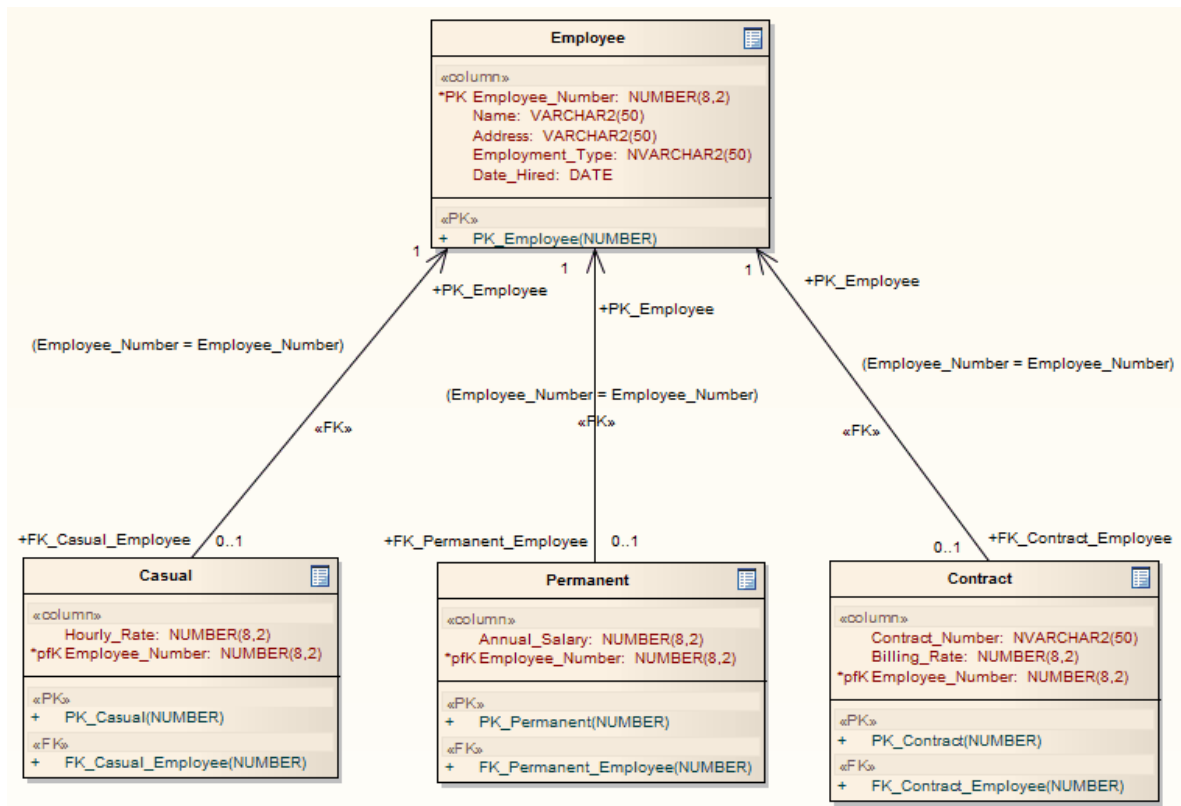
#### Example

The ERD elements





After transformation, become the Data Model elements



## Notes

- Sometimes you might go back to the ERD, make some changes and then need to do another transformation; in this case, to achieve better results, always **delete** the **previous** transformation package before doing the next transformation

## Learn more

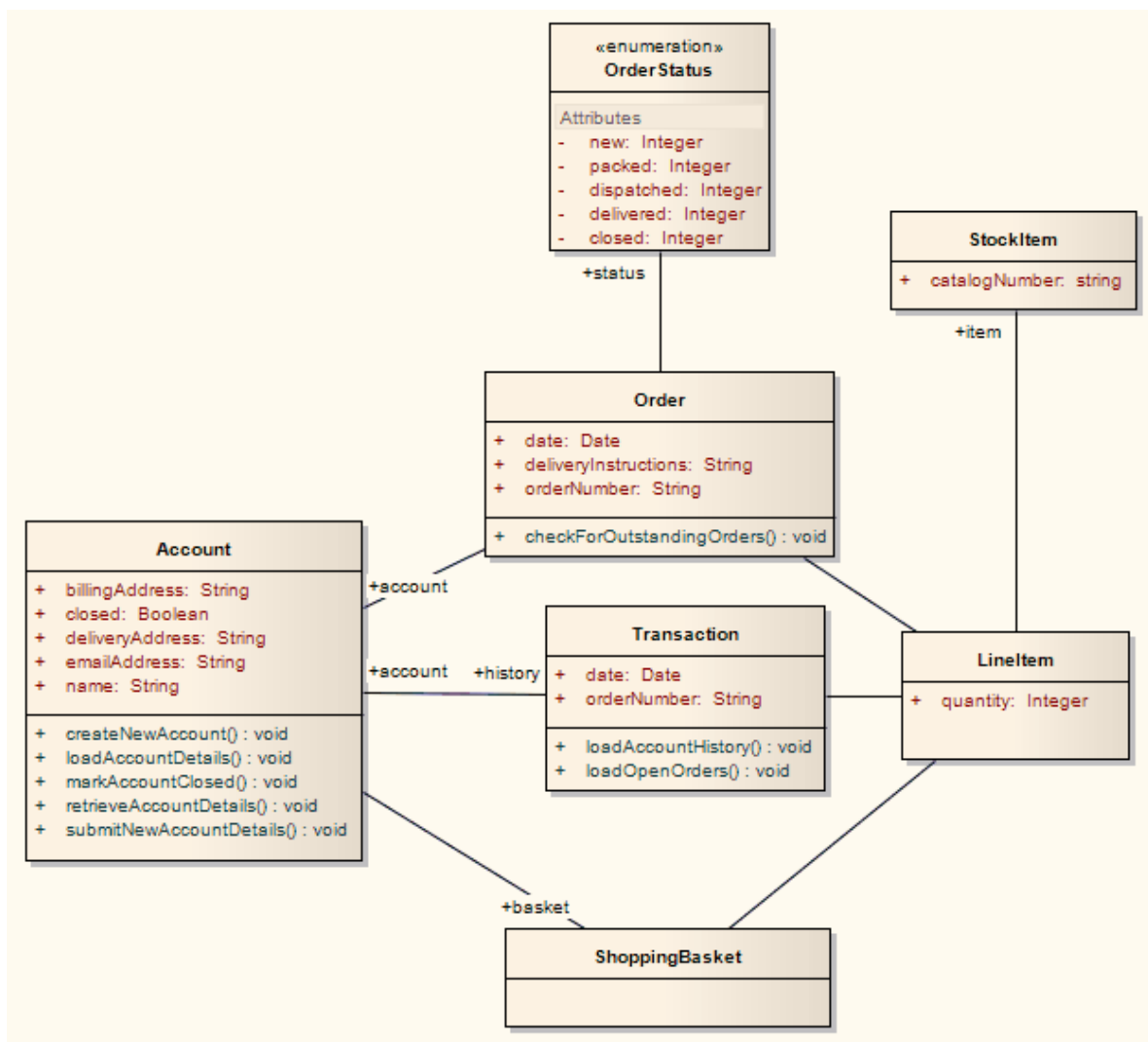
- [Intermediary Language](#) <sup>[2053]</sup>
- [Data Models](#) <sup>[1937]</sup>
- [Physical Data Model](#) <sup>[2335]</sup>

## 10.2.7 Java Transformation

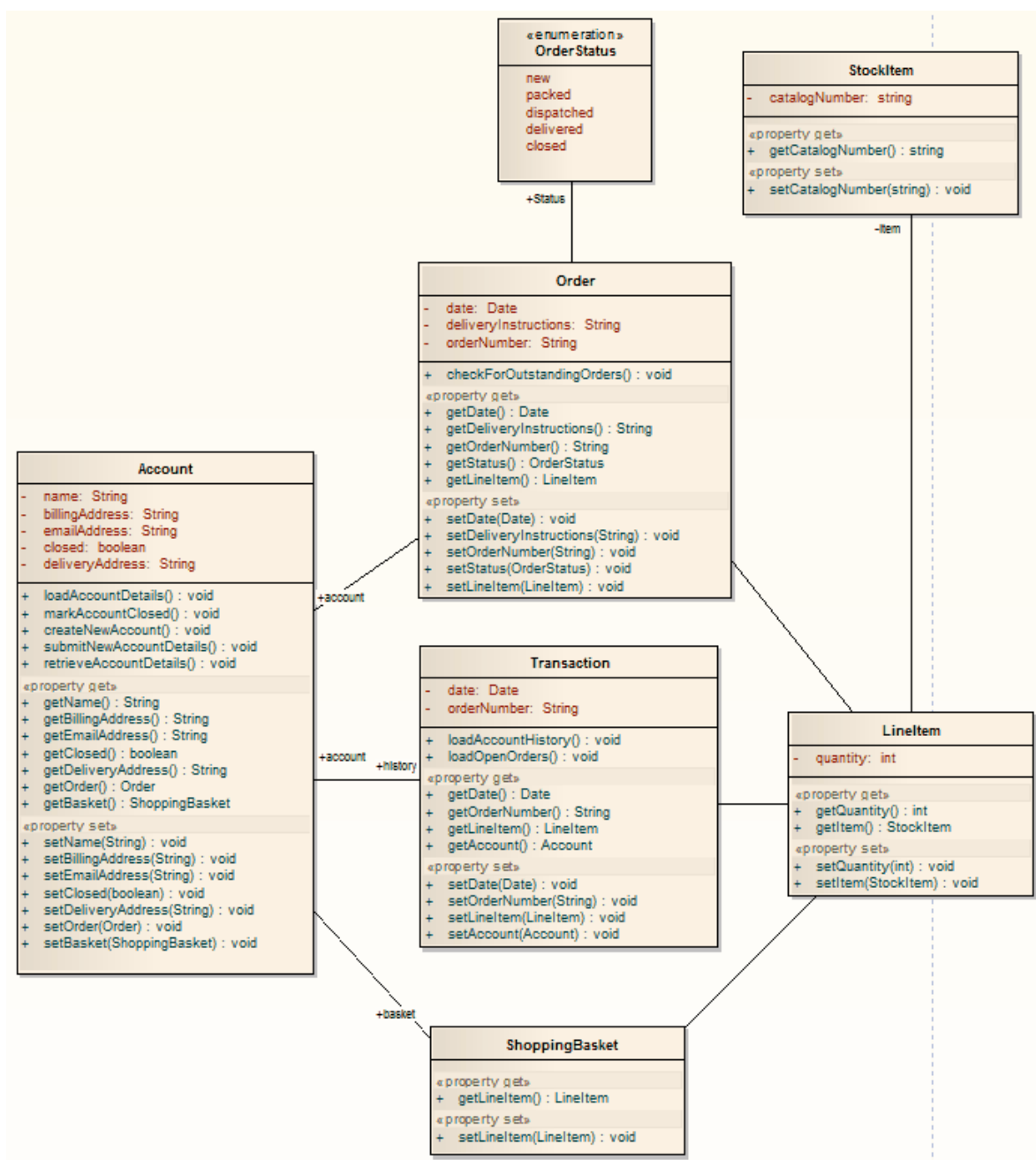
The **Java** transformation converts Platform-Independent Model (PIM) element types to Java-specific Class element types, and creates encapsulation (producing the getters and setters) according to the options you have set for creating properties from Java attributes (on the Java Specifications page of the Options dialog). Note that the **public** attributes in the PIM are converted to **private** attributes in the PSM. All operations in the interface are transformed into pure virtual methods.

## Example

The PIM elements



After transformation, become the PSM elements



### Learn more

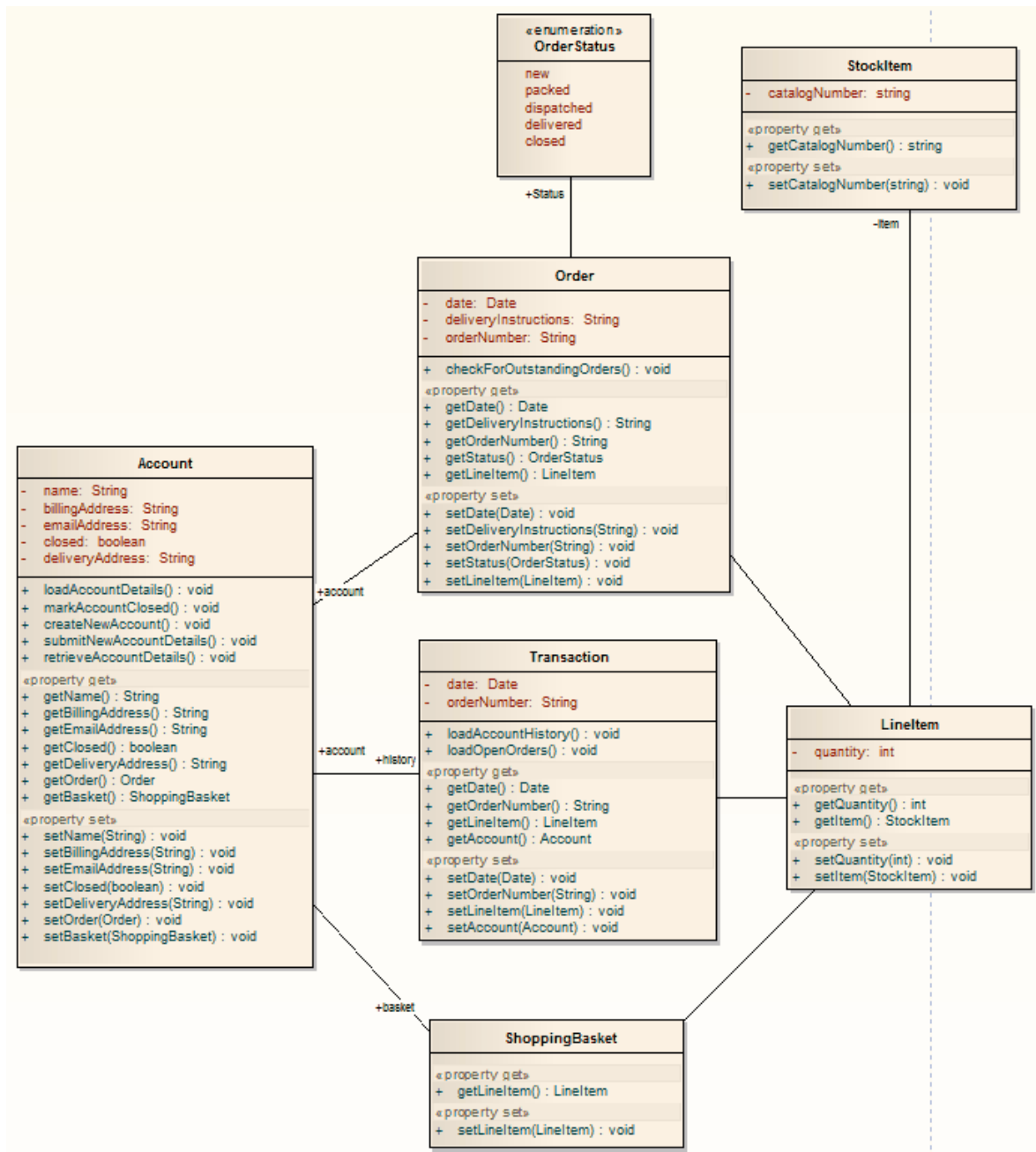
- [Java Options](#) <sup>2272</sup>
- [Software Engineering](#) <sup>2073</sup>
- [Generate Source Code](#) <sup>2111</sup>

### 10.2.8 JUnit Transformation

The **JUnit** transformation converts an existing Java Class with public methods into a Class with a test method for each public method. The resulting Class can then be generated and the tests filled out and run by JUnit.

#### Example

The Java model elements (originally transformed from the PIM)



After transformation, become the PSM elements

<div>junit.framework.TestCase</div> <div>AccountTest</div> <div>+ AccountTest(String) + <u>main(String[]) : void</u> # setUp() : void # tearDown() : void + testCreateNewAccount() : void + testGetBasket() : void + testGetBillingAddress() : void + testGetClosed() : void + testGetDeliveryAddress() : void + testGetEmailAddress() : void + testGetName() : void + testGetOrder() : void + testLoadAccountDetails() : void + testMarkAccountClosed() : void + testRetrieveAccountDetails() : void + testSetBasket() : void + testSetBillingAddress() : void + testSetClosed() : void + testSetDeliveryAddress() : void + testSetEmailAddress() : void + testSetName() : void + testSetOrder() : void + testSubmitNewAccountDetails() : void</div>	<div>junit.framework.TestCase</div> <div>OrderTest</div> <div>+ <u>main(String[]) : void</u> + OrderTest(String) # setUp() : void # tearDown() : void + testCheckForOutstandingOrders() : void + testGetDate() : void + testGetDeliveryInstructions() : void + testGetLineItem() : void + testGetOrderNumber() : void + testGetStatus() : void + testSetDate() : void + testSetDeliveryInstructions() : void + testSetLineItem() : void + testSetOrderNumber() : void + testSetStatus() : void</div>	<div>junit.framework.TestCase</div> <div>TransactionTest</div> <div>+ <u>main(String[]) : void</u> # setUp() : void # tearDown() : void + testGetAccount() : void + testGetDate() : void + testGetLineItem() : void + testGetOrderNumber() : void + testLoadAccountHistory() : void + testLoadOpenOrders() : void + testSetAccount() : void + testSetDate() : void + testSetLineItem() : void + testSetOrderNumber() : void + TransactionTest(String)</div>	
<div>junit.framework.TestCase</div> <div>LineItemTest</div> <div>+ LineItemTest(String) + <u>main(String[]) : void</u> # setUp() : void # tearDown() : void + testGetItem() : void + testGetQuantity() : void + testSetItem() : void + testSetQuantity() : void</div>	<div>junit.framework.TestCase</div> <div>ShoppingBasketTest</div> <div>+ <u>main(String[]) : void</u> # setUp() : void + ShoppingBasketTest(String) # tearDown() : void + testGetLineItem() : void + testSetLineItem() : void</div>	<div>junit.framework.TestCase</div> <div>StockItemTest</div> <div>+ <u>main(String[]) : void</u> # setUp() : void + StockItemTest(String) # tearDown() : void + testGetCatalogNumber() : void + testSetCatalogNumber() : void</div>	<div>junit.framework.TestCase</div> <div>OrderStatusTest</div> <div>+ <u>main(String[]) : void</u> + OrderStatusTest(String) # setUp() : void # tearDown() : void</div>

### Notes

- For each Class in the Java model, a corresponding Test Class has been created containing a test method for every public method in the source Class, plus the methods required to appropriately set up the tests; you fill in the details of each test

### Learn more

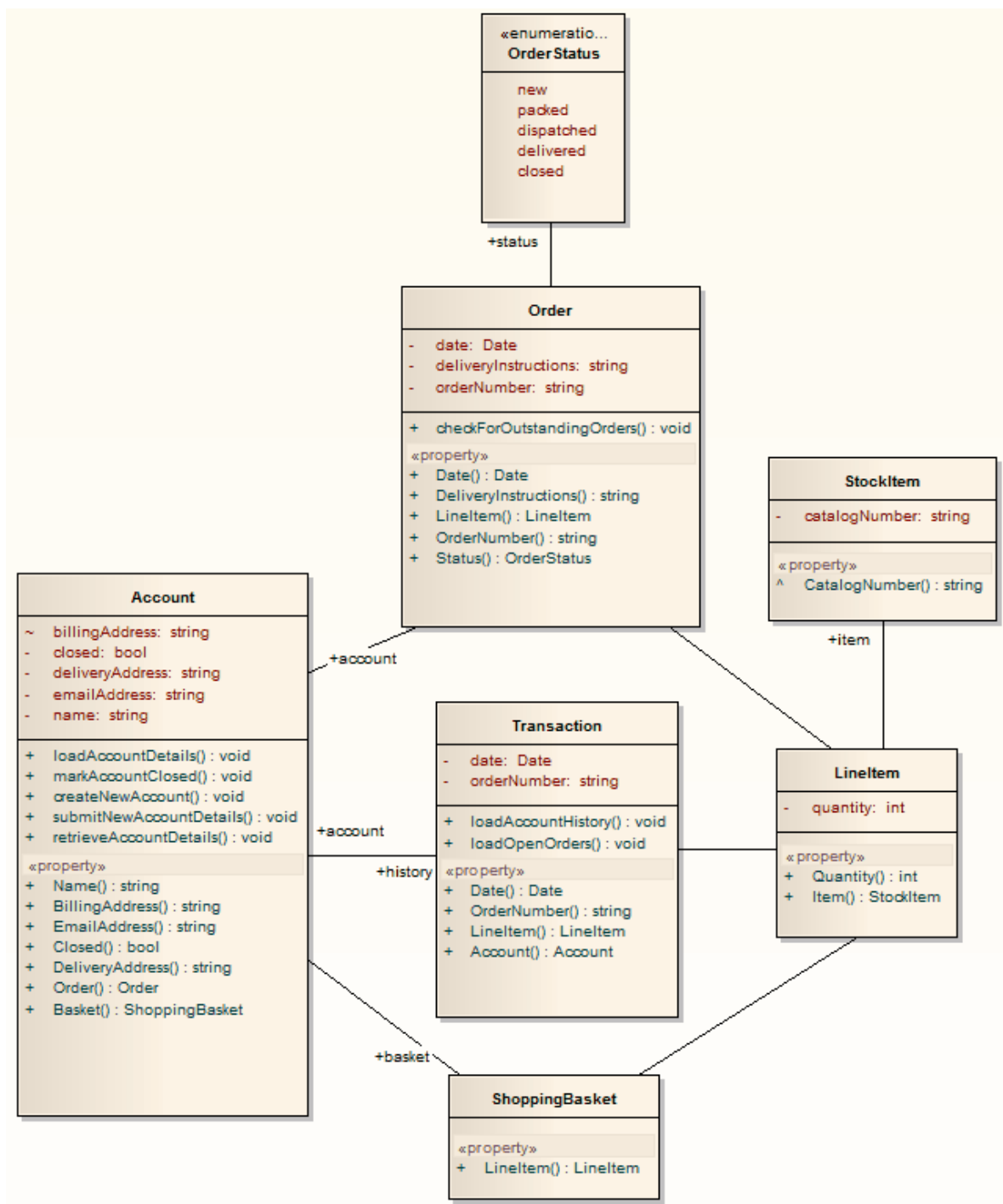
- [Unit Testing](#) <sup>[257]</sup>

## 10.2.9 NUnit Transformation

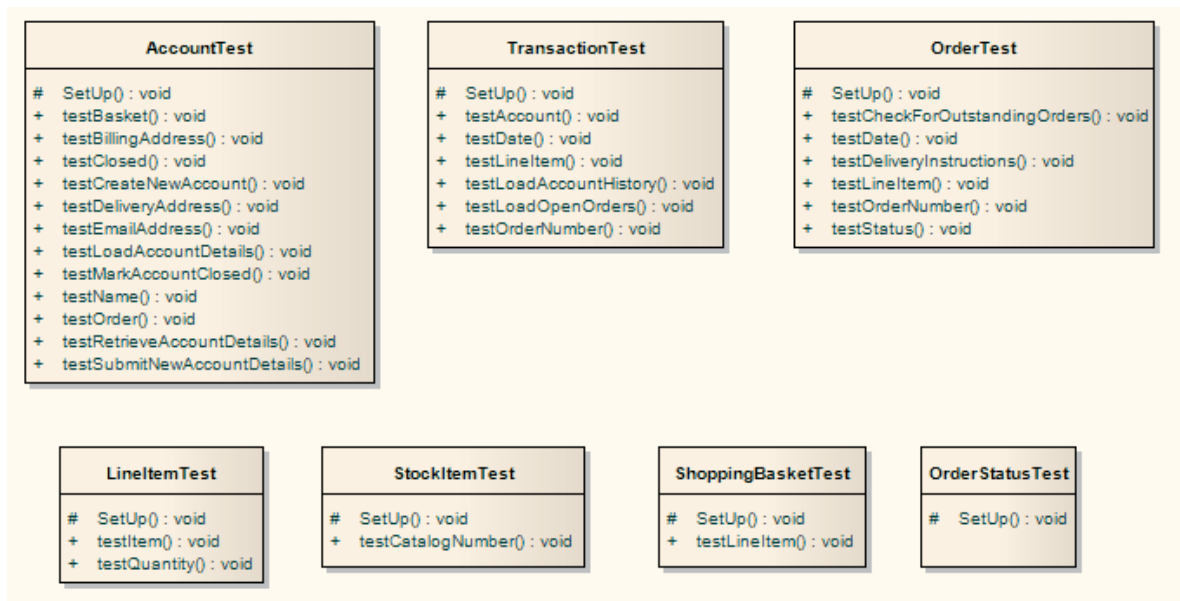
The **NUnit** transformation converts an existing .Net compatible Class with public methods into a Class with a test method for each public method. The resulting Class can then be generated and the tests filled out and run by NUnit.

### Example

The C# elements (originally transformed from the PIM)



After transformation, become the PSM elements



### Notes

- For each Class in the C# model, a corresponding Test Class has been created containing a test method for every public method in the source Class, plus the methods required to appropriately set up the tests; you fill in the details of each test

### Learn more

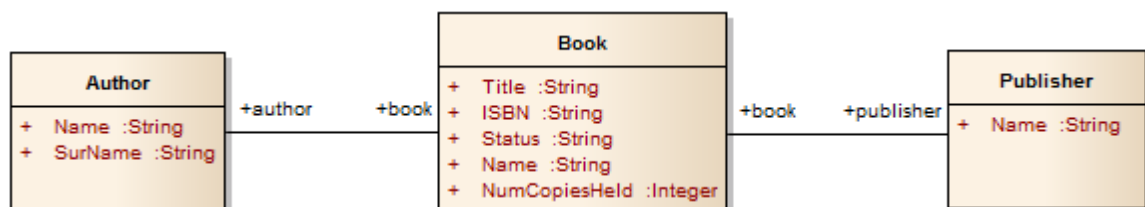
- [Unit Testing](#) <sup>[2573]</sup>

## 10.2.10 PHP Transformation

The **PHP** transformation converts Platform-Independent Model (PIM) element types to language-specific PHP Class element types and creates encapsulation (producing the getters and setters) according to the options you have set for creating properties from PHP attributes (on the PHP Specifications page of the Options dialog). Note that the **public** attributes in the PIM are converted to **private** attributes in the PSM.

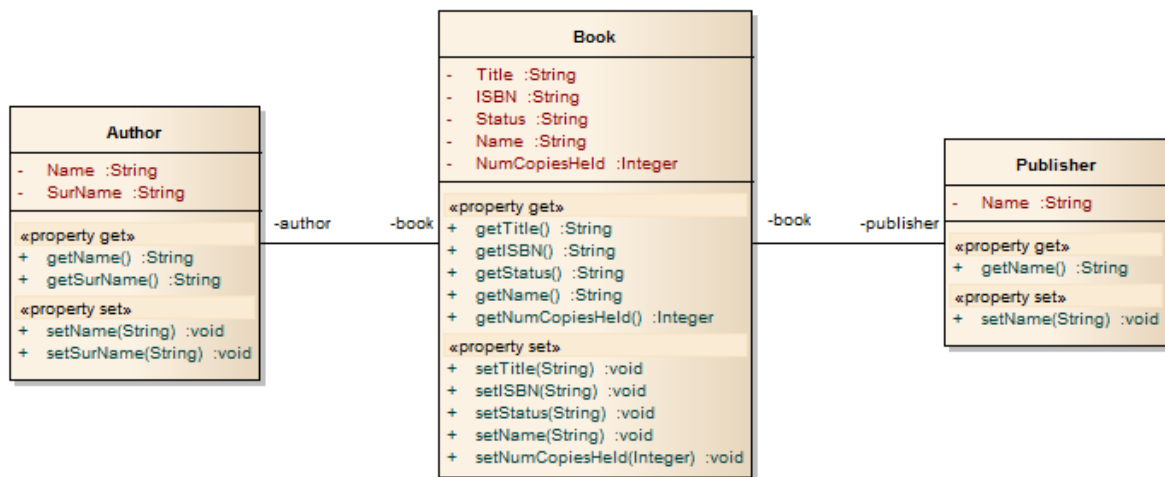
### Example

The PIM elements



After transformation, become the PSM elements



Learn more

- [PHP Options](#) <sup>2274</sup>

**10.2.11 Sequence/Communication Diagram Transformations**

It is possible to transform a Sequence diagram into a Communication diagram, and to transform a Communication diagram into a Sequence diagram. In each case, every element or message in the source diagram type is mapped 1:1 to a matching element or message in the target diagram.

**Access** **Open source diagram | Tools | Model Transformation (MDA) | Transform Current Package (Ctrl+Shift+H)** or **Open source diagram, click on element | Tools | Model Transformation (MDA) | Transform Selected Elements (Ctrl+H)**

Perform a Transformation

Field/Button	Action	See also
<b>Elements</b>	Lists and highlights all the elements from the diagram, which will be included in the transformation.	
<b>Transformations</b>	<p>Select:</p> <ul style="list-style-type: none"> <li>• The <b>Communication</b> checkbox, if transforming a Sequence diagram into a Communication diagram, or</li> <li>• The <b>Sequence</b> checkbox, if transforming a Communication diagram into a Sequence diagram</li> </ul> <p>In either case, the Browse Project dialog displays. Click on the <b>OK</b> button; in these transformations the target diagram is automatically created in the same package as the source diagram.</p>	
<b>Do Transform</b>	Click on this button to execute the transformation.	

Field/Button	Action	See also
	<p>The target diagram is created and listed in the Project Browser under the parent package with the name (depending on which transformation you have executed):</p> <p><i>&lt;source diagram name&gt; Communication</i> or  <i>&lt;source diagram name&gt; Sequence</i></p>	

### Notes

- Once you have selected the **Communication** or **Sequence** checkbox, these transforms ignore any other field setting in the dialog and will perform a direct transformation of every element in the source diagram

### Learn more

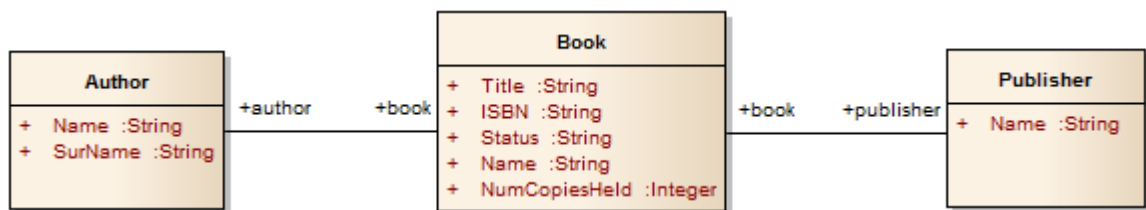
- [Transform Elements](#) <sup>[2017]</sup>
- [Sequence Diagram](#) <sup>[1249]</sup>
- [Communication Diagram](#) <sup>[1259]</sup>

## 10.2.12 VB.Net Transformation

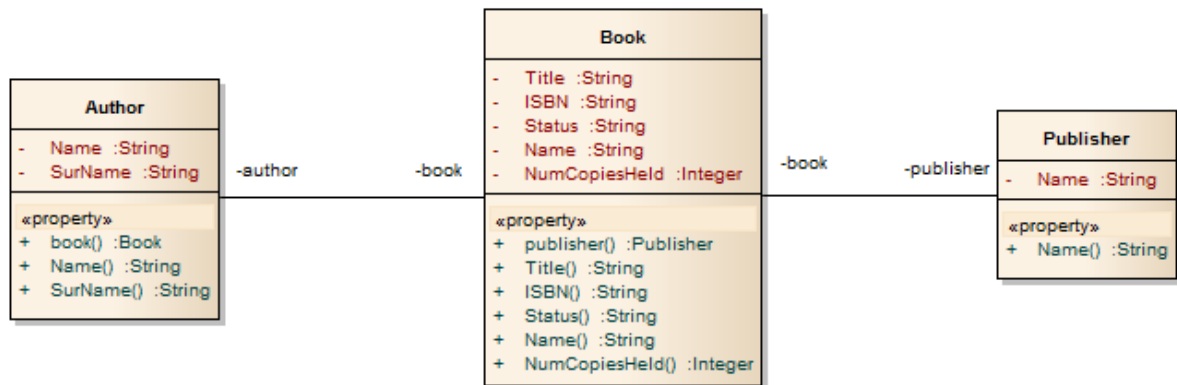
The **VB.Net** transformation converts Platform-Independent Model (PIM) element types to language-specific VB.Net Class element types, and creates encapsulation according to the options you have set for creating properties from VB.Net attributes (on the VB.Net Specifications page of the Options dialog). Note that the **public** attributes in the PIM are converted to **private** attributes in the PSM.

### Example

The PIM elements



After transformation, become the PSM elements

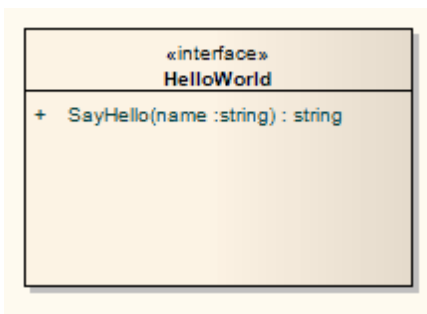


### Learn more

- [VB.Net Options](#) 2275

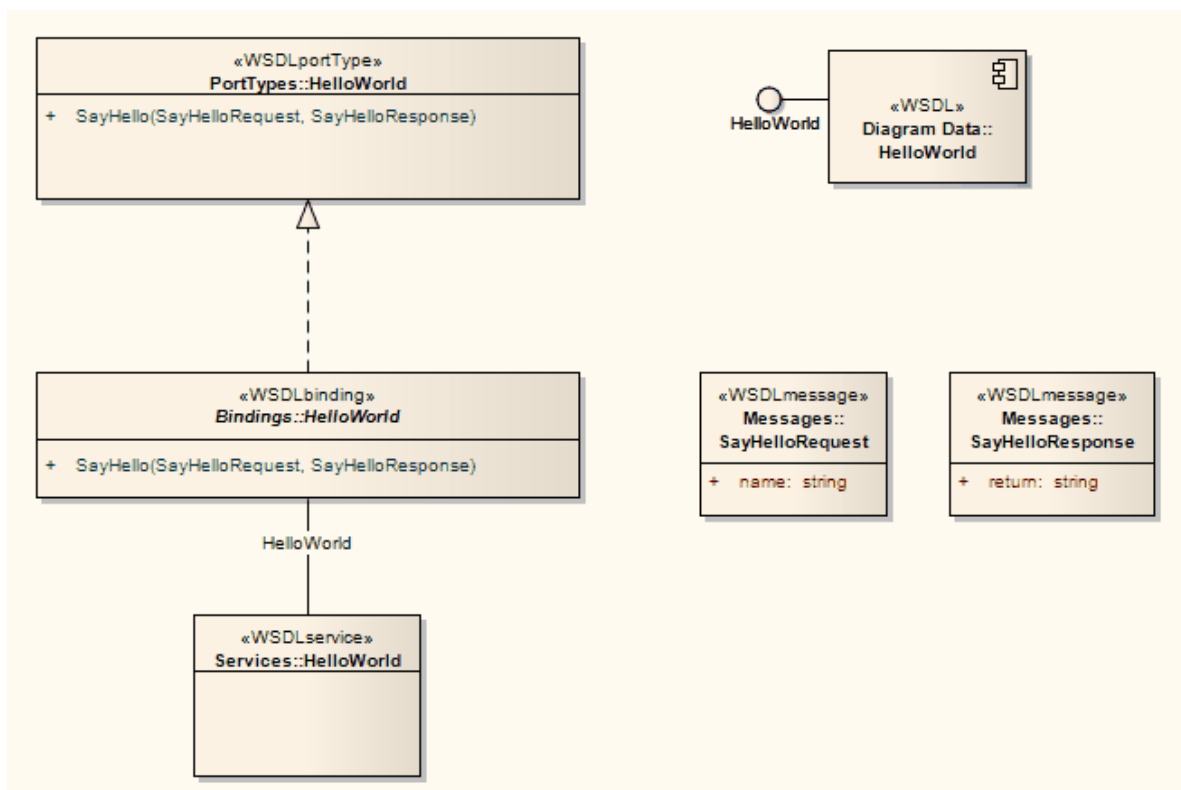
## 10.2.13 WSDL Transformation

The **WSDL transformation** converts a simple model into an expanded model of a WSDL interface that is suitable for generation. Take the following example interface:



Transformation of this generates the corresponding WSDL Component, Service, Port Type, Binding and Messages as follows:

- Classes are handled in the same way as in the XSD Transformation
- All **in** parameters are transformed into WSDL **Message Parts** in the **Request** message
- The return value and all **out** and **return** parameters are transformed into WSDL **Message Parts** in the **Response** message
- All methods where a value is returned are transformed into **Request-Response** operations, and all methods not returning a value are transformed into **OneWay** operations
- The transformation does not handle the generation of **Solicit-Response** and **Notification** methods or faults



In the resulting package you can then fill out the specifics using the WSDL editing capabilities of Enterprise Architect, and finally generate the package using the WSDL generation tools.

#### Learn more

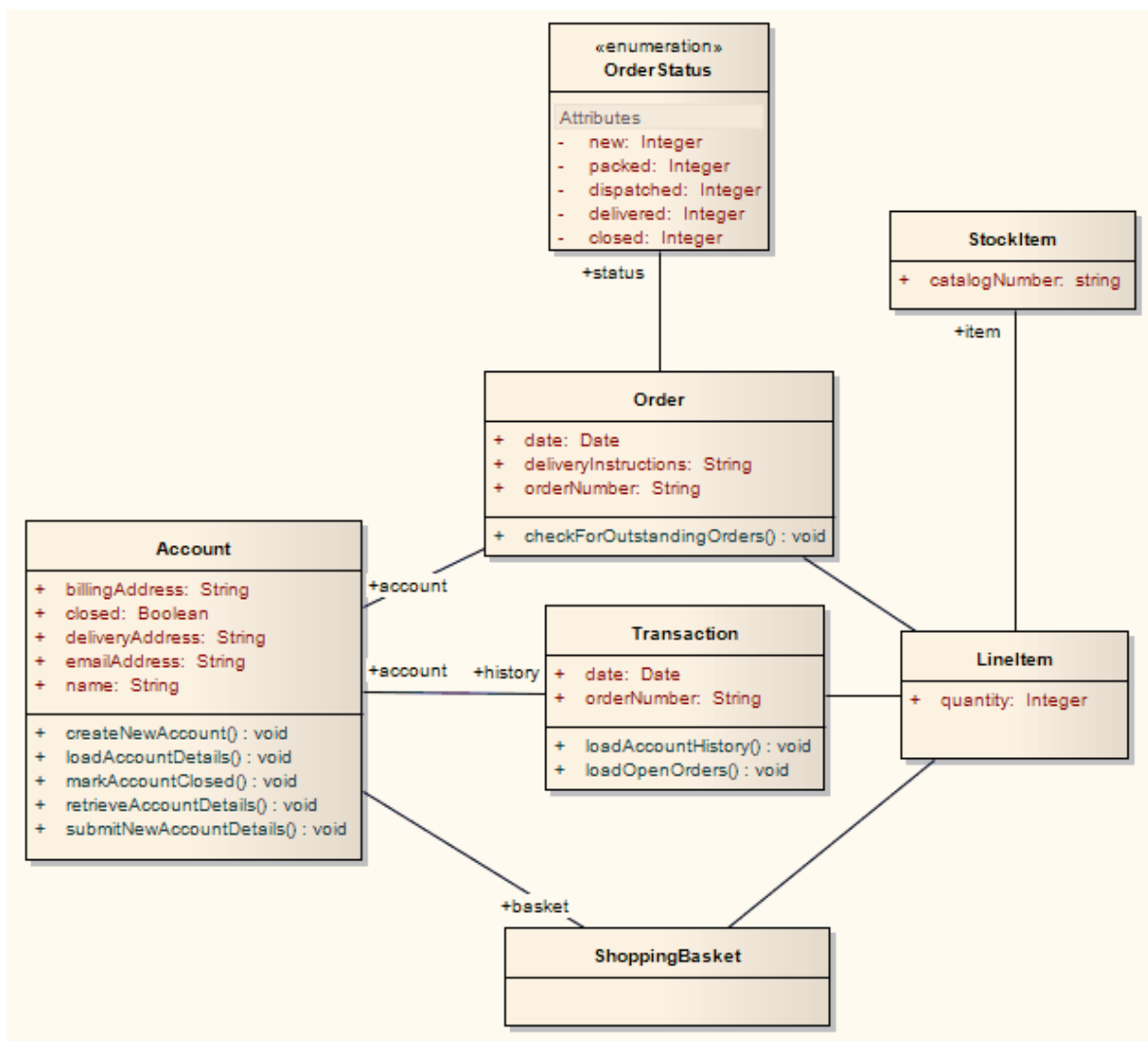
- [WSDL Model Structure](#) <sup>[2429]</sup>
- [Model WSDL](#) <sup>[2426]</sup>
- [XSD Transformation](#) <sup>[2044]</sup>
- [WSDL Generation](#) <sup>[2445]</sup>

### 10.2.14 XSD Transformation

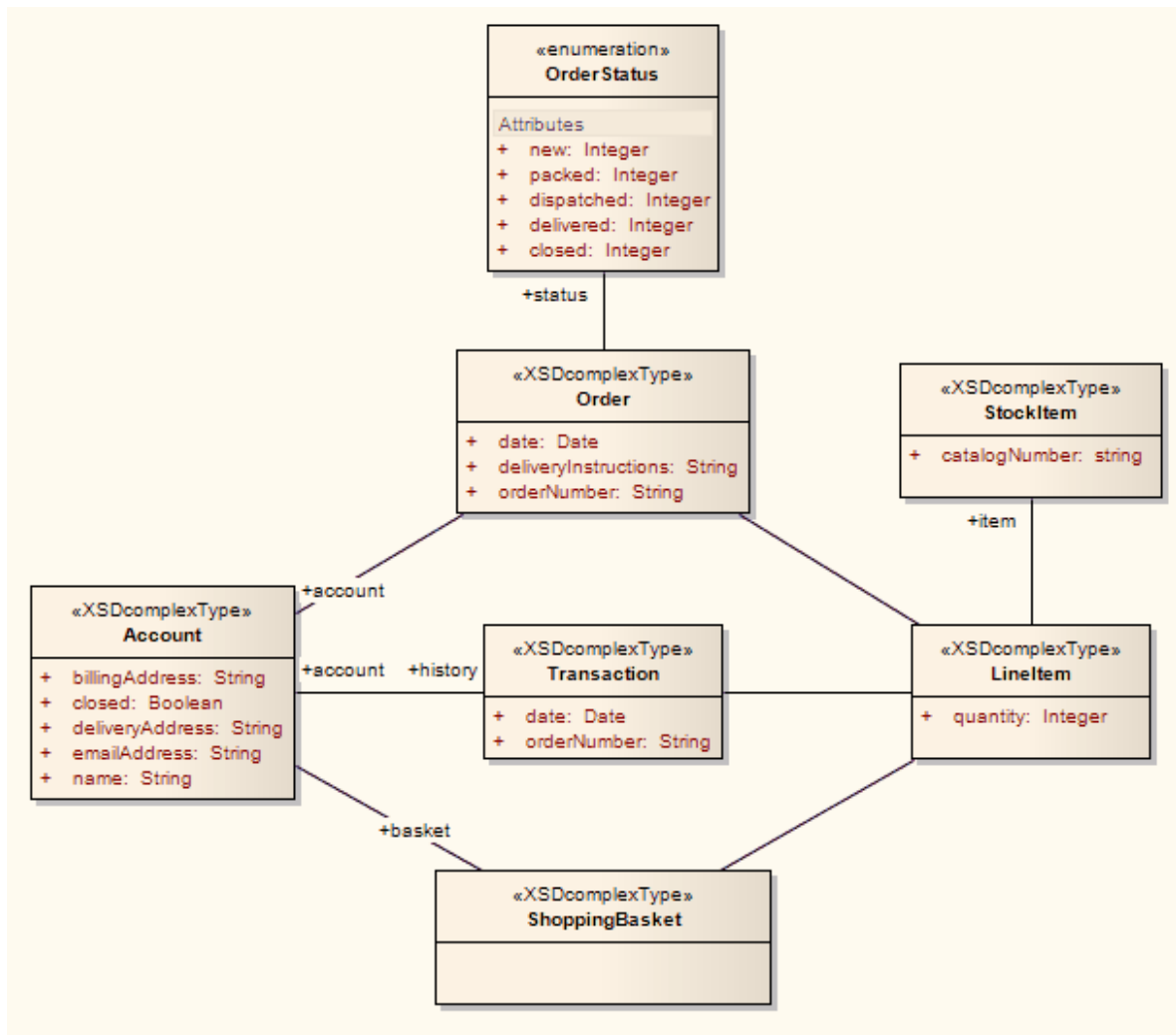
The **XSD transformation** converts Platform-Independent Model (PIM) elements to UML Profile for XML elements as an intermediary step in creating an XML Schema. Each selected PIM Class element is converted to an `«XSDcomplexType»` element.

#### Example

The PIM elements



After transformation become the PSM elements



These in turn generate this XSD

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Account" type="Account"/>
  <xs:complexType name="Account">
    <xs:sequence>
      <xs:element name="name" type="xs:string"/>
      <xs:element name="billingAddress" type="xs:string"/>
      <xs:element name="emailAddress" type="xs:string"/>
      <xs:element name="closed" type="xs:boolean"/>
      <xs:element name="deliveryAddress" type="xs:string"/>
      <xs:element ref="Order"/>
      <xs:element ref="ShoppingBasket"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="LinelItem" type="LinelItem"/>
  <xs:complexType name="LinelItem">
    <xs:sequence>
      <xs:element name="quantity" type="xs:integer"/>
      <xs:element ref="StockItem"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Order" type="Order"/>
  <xs:complexType name="Order">
    <xs:sequence>
      <xs:element name="date" type="xs:date"/>
      <xs:element name="deliveryInstructions" type="xs:string"/>
      <xs:element name="orderNumber" type="xs:string"/>
      <xs:element ref="Account"/>
      <xs:element ref="Transaction"/>
      <xs:element ref="LinelItem"/>
      <xs:element ref="StockItem"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Transaction" type="Transaction"/>
  <xs:complexType name="Transaction">
    <xs:sequence>
      <xs:element name="date" type="xs:date"/>
      <xs:element name="orderNumber" type="xs:string"/>
      <xs:element ref="Account"/>
      <xs:element ref="LinelItem"/>
      <xs:element ref="ShoppingBasket"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="ShoppingBasket" type="ShoppingBasket"/>
  <xs:complexType name="ShoppingBasket">
    <xs:sequence>
      <xs:element name="basket" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="StockItem" type="StockItem"/>
  <xs:complexType name="StockItem">
    <xs:sequence>
      <xs:element name="catalogNumber" type="xs:string"/>
      <xs:element ref="LinelItem"/>
      <xs:element ref="ShoppingBasket"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>

```

```

        <xs:sequence>
            <xs:element name="date" type="xs:date"/>
            <xs:element name="deliveryInstructions" type="xs:string"/
>
            <xs:element name="orderNumber" type="xs:string"/>
            <xs:element ref="LineItem"/>
            <xs:element name="status" type="OrderStatus"/>
        </xs:sequence>
    </xs:complexType>
    <xs:simpleType name="OrderStatus">
        <xs:restriction base="xs:string">
            <xs:enumeration value="new"/>
            <xs:enumeration value="packed"/>
            <xs:enumeration value="dispatched"/>
            <xs:enumeration value="delivered"/>
            <xs:enumeration value="closed"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:element name="ShoppingBasket" type="ShoppingBasket"/>
    <xs:complexType name="ShoppingBasket">
        <xs:sequence>
            <xs:element ref="LineItem"/>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="StockItem" type="StockItem"/>
    <xs:complexType name="StockItem">
        <xs:sequence>
            <xs:element name="catalogNumber" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="Transaction" type="Transaction"/>
    <xs:complexType name="Transaction">
        <xs:sequence>
            <xs:element name="date" type="xs:date"/>
            <xs:element name="orderNumber" type="xs:string"/>
            <xs:element ref="Account"/>
            <xs:element ref="LineItem"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>

```

### Learn more

- [Generate XSD](#) 

### Learning Center topics

- (Alt+F1) | **Enterprise Architect | SOA and XML Engineering | XSD | UML to XML Schema**

## 10.3 Edit Transformation Templates

A single transformation applies a number of **transformation templates**, each of which defines a **type of object** that is acted on within the transformation, and the **actions** that are performed on objects of that type. The system provides a range of built-in default templates, and each **type of transformation** uses a specific subset of these templates. Typically, the transformation type and the subset of templates are tailored to the language the model is being transformed **into**. Some default templates within a set have no content; these are 'latent', and represent the potential for acting on an object that is not generally included in the transformation but is perfectly valid if you wanted to include it. For example, the **Linked Class Base** template in the **C#** transformation.

You can tailor the transformation templates in various ways:

- **Adjust** the code in one or more of the templates in a default set
- **Add** code to a 'latent' default template
- Add a new **custom template**, based on **one** of the defaults but serving a different purpose that you define
- Add a new **transformation type** containing - initially - a basic **set** of default templates
- Add (or remove) a stereotyped override for a template




A stereotyped override directs the transformation to use the modified template only if the element and/or feature are of the specified stereotyped types. If the object or feature are not of that type, the transformation applies the base template.

**Access**   **Tools | MDA Transformation Templates** (Ctrl+Alt+H)

### Edit Transformation Templates

Field/Option/ Button	Action	See also
<b>Language</b>	Click on the drop-down arrow and select the name of the transformation.	
<b>New Transformation Type</b>	Click on this button if you want to create a new transformation. A prompt displays for the name of the transformation; type in the name and click on the <b>OK</b> button.  The Templates list shows the default set of built-in templates, from which you can develop your transformation. Your custom transformation is <b>not saved or available</b> for use <b>unless</b> you add and/or edit one or more templates in the transformation.	
<b>Templates</b>	Lists the transformation templates for the current transformation.  Click on a template name to highlight it and display its content in the Template panel. The <b>Modified</b> column indicates whether you have edited the template for this transformation.	<a href="#">The Code Template Editor in MDG Development</a> <sup>[170]</sup>
<b>Template</b>	Displays the contents of the currently-selected template, and provides the editor facilities for modifying the template (right-click	<a href="#">Write Transformations</a>



Field/Option/ Button	Action	See also
	on the code text).	 <a href="#">The Code Template Editor in MDG Development</a>  <a href="#">Code Template Syntax</a> 
<b>Stereotype Overrides</b>	Lists the stereotyped templates, for the active base template.  The <b>Modified</b> column indicates whether you have modified a stereotyped template.	
<b>Add New Custom Template</b>	<p>Click on this button to create a custom template to add to the current transformation.</p> <p>A dialog displays, prompting you to specify:</p> <ul style="list-style-type: none"> <li>The object type (base template type) that this new template will respond to - click on the drop-down arrow and select the name (custom template types are not included in this list)</li> <li>The name of the new template - type in the appropriate text</li> </ul> <p>Click on the <b>OK</b> button. The new template name is added to the list of templates, and it is opened in the template editor ready for you to add its code.</p>	
<b>Add New Stereotyped Override</b>	<p>Click on this button to add a new stereotype override for the currently selected template. A dialog displays, prompting you to specify the:</p> <ul style="list-style-type: none"> <li>Base Class (stereotyped Class type - click on the drop-down arrow and click on the type in the list) and/or</li> <li>Feature (click on the drop-down arrow and click on the stereotyped feature in the list)</li> </ul> <p>Click on the <b>OK</b> button. The override is added to the <b>Stereotype Overrides</b> list.</p>	
<b>Get Default Template</b>	<p>Click on this button to update the editor display with the <b>default version</b> of the current <b>built-in</b> template or to <b>clear the content</b> of the current <b>custom</b> template.</p> <p>If you have saved the changed template, re-instating the default version is a change, so the <b>Modified</b> field still displays the word <b>Yes</b>.</p>	
<b>Save</b>	Click on this button to save the new or edited current template. You	

Field/Option/ Button	Action	See also
	cannot switch to another template without saving the current template, so this effectively saves the transition as well.	
<b>Delete</b>	Click on this button to delete the current <b>custom</b> template or stereotype override, or the most recent <b>changes</b> to a built-in template (effectively returning it to the default, base content). You cannot <b>delete</b> a <b>built-in</b> template.  You are prompted to confirm the deletion.	
<b>Help</b>	Click on this button to display this Help topic.	

**Notes**

- Transformation template editing is based very strongly on code generation template editing; for additional information on editing transformation templates see the *Code Template Editor* section and the *Editing Source Code* topic

**Learn more**

- [Code Template Framework Tool](#)<sup>[228]</sup> (Using)
- [Code Template Framework](#)<sup>[163]</sup> (Developing and Editing)
- [Code Template Editor](#)<sup>[164]</sup>
- [Editing Source Code](#)<sup>[214]</sup>
- [Export Code Generation and Transformation Templates](#)<sup>[163]</sup>
- [Import Code Generation and Transformation Templates](#)<sup>[163]</sup>

**Learning Center topics**

- (Alt+F1) | **Enterprise Architect | Model Transformations | Transform | Write Transforms**





## 10.4 Write Transformations

In Enterprise Architect, you have the facility to **write your own transformations**. Transformations are based on the Code Generation Template Framework, therefore it is suggested that you read and understand the topics covering **Code Generation Templates** prior to using the Transformation Template language.

**Access** [Tools | MDA Transformation Templates](#) (Ctrl+Alt+H)

### Factors concerning Transformation Templates

Factor	Detail	See also
<b>Default Transformation Templates</b>	Enterprise Architect provides a set of default transformation templates that you can use 'as is' or customize to your requirements.	<a href="#">Default Transformation Templates</a> <sup>[2052]</sup>
<b>General Syntax for the Intermediary Language</b>	Transformations in Enterprise Architect generate an intermediary code form of the model being created in the transformation. You can review and edit this code.	<a href="#">Intermediary Language</a> <sup>[2053]</sup>
<b>Intermediary Language Debugging</b>	You can also debug transformation scripts by checking the intermediary code generated from the Transform script.	<a href="#">Intermediary Language Debugging</a> <sup>[2053]</sup>
<b>Editing transformation templates and code</b>	When writing transformations, you use the facilities of the common Code Editor.	<a href="#">Editing Source Code</a> <sup>[2146]</sup>
<b>Code Template Framework</b>	You use the Code Template Framework to perform forward engineering of UML models. The Transformation Template Framework is derived from this.	<a href="#">Code Template Framework Tool</a> <sup>[2281]</sup> (Using) <a href="#">Code Template Framework</a> <sup>[1631]</sup> (Modifying)
<b>Syntax for Creating Objects</b>	To generate objects or elements in a transformation, you apply a specific syntax in the template script.	<a href="#">Objects</a> <sup>[2055]</sup>
<b>Syntax for Creating Connectors</b>	To generate connectors (relationships) in a transformation, you also apply a specific syntax in the template script.	<a href="#">Connectors</a> <sup>[2060]</sup> <a href="#">Transform Connectors</a> <sup>[2063]</sup>
<b>Transforming Duplicate Information</b>	In many transformations there is a substantial amount of information to be copied. Rather than place this information in the template, you can use macros to read it from its source.	<a href="#">Copy Information</a> <sup>[2066]</sup>
<b>Transforming</b>	In a transformation template, if you are transforming	<a href="#">Transform Template</a>

Factor	Detail	See also
<b>Template Parameter Substitutions</b>	Template Binding connector binding parameter substitutions, you can use the Template Parameter substitution macros.	<a href="#">Parameter Substitution</a>  <sup>2065</sup>
<b>Converting Types</b>	You can apply various methods for converting data types to different target platform types.	<a href="#">Convert Types</a>  <sup>2066</sup>
<b>Converting Names</b>	You can apply various methods for converting names of elements to different target platform naming conventions.	<a href="#">Convert Names</a>  <sup>2067</sup>
<b>Cross References</b>	During a transformation, you can perform cross verification of transformed elements.	<a href="#">Cross References</a>  <sup>2068</sup>

#### Notes

- Further hints and tips can be gleaned from a close study of the Transformation Templates provided with Enterprise Architect
- The Transformation Template editor applies the facilities of the common Code Editor

#### Learn more

- [Code and Transform Templates](#)  <sup>1632</sup>

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Model Transformations | Transform | Write Transforms**

### 10.4.1 Default Transformation Templates

Transformation templates provide the ability to represent the existing information in a model in a modified way. When creating a new transformation Enterprise Architect provides a **default** set of transformation templates that perform a direct copy of the source model to the target model. This allows you to think in terms of how the source model and target model are different. For each template you are able to prevent properties from being copied and add additional information until the appropriate target model is created.

You can list and examine the default templates in the Transformation Editor. The combination of default templates varies according to the language you are transforming.

Access   **Tools | MDA Transformation Templates (Ctrl+Alt+H)**

#### Notes

- When creating a new transformation you must modify at least **one** template before the new transformation becomes available

Learn more

- [Write Transformations](#) <sup>[2051]</sup>
- [Edit Transformation Templates](#) <sup>[2048]</sup>
- [Copy Information](#) <sup>[2066]</sup>

### 10.4.2 Intermediary Language

All transformations in Enterprise Architect create an **intermediary language** form of the model to generate. You can access and edit the file containing this intermediary language code using an external editor. Each object is represented in this language by the **object type** (for example, Class, Action, Method, Generalization or Tag) followed by the **object properties** and the **features** that it is made from; the grammar of the object description resembles this:

```

element :
    elementName { (elementProperty | element)* }

elementProperty:
    packageName
    stereotype
    propertyName = " propertyValueSymbol * "

packageName:
    name = " propertyValueSymbol * " ( . " propertyValueSymbol * " ) *
stereotype:
    stereotype = " propertyValueSymbol * " ( , " propertyValueSymbol * "
) *

propertyValueSymbol :
    \ \
    \ "
    Any character except " (U+0022), \ (U+005C)

```

- *elementName* is any one of the set of element types
- *propertyName* is any one of the set of properties

Literal strings can be included in property values by 'escaping' a quote character:

```
default = "\" Some string value.\""
```

Learn more

- [Intermediary Language Debugging](#) <sup>[2053]</sup>
- [Objects](#) <sup>[2055]</sup>
- [Connectors](#) <sup>[2060]</sup>

### 10.4.3 Intermediary Language Debugging

The script from an MDA template produces intermediate language text. However, on generating the model this script could return errors. When an error occurs, you can **view and debug** the generated text externally, preferably in an editor that prompts on updates to the file alterations.

**Access** **Project Browser Package context menu | Advanced | Transform Package** (Ctrl+Shift+H)

**Example**

For a MySQL database, the template code might resemble the following:

```
$enumFieldname = "test"

Column
{
    name=%qt%%CONVERT_NAME($enumFieldname, "Pascal Case", "Camel Case")%%qt%
    type=%qt%%CONVERT_TYPE(genOptDefaultDatabase, "Enum")%%qt%
}
```

This returns the output in the generated text file as:

```
Column
{
    name = "test"
    type = "ENUM"
}
```

If there is an error in the original transform, such as a spelling error - 'Colum' - clicking the **Do Transform** button returns an error message referring to the first line of intermediate code that includes the error 'Colum'.

**Debug when errors are returned on generating altered code**

Step	Description	See also
1	Select the Package to be transformed, and the <b>Transform Package</b> context menu. The Model Transformations dialog displays.	<a href="#">Transform Elements</a> <sup>2017</sup>
2	In the <b>Name</b> column, select the checkbox against the type of transformation being altered.	
3	In the <b>Intermediary File</b> field, click on the ( ... ) button and set the file location into which to generate the code.	
4	Select the <b>Write Always</b> checkbox, and click on the <b>Write Now</b> button to generate the script.  This only generates the script, not the model.	
5	If an error is returned specifying the line number of the problem, open the file in an external Code Editor (with Line Numbering) and locate the line number of problem.	
6	Alter the template code to correct the error.	

Step	Description	See also
7	Click on the <b>Do Transform</b> button to check that the alteration has corrected the problem.	

#### 10.4.4 Objects

**Objects** are generated in a transformation as text in the following form:

```
objectType
{
    objectProperties*
    XRef{xref}*
    Tag{tag}*
    Attribute{attributes}*
    Operation{operations}*
    Classifier{classifiers}*
    Parameter{parameters}*
}
```

For example:

```
Class
{
    name = "Example"
    language = "C++"
    Tag
    {
        name = "defaultCollectionClass"
        value = "List"
    }
    Attribute
    {
        name = "count"
        type = "int"
    }
}
```

Every object created in a transformation should include an **XRef** syntax element (see the end of this topic), as it helps the system to synchronize with the object and makes it possible to create a connector to that Class in the transformation.

#### Syntax elements in the code

Element	Detail	See also
<b>objectType</b>	<p>objectType is <b>one</b> of these:</p> <ul style="list-style-type: none"> <li>• <i>Action</i></li> <li>• <i>ActionPin</i></li> <li>• <i>Activity</i></li> <li>• <i>ActivityParameter</i></li> <li>• <i>ActivityPartition</i></li> <li>• <i>ActivityRegion</i></li> </ul>	

Element	Detail	See also
	<ul style="list-style-type: none"> <li>• <i>Actor</i></li> <li>• <i>Association</i></li> <li>• <i>Change</i></li> <li>• <i>Class</i></li> <li>• <i>Collaboration</i></li> <li>• <i>CollaborationUse</i></li> <li>• <i>Component</i></li> <li>• <i>DeploymentSpecification</i></li> <li>• <i>DiagramFrame</i></li> <li>• <i>Decision</i></li> <li>• <i>EntryPoint</i></li> <li>• <i>Event</i></li> <li>• <i>ExceptionHandler</i></li> <li>• <i>ExecutionEnvironment</i></li> <li>• <i>ExitPoint</i></li> <li>• <i>ExpansionNode</i></li> <li>• <i>ExpansionRegion</i></li> <li>• <i>ExposedInterface</i></li> <li>• <i>GUIElement</i></li> <li>• <i>InteractionFragment</i></li> <li>• <i>InteractionOccurrence</i></li> <li>• <i>InteractionState</i></li> <li>• <i>Interface</i></li> <li>• <i>InterruptibleActivityRegion</i></li> <li>• <i>Issue</i></li> <li>• <i>Iteration</i></li> <li>• <i>Object</i></li> <li>• <i>ObjectNode</i></li> <li>• <i>MessageEndpoint</i></li> <li>• <i>Node</i></li> <li>• <i>Package</i></li> <li>• <i>Parameter</i></li> <li>• <i>Part</i></li> <li>• <i>Port</i></li> <li>• <i>ProvidedInterface</i></li> <li>• <i>RequiredInterface</i></li> <li>• <i>Requirement</i></li> <li>• <i>Sequence</i></li> <li>• <i>State</i></li> </ul>	



Element	Detail	See also
	<ul style="list-style-type: none"> <li>• <i>StateMachine</i></li> <li>• <i>StateNode</i></li> <li>• <i>Synchronization</i></li> <li>• <i>Table</i></li> <li>• <i>TimeLine</i></li> <li>• <i>Trigger</i></li> <li>• <i>UMLDiagram</i></li> <li>• <i>UseCase</i></li> </ul>	
<b>objectProperties</b>	<p>object Properties is <b>zero</b>, or <b>one instance</b> of <b>one</b> or <b>more</b> of these:</p> <ul style="list-style-type: none"> <li>• <i>Abstract</i></li> <li>• <i>Alias</i></li> <li>• <i>Arguments</i></li> <li>• <i>Author</i></li> <li>• <i>Cardinality</i></li> <li>• <i>Classifier</i></li> <li>• <i>Complexity</i></li> <li>• <i>Concurrency</i></li> <li>• <i>Filename</i></li> <li>• <i>Header</i></li> <li>• <i>Import</i></li> <li>• <i>IsActive</i></li> <li>• <i>IsLeaf</i></li> <li>• <i>IsRoot</i></li> <li>• <i>IsSpecification</i></li> <li>• <i>Keyword</i></li> <li>• <i>Language</i></li> <li>• <i>Multiplicity</i></li> <li>• <i>Name</i></li> <li>• <i>Notes</i></li> <li>• <i>Persistence</i></li> <li>• <i>Phase</i></li> <li>• <i>Scope</i></li> <li>• <i>Status</i></li> <li>• <i>Stereotype</i></li> <li>• <i>Version</i></li> <li>• <i>Visibility</i></li> </ul>	

Element	Detail	See also
<b>Attribute</b>	<p><i>Attribute</i> has the same structure as <i>object Type</i>, and includes these properties:</p> <ul style="list-style-type: none"> <li>• <i>Alias</i></li> <li>• <i>Classifier</i></li> <li>• <i>Collection</i></li> <li>• <i>Container</i></li> <li>• <i>Containment</i></li> <li>• <i>Constant</i></li> <li>• <i>Default</i></li> <li>• <i>Derived</i></li> <li>• <i>LowerBound</i></li> <li>• <i>Name</i></li> <li>• <i>Notes</i></li> <li>• <i>Ordered</i></li> <li>• <i>Scope</i></li> <li>• <i>Static</i></li> <li>• <i>Stereotype</i></li> <li>• <i>Type</i></li> <li>• <i>UpperBound</i></li> <li>• <i>Volatile</i></li> </ul> <p><i>Attribute</i> also includes these elements:</p> <ul style="list-style-type: none"> <li>• <i>Classifier</i></li> <li>• <i>Tag</i></li> <li>• <i>XRef</i></li> </ul>	
<b>Operation</b>	<p><i>Operation</i> has the same structure as <i>object Type</i>, and includes these properties:</p> <ul style="list-style-type: none"> <li>• <i>Abstract</i></li> <li>• <i>Alias</i></li> <li>• <i>Behavior</i></li> <li>• <i>Classifier</i></li> <li>• <i>Code</i></li> <li>• <i>Constant</i></li> <li>• <i>IsQuery</i></li> <li>• <i>Name</i></li> <li>• <i>Notes</i></li> <li>• <i>Pure</i></li> </ul>	

Element	Detail	See also
	<ul style="list-style-type: none"> <li>• <i>ReturnArray</i></li> <li>• <i>Scope</i></li> <li>• <i>Static</i></li> <li>• <i>Stereotype</i></li> <li>• <i>Type</i></li> </ul> <p>Operation also includes these elements:</p> <ul style="list-style-type: none"> <li>• <i>Classifier</i></li> <li>• <i>Parameter</i></li> <li>• <i>Tag</i></li> <li>• <i>XRef</i></li> </ul>	
<b>Parameter</b>	<p>Parameter has the same structure as object Type, and includes the Tag element and these properties:</p> <ul style="list-style-type: none"> <li>• <i>Classifier</i></li> <li>• <i>Default</i></li> <li>• <i>Fixed</i></li> <li>• <i>Name</i></li> <li>• <i>Notes</i></li> <li>• <i>Kind</i></li> <li>• <i>Stereotype</i></li> </ul>	
<b>Tag</b>	<p>Tag has these properties:</p> <ul style="list-style-type: none"> <li>• <i>Name</i></li> <li>• <i>Value</i></li> </ul>	

### Special Cases

Certain types of object have variations of the object definition syntax.

Object	Detail	See also
<b>Packages</b>	<p>Packages differ from other objects in the following ways:</p> <ul style="list-style-type: none"> <li>• They have a reduced set of properties: <i>alias</i>, <i>author</i>, <i>name</i>, <i>namespaceRoot</i>, <i>notes</i>, <i>scope</i>, <i>stereotype</i> and <i>version</i></li> <li>• The property <i>namespaceRoot</i> is only given to Packages</li> <li>• A <i>name</i> must be specified for each Package</li> <li>• The <i>name</i> property can be a qualified name; when a qualified name is specified, the properties given are applied only to the final package</li> </ul>	

Object	Detail	See also
	<ul style="list-style-type: none"> <li>Only Packages can contain other packages</li> <li>Packages can't contain attributes and operations</li> </ul>	
<b>XRef</b>	<p>Cross references are defined using the transform statements. The properties include:</p> <ul style="list-style-type: none"> <li><i>Namespace</i></li> <li><i>Name</i></li> <li><i>Source</i></li> <li><i>Notes</i></li> </ul>	<a href="#">Cross References</a> <small>[2068]</small>
<b>Tables</b>	<p>Tables are a special type of object, with the following differences from other object types:</p> <ul style="list-style-type: none"> <li>They can include columns and primary keys</li> <li>They cannot include attributes</li> </ul>	<a href="#">Foreign Key</a> <small>[2068]</small>
<b>Columns</b>	<p>Columns are similar to attributes, but have an <i>autonumber</i> element containing <i>Startnum</i> and its increment, and the following added properties:</p> <ul style="list-style-type: none"> <li><i>Length</i></li> <li><i>NotNull</i></li> <li><i>Precision</i></li> <li><i>PrimaryKey</i></li> <li><i>Scale</i></li> <li><i>Unique</i></li> </ul> <p>In the column definition, you cannot assign a value to the <i>NotNull</i>, <i>PrimaryKey</i> or <i>Unique</i> properties.</p>	

### 10.4.5 Connectors

The process of creating **connectors** in a transformation has the same form as for creating elements. It is a little more complex, because you also define each **end** of the connector.

Connectors are represented in the Intermediary language as:

```

Connector Type
{
    connector Properties*
    AssociationClass {associationClassProperties*}
    Source {sourceProperties*}
    Target {targetProperties*}
}

```

For example:

```


Association
{
  name="anAssociation"
  stereotype=""
  direction="Unspecified"
  Source
  {
    access="Private"
    navigability="Unspecified"
  }
  Target
  {
    access="Private"
    multiplicity="1..*"
  }
}

```

#### Syntax elements in the code

Element	Detail	See also
<b>ConnectorType</b>	<p>Connect or Type is one of these:</p> <ul style="list-style-type: none"> <li>• Abstraction</li> <li>• Aggregation</li> <li>• Assembly</li> <li>• Association</li> <li>• Collaboration</li> <li>• ControlFlow</li> <li>• Connector</li> <li>• Delegate</li> <li>• Dependency</li> <li>• Deployment</li> <li>• ForeignKey</li> <li>• Generalization</li> <li>• InformationFlow</li> <li>• Instantiation</li> <li>• Interface</li> <li>• InterruptFlow</li> <li>• Manifest</li> <li>• Nesting</li> <li>• NoteLink</li> <li>• ObjectFlow</li> <li>• Package</li> <li>• Realization</li> <li>• Sequence</li> <li>• Substitution</li> </ul>	

Element	Detail	See also
	<ul style="list-style-type: none"> <li>• TemplateBinding</li> <li>• Transition</li> <li>• Usage</li> <li>• UseCase</li> <li>• Uses</li> </ul>	
<b>connectorProperties</b>	<p><code>connectorProperties</code> is <b>zero</b>, or <b>one</b> instance of <b>one or more</b> of these:</p> <ul style="list-style-type: none"> <li>• alias</li> <li>• direction</li> <li>• notes</li> <li>• name</li> <li>• stereotype</li> <li>• XRef</li> </ul>	
<b>associationClassProperties</b>	<p><code>associationClassProperties</code> are one instance of these:</p> <ul style="list-style-type: none"> <li>• Classifier</li> <li>• XRef</li> </ul>	
<b>sourceProperties</b> <b>targetProperties</b>	<p><code>sourceProperties</code> and <code>targetProperties</code> are each a reference to an element and <b>zero</b>, or <b>one</b> instance of <b>one or more</b> of these:</p> <ul style="list-style-type: none"> <li>• aggregation</li> <li>• alias</li> <li>• allowduplicates</li> <li>• changeable</li> <li>• constraint</li> <li>• containment</li> <li>• navigability</li> <li>• memberType</li> <li>• multiplicity</li> <li>• Notes</li> <li>• ordered</li> <li>• qualifier</li> <li>• role</li> <li>• scope</li> <li>• stereotype</li> </ul>	
<b>Element Reference</b>	An element reference is either a <b>guid</b> that references an	

Element	Detail	See also
	<p>element that already exists before the transformation, or an <b>XRef</b> to reference an element that is created by a transformation.</p> <ul style="list-style-type: none"> <li>• guid</li> <li>• XRef</li> </ul>	<a href="#">Cross References</a>  <small>[2068]</small>

### Notes

- Each connector is transformed at both end objects, therefore the connector might appear twice in the transformation; this is not a problem, although you should check carefully that the connector is generated exactly the same way, regardless of which end is on the current Class

### Learn more

- [Transform Connectors](#)  [2063]
- [Transform Foreign Keys](#)  [2065]

## 10.4.6 Transform Connectors

You can use two different types of Class as a connector end: a Class **created by a transformation**, or an **existing** Class for which you already know the **GUID**.

### Connect to a Class Created by a Transformation

The most common connection is to a Class created by a transformation; to create this connection you use three items of information:

- The original Class GUID
- The name of the transformation
- The name of the transformed Class

This type of connector is created using the **TRANSFORM\_REFERENCE** function macro; when the element is in the current transformation, it can be safely omitted from the transformation. The simplest example of this is when you have created multiple Classes from a single Class in a transformation, and you want a connector between them; consider this script from the EJB Entity transformation:

```
Dependency
{
  %TRANSFORM_REFERENCE( " EJBReal i zeHome" , cl assGUI D) %
st er eot ype=" EJBReal i zeHome"
  Sour ce
  {
    %TRANSFORM_REFERENCE( " EJBEnt i tyBean" , cl assGUI D) %
  }
  Tar get
  {
    %TRANSFORM_REFERENCE( " EJBHomeI nt er f ace" , cl assGUI D) %
  }
}
```

In this script there are three uses of the **TRANSFORM\_REFERENCE** macro: one to identify the connector for

synchronization purposes and the other two to identify the ends; all three use the same source GUID, because they all come from the one original Class. None of the three have to specify the transformation because the two references are to something within the current transformation - each of them then only has to identify the transform name.

It is also possible to create a connector from another connector. You can create a connector template and list all connectors connected to a Class from the Class level templates; you don't have to worry about only generating the connector once, because if you have created a **TRANSFORM\_REFERENCE** for the connector then the system automatically synchronizes them.

This script copies the source connector:

```
%connector Type%
{
    %TRANSFORM_CURRENT( ) %
    %TRANSFORM_REFERENCE( " Connector " , connect or GUI D) %

    Source
    {
        %TRANSFORM_REFERENCE( " Cl ass " , connect or SourceGUI D) %
        %TRANSFORM_CURRENT( " Source " ) %
    }
    Target
    {
        %TRANSFORM_REFERENCE( " Cl ass " , connect or Dest GUI D) %
        %TRANSFORM_CURRENT( " Target " ) %
    }
}
```

### Connecting to a Class for which you know the GUID

The second type of Class that you can use as a connector end is an **existing** element for which you know the current **GUID**. To create this connection, specify the GUID of the target Class in either the source or target end; this script creates a Dependency **from** a Class created in a transformation, **to** the Class it was transformed from:

```
Dependency
{
    %TRANSFORM_REFERENCE( " SourceDependency " , cl assGUI D) %
    stereotype="transformedFrom"
    Source
    {
        %TRANSFORM_REFERENCE( " Cl ass " , cl assGUI D) %
    }
    Target
    {
        GUI D=%qt %%cl assGUI D%%qt %
    }
}
```

### Notes

- Each connector is transformed at both end objects, therefore the connector might appear twice in the transformation; this is not a problem, although you should check carefully that the connector is generated exactly the same way, regardless of which end is on the current Class

### Learn more

- [Cross References](#) 



- [Copy Information](#)<sup>[2066]</sup>
- [Connectors](#)<sup>[2060]</sup>

### 10.4.7 Transform Foreign Keys

**ForeignKey** is a special case where not just a connector is created - you must also list the **columns** involved in the transformation. In addition, **tags** specified on the connector are actually created on the foreign key operation in the source Class, and you can add a cascade property:

```
cascade="update", "delete"
```

#### Example

This example is from the DDL transformation **Connector** template. For the template:

```
ForeignKey
{
  %TRANSFORM_REFERENCE( " FK1" , connector GUI D) %
  Source
  {
    %TRANSFORM_REFERENCE( " LinkTable" , connector GUI D) %
    name=%qt %%connector Dest ElementName%%qt %
    Column
    {
      name=%qt %%CONVERT_NAME( connector Dest ElementName, " Pascal
Case", " Camel Case" ) %d %qt %
      type=%qt %%CONVERT_TYPE( genOpt Default Database, " Integer" ) %%qt %
    }
  }
  Target
  {
    %TRANSFORM_REFERENCE( " Table" , connector Dest ElementGUI D) %
    Column
    {
      name=%qt %%CONVERT_NAME( connector Dest ElementName, " Pascal
Case", " Camel Case" ) %d %qt %
      type=%qt %%CONVERT_TYPE( genOpt Default Database, " Integer" ) %%qt %
    }
  }
}
```

This code is generated for a Foreign Key:

```
ForeignKey
{
  XRef { namespace=" DDL" name=" FK1" source="{ 18ED24EA- 0669- 499c- A066-
FABEE6772ED1}" }
  Source
  {
    XRef { namespace=" DDL" name=" Table" source="{ CC6A084C- 9B56- 4a1e-
8196- C66A74E59B66}" }
    name=" FK_account "
    multiplicity=" 0..*"
    Column
    {
      name=" account "
      type=" I NTEGER"
    }
  }
  Target
  {
    XRef { namespace=" DDL" name=" Table" source="{ 791AEAB5- 7831- 4d2a-
92C7- 198C10B31AF0}" }
    multiplicity=" 1"
  }
}
```

```

Column
{
    name="accountID"
    type="INTEGER"
}
}

```

### 10.4.8 Copy Information

In many transformations there is a substantial amount of information to be copied.

It would be tedious to type all of the common information into a template so that it is copied to the transformed Class; the alternative is to use the **TRANSFORM\_CURRENT** and **TRANSFORM\_TAGS** function macros.

#### Use of Macros

Objective	Detail	See also
<b>Copy Object</b>	<b>TRANSFORM_CURRENT(&lt;listOfExcludedItems&gt;)</b>  The function generates an exact copy of all the properties of the current item, except for the items named in <listOfExcludedItems>.	<a href="#">Objects</a> <sup>[2055]</sup>
<b>Copy Connector</b>	Another form of the function is available when transforming connectors to copy either end of the connector:  <b>TRANSFORM_CURRENT(&lt;connectorEnd&gt;,&lt;listOfExcludedItems&gt;)</b>  This generates an exact copy of the connector end specified by <connectorEnd> (either <b>Source</b> or <b>Target</b> ) except for the items named in <listOfExcludedItems>.	<a href="#">Connector</a> <sup>[2060]</sup>
<b>Copy Tags</b>	<b>TRANSFORM_TAGS(&lt;listOfExcludedItems&gt;)</b>  The function generates an exact copy of all the Tagged Values of the current item, except for the items named in <listOfExcludedItems>.	

### 10.4.9 Convert Types

Different target **platforms** almost certainly require different **data types**, so you usually require a method of converting between types. This is offered by the macro:

**CONVERT\_TYPE(<destinationLanguage>, <originalType>)**

This function converts <originalType> to the corresponding type in <destinationLanguage> using the datatypes and common types defined in the model, where <originalType> is assumed to be a platform independent common type.

A similar macro is available when transforming common datatypes to the datatypes for a specified database:

**CONVERT\_DB\_TYPE(<destinationDatabase>, <originalType>)**

This function converts *<originalType>* to the corresponding datatypes in *<destinationDatabase>*, which is defined in the model; *<originalType>* refers to a platform independent common datatype.

### 10.4.10 Convert Names

Different target **platforms** use different **naming conventions**, so you might not want to copy the names of your elements directly into the transformed models. To facilitate this requirement, the transformation templates provide a **CONVERT\_NAME** function macro.

Another way in which you can transform a name is to remove a prefix from the original name, with the **REMOVE\_PREFIX** macro.

#### CONVERT\_NAME <originalName>, <originalFormat>, <targetFormat>)

This macro converts *<originalName>*, which is assumed to be in *<originalFormat>*, to *<targetFormat>*.

The supported formats are:

- **Camel Case:** new words start with a capital letter except for the first word, which begins with a lower case letter; for example, *myVariableTable*
- **Pascal Case:** the same as Camel Case but the first letter of the first word is upper case; for example, *MyVariableTable*
- **Spaced:** words are separated by spaces; the case of letters is ignored
- **Underscored:** words are separated by underscores; the case of letters is ignored

The **original format** might also specify a list of delimiters to be used. For example a value of ' \_ ' breaks words whenever either a space or underscore is found. The **target format** might also use a format string that specifies the case for each word and a delimiter between them. It takes the following form:

*<firstWord>(<delimiter>)<otherWords>*

- *<firstWord>* controls the case of the first word (see below)
- *<delimiter>* is the string generated between words
- *<otherWords>* applies to all words after the first word

Both *<firstWord>* and *<otherWords>* are a sequence of two characters. The first character represents the case of the first letter of that word, and the second character represents the case of all subsequent letters. An upper case letter forces the output to upper case, a lower case letter forces the output to lower case, and any other character preserves the original case.

**Example 1:** To capitalize the first letter of each word and separate multiple words with a space:

" Ht ( ) Ht " to output " My Var i abl e Tabl e "

**Example 2:** To generate the equivalent of Camel Case, but reverse the roles of upper and lower case; that is, all characters are upper case except for the first character of each word *after* the first word:

" HT( ) hT" to output " MY vARI ABLE t ABLE "

#### REMOVE\_PREFIX(<originalName>, <prefixes>)

This macro removes any prefix found in *<prefixes>* from *<originalName>*. The prefixes are specified in a semi-colon separated list.

The macro is often used in conjunction with the **CONVERT\_NAME** macro. For example, this code creates a get property name according to the options for Java:

```
$propertyName=%REMOVE_PREFIX( attributeName, genOptPropertyPrefix) %
%if genOptGenCapitalisedProperties=="T" %
$propertyName=%CONVERT_NAME( $propertyName, "camel case", "pascal case") %
%endif %
```

#### Notes

- Acronyms are not supported when converting from Camel Case or Pascal Case

### 10.4.11 Cross References

**Cross References** are an important part of transformations. You can use them to:

- Find the transformed Class to synchronize with
- Create connectors between transformed Classes
- Specify a classifier of a type
- Determine where to transform to for future transformations

Each cross reference has three different parts:

- A **Namespace**, corresponding to the transformation that generated the element
- A **Name**, which is a unique reference to something that can be generated in the above transformation, and
- A **Source**, which is the GUID of the element that this element was created from

When writing the templates for a transformation it is easiest to generate the cross references using the macro defined for this purpose:

**TRANSFORM\_REFERENCE** (<name>, <sourceGuid>, <namespace>)

The three parameters are optional. The macro generates a reference that resembles the following.

```
XRef { namespace=" <namespace>" name=" <name>" source=" <sourceGuid>"
```

- If <name> is not specified the macro gets the name of the current template
- If <sourceGUID> is not specified the macro gets the GUID of the current Class
- If <namespace> is not specified the macro gets the name of the current transformation

The only time that this should be specified is when creating a connector to a Class created in a different transformation.

A good example of the use of cross references is in the **DDL transformation** provided with Enterprise Architect. In the **Class** template a cross reference is created with the name *Table*. Then up to two different connectors can be created, each of which must identify the two Classes it connects using cross references, while having its own unique cross reference.

#### Specify Classifiers

Objects, attributes, operations and parameters can all reference another element in the model as their type. When this type is created from a transformation you must use a cross reference to specify it, using the macro:

**TRANSFORM\_CLASSIFIER** (<name>, <sourceGuid>, <namespace>)

This macro generates a cross reference within a classifier element, where the parameters are identical to the **TRANSFORM\_REFERENCE** macro but the name *Classifier* is generated instead of *XRef*.

If the target classifier already exists in the model before the transformation, **TRANSFORM\_CLASSIFIER** is inappropriate, so instead the GUID can be given directly to a classifier attribute.

If a classifier is specified for any type, it overrides that type.

#### Learn more

- [DDL Transformation](#)<sup>[2025]</sup>
- [XRef element](#)<sup>[2055]</sup>

### 10.4.12 Transform Template Parameter Substitution

If you want to provide access in a transformation template to data concerning the transformation of a Template Binding connector's binding parameter substitution in the model, you can use the **Template Parameter substitution macros**.




#### Factors in the Transformation

Factor	Detail	See also
<b>Intermediary Language</b>	<p>Template Parameter Substitutions are represented in the Intermediary language as:</p> <pre> TemplateParameterSubstitution {     Formal { FormalProperties }     Actual { ActualProperties } } </pre> <p>For example:</p> <pre> TemplateParameterSubstitution {     Formal     {         name=%qt%%parameterSubstitutionFormal%%qt%     }     Actual     {         name=%qt%%parameterSubstitutionActual%%qt%         %TRANSFORM_CLASSIFIER("Class", parameterSubstitutionAc     } } </pre>	
<b>Formal Propertie</b>	<b>FormalProperties</b> and <b>ActualProperties</b> are zero, or one instance of one of the following properties:	

Factor	Detail	See also
<b>s or Actual Properties</b>	<ul style="list-style-type: none"> <li>name</li> <li>classifier</li> </ul>	
<b>Transform of Parameter Substitution on Actual parameter</b>	<p>If the Actual parameter is assigned a String Expression, it will transform as Actual name. You can assign the Actual Classifier if you know the GUID:</p> <pre> TemplateParameterSubstitution (     Formal     {         name=%qt%%parameterSubstitutionFormal%%qt%     }     Actual     {         name=%qt%%parameterSubstitutionActual%%qt%         classifier=%qt%%parameterSubstitutionActualClassifier%     } ) </pre> <p>If you want the Actual parameter to be transformed so that its Classifier is assigned with an element that is transformed, then use <b>TRANSFORM_CLASSIFIER</b> or <b>TRANSFORM_REFERENCE</b>, as shown:</p> <pre> TemplateParameterSubstitution {     Formal     {         name=%qt%%parameterSubstitutionFormal%%qt%     }     Actual     {         name=%qt%%parameterSubstitutionActual%%qt%         %TRANSFORM_CLASSIFIER("Class", parameterSubstitutionAc     } } </pre> <p>Or</p> <pre> TemplateParameterSubstitution {     Formal     {         name=%qt%%parameterSubstitutionFormal%%qt%     }     Actual     {         name=%qt%%parameterSubstitutionActual%%qt%         %TRANSFORM_REFERENCE("Class", parameterSubstitutionAct     } } </pre>	

**Learn more**

- [Write Transformations](#) <sup>[2051]</sup>

- [Template Parameter Substitution Macros](#) 
- [Template Binding](#) 
- [Parameter Substitution](#) 

# Part

---



XI



## 11 Software Engineering



Software Engineering is the process of designing, implementing and maintaining software. This section describes how Enterprise Architect can assist in this process by providing a design environment and tools including **automated code generation**, **reverse engineering** of source code and **synchronization** between the source code and model.

### Topics

Topic	Link
Software Design	<a href="#">GoF Patterns</a> <sup>[2285]</sup>
Development Environment	<a href="#">Development Tools</a> <sup>[2074]</sup> <a href="#">Code Build &amp; Debug</a> <sup>[2172]</sup> <a href="#">MDG Integration and Code Engineering</a> <sup>[2078]</sup>
Generate Source Code	<a href="#">Generate Source Code</a> <sup>[2111]</sup>
Importing Source Code	<a href="#">Importing Source Code</a> <sup>[2136]</sup>

### Learn more

- Enterprise Architect provides a set of tools that goes beyond traditional debugging capabilities, see [Visual Execution Analyzer](#) <sup>[2527]</sup>
- [Database Engineering](#) <sup>[2334]</sup> is also available, including keys, triggers, constraints, RI and other relational database features, for and from a range of database products
- Enterprise Architect provides technologies to support Service Oriented Architecture; see [SOA and XML Engineering](#) <sup>[2386]</sup>

## 11.1 Development Tools



### Topics

Topic	Detail	See also
<b>A Tightly Integrated Development Environment with Outstanding Tools and Functionality</b>	<ul style="list-style-type: none"> <li>• Design and elaborate your Class model in UML and other technologies</li> <li>• Generate code and reverse engineer existing code</li> <li>• Customize code generation with templates</li> <li>• Invoke compilers and other command line tools to build, run and deploy your application</li> <li>• Use the built in source code editor to write and maintain source files</li> <li>• Use the built in debugger to inspect and correct code at run time</li> <li>• Use the Profiler to fine tune Windows native C/C++ code</li> <li>• Use the visualization tools to generate Sequence diagrams from executing code</li> <li>• Use the testing capabilities to validate execution with Class- and method-level constraints and invariants</li> <li>• Link to junit and NUnit test cases</li> <li>• Link or integrate with Eclipse or Visual Studio when necessary to further enhance your development process</li> </ul>	
<b>Abstract</b>	<p>Enterprise Architect (Professional edition and above) provides an extraordinary range of features and a unique, rich, tightly coupled toolset for rapidly developing, visualizing, debugging, testing, maintaining and generally working with even the most complex source code.</p> <p>A wide range of standard programming languages are supported in design and reverse engineering; inbuilt editors with custom syntax highlighting and tight integration with the model provide a powerful and effective programming environment; coupled with the build/debug and testing capabilities, Enterprise Architect is both a remarkable analysis and design tool, and a formidable debug and testing environment.</p> <p>Enterprise Architect's Visual Execution Analyzer also supports Java, .NET and native Windows applications written in C/C++ using Microsoft's compilers; in addition to powerful tools for standard debugging, Enterprise Architect supports recording Sequence diagrams from executing code, capturing of method calls and conversion into custom call sets for testing purposes, validation of execution against State diagrams, and more.</p> <p>The Model Driven Development Environment (MDDE) provides one of the richest and most useful feature sets available for working with source code and ensuring the construction of robust and effective applications; the MDDE integrates code and model by providing options to either generate source code from the model or reverse engineer existing source code into a model - source</p>	

Topic	Detail	See also
	code and model can be synchronized in either direction.	
<b>Application Patterns</b>	Enterprise Architect provides complete starter projects, including model information, code and build scripts, for several basic application types including: <ul style="list-style-type: none"> <li>• MFC Windows applications</li> <li>• Java programs</li> <li>• ASP.NET web services</li> </ul>	<a href="#">Generate Application Pattern</a> <small>[2077]</small>
<b>Relationships of Software Engineering Units</b>	<pre> graph TD     Design[Design] --&gt; CG[Code Generation]     CG --&gt; SC1[/Source Code/]     SC1 --&gt; Import[Import]     Import --&gt; CE[Code Editing]     CE --&gt; Build[Build]     Build --&gt; Test[Test]     Build --&gt; Debug[Debug]     Build --&gt; Run[Run]     Build --&gt; Deploy[Deploy] </pre>	
<b>Supported Languages</b>	Popular languages supported include: <ul style="list-style-type: none"> <li>• C/ C++</li> <li>• Java</li> <li>• Microsoft .NET family</li> <li>• ADA</li> <li>• Python</li> <li>• Perl</li> </ul> <p>Toolboxes provide for different modeling technologies.</p>	

### Notes

- Although you can generate and reverse engineer code in a range of languages, Execution Analysis debugging and recording are supported for the following platforms / languages only:
  - Microsoft Windows Native C
  - Microsoft Windows Native C++
  - Microsoft Windows Visual Basic
  - Microsoft .NET Family (C#, J#, VB)

- Sun Microsystems Java

#### Learn more

- [Software Engineering](#) <sup>[2073]</sup>
- [Getting Started](#) <sup>[2173]</sup>
- [Analyzer Scripts](#) <sup>[2175]</sup>
- [Editing Source Code](#) <sup>[2146]</sup>
- [Debugging Actions](#) <sup>[2231]</sup>

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Software Engineering**

### 11.1.1 Overview of Development

Code Engineering with Enterprise Architect broadly encompasses various processes for generating or transforming code from your UML model and importing code into the model, to support model development in several coding languages, database development and SOA development.

#### Topics

Topic	Detail	See also
<b>Model Driven Code Engineering</b>	<ul style="list-style-type: none"> <li>• Source code generation and reverse engineering for many popular languages, including C++, C#, Java, Delphi, VB.Net, Visual Basic, ActionScript, Python and PHP</li> <li>• A built in 'syntax highlighting' source code editor</li> <li>• Code generation templates, which enable you to customize the generated source code to your company specifications</li> </ul>	<a href="#">Software Engineering</a> <sup>[2073]</sup>
<b>Transformations for Rapid Development</b>	<ul style="list-style-type: none"> <li>• Advanced Model Driven Architecture (MDA) transformations using transformation templates</li> <li>• Built-in transformations for DDL, C#, Java, EJB and XSD</li> <li>• One Platform Independent Model can be used to generate and synchronize multiple Platform Specific Models, providing a significant productivity boost</li> </ul>	<a href="#">Model Transformations - MDA</a> <sup>[2013]</sup>
<b>Visual Execution Analysis / Debugging, Verification and Visualization</b>	<ul style="list-style-type: none"> <li>• Execute build, test, debug, run and deploy scripts</li> <li>• Integrate UML development and modeling with source development and compilation</li> <li>• Generate NUnit and JUnit test Classes from source Classes using MDA Transformations</li> <li>• Integrate the test process directly into the Enterprise Architect IDE</li> </ul>	<a href="#">Visual Execution Analysis</a> <sup>[2527]</sup>

Topic	Detail	See also
	<ul style="list-style-type: none"> <li>• Debug .NET, Java and Microsoft Native (C, C++ and Visual Basic) applications</li> <li>• Design and execute Test suites based on <i>Programming by Contract</i> principles</li> </ul>	
<b>Database Modeling</b>	<p>Enterprise Architect enables you to:</p> <ul style="list-style-type: none"> <li>• Reverse engineer from many popular DBMSs, including SQL Server, My SQL, Access, PostgreSQL and Oracle 9i, 10g or 11g</li> <li>• Model database tables, columns, keys, foreign keys and complex relationships using UML and an inbuilt data modeling profile</li> <li>• Forward generate DDL scripts to create target database structures</li> </ul>	<a href="#">Database Engineering</a> <sup>[2334]</sup>
<b>XML Technology Engineering</b>	<p>Enterprise Architect enables you to rapidly model, forward engineer and reverse engineer two key W3C XML technologies:</p> <ul style="list-style-type: none"> <li>• XML Schema (XSD)</li> <li>• Web Service Definition Language (WSDL)</li> </ul> <p>XSD and WSDL support is critical for the development of a complete <b>Service Oriented Architecture</b> (SOA), and the coupling of UML 2.4.1 and XML provides the natural mechanism for implementing XML-based SOA artifacts within an organization.</p>	<a href="#">SOA and XML</a> <sup>[2386]</sup> <a href="#">XML Schema - XSD</a> <sup>[2387]</sup> <a href="#">Web Services - WSDL</a> <sup>[2423]</sup>

### 11.1.2 Application Patterns (Model + Code)



To get you going with a code based project as fast as possible, Enterprise Architect enables you to generate starter projects including model information, code and build scripts for one of several basic application types. Patterns include:

- MFC Windows applications
- Java programs
- ASP.NET web services

**Access** **Project | New Model (Ctrl+Shift+M) > Application Patterns**  
**Project Browser | Package context menu | Add a Model using Wizard > Application Patterns**

#### Reference

Field	Usage	See also
<b>Technology</b>	Select the appropriate technology.	

Field	Usage	See also
<b>Name</b>	Displays the application patterns available for the selected technology; select the required pattern to import.	
description field	Displays a description of the selected pattern.	
<b>Destination folder</b>	Browse for and select the directory in which to load the source code for the application.	
<b>Use Local Path</b>	Enable the selection of an existing local path to place the source code under; changes the <b>Destination folder</b> field to a drop-down selection.	
<b>Compiler command</b>	Displays the default compiler command path for the selected technology; you must either: <ul style="list-style-type: none"> <li>• Confirm that the compiler can be found at this path, or</li> <li>• Edit the path to the compiler location</li> </ul>	<a href="#">Local Paths</a>  <a href="#">Local Paths Dialog</a> 
<b>Edit Local Paths</b>	Many application patterns specify their compiler using a local path.  The first time you use any pattern you must click on this button to ensure the local path points to the correct location.  The Local Paths dialog displays.	

### Notes

- If required, you can define custom application patterns by adding files to the *AppPatterns* directory where Enterprise Architect is installed; top level directories are listed as Technologies and can contain an icon file to customize the icon displayed for the technology

Directories below this are defined as groups in the patterns list; the patterns are defined by the presence of four files with a matching name: a zip file (.zip), XML file (.xml), config file (.cfg) and optional icon (.ico)

- The *config* file supports the following fields:
  - [provider], [language], [platform], [url], [description], [version] - all displayed in the description field
  - [xmrootpaths] - the root path of the source code in the exported xmi; this is replaced with the selected destination folder when the user applies the application pattern

### Learn more

- [Model Wizard](#) 

## 11.2 MDG Integration and Code Engineering

### Topics

Topic	Detail	See also
<b>Abstract/Usage</b>	<p>MDG Integration for Eclipse and MDG Integration for Visual Studio are standalone products that provide an enhanced code engineering functionality between Enterprise Architect and the development environments.</p> <p>The MDG Integration programs provide a lightweight bridge between Enterprise Architect and the development environment, offering enhanced code generation, reverse engineering and synchronization between code and the UML model; merging changes can be achieved with minimal effort, and navigation between model and source code is significantly enhanced.</p> <p>Trial versions of MDG Integration for Eclipse and MDG Integration for Visual Studio can be downloaded from the Sparx Systems website.</p>	<p><a href="http://www.sparxsystems.com/products/mdg/int/eclipse/index.html">www.sparxsystems.com/products/mdg/int/eclipse/index.html</a> (Online Resource)</p> <p><a href="http://www.sparxsystems.com/products/mdg/int/vs/index.html">www.sparxsystems.com/products/mdg/int/vs/index.html</a> (Online Resource)</p>

## 11.3 Modeling Conventions



In order to get the most out of the round trip engineering in Enterprise Architect, you must be familiar with the modeling conventions used when generating and reverse engineering the languages you use. This topic describes the stereotypes, Tagged Values and other conventions used in code engineering in Enterprise Architect for the following supported languages:

Topic	Link
Action Script	<a href="#">ActionScript</a> <sup>[2081]</sup>
Ada 2005 (Systems Engineering and Ultimate editions)	<a href="#">Ada 2005</a> <sup>[2082]</sup>
C	<a href="#">C</a> <sup>[2084]</sup>
C#	<a href="#">C#</a> <sup>[2088]</sup>
C++	<a href="#">C++</a> <sup>[2090]</sup>
Delphi	<a href="#">Delphi</a> <sup>[2095]</sup>
Java	<a href="#">Java</a> <sup>[2096]</sup>
PHP	<a href="#">PHP</a> <sup>[2099]</sup>
Python	<a href="#">Python</a> <sup>[2100]</sup>
SystemC (Systems Engineering and Ultimate editions)	<a href="#">SystemC</a> <sup>[2100]</sup>
Verilog (Systems Engineering and Ultimate editions)	<a href="#">Verilog</a> <sup>[2105]</sup>
VHDL (Systems Engineering and Ultimate editions)	<a href="#">VHDL</a> <sup>[2106]</sup>
Visual Basic	<a href="#">Visual Basic</a> <sup>[2109]</sup>
Visual Basic .NET	<a href="#">Visual Basic .NET</a> <sup>[2102]</sup>

### Notes

- Enterprise Architect incorporates a number of visibility indicators or scope values for its supported languages; these include, for:
  - All languages - Public (+), Protected (#) and Private (-)
  - Java - Package (~)
  - Delphi - Published (^)
  - C# - Internal (~), Protected Internal (^)
  - ActionScript - Internal (~)
  - VB.NET - Friend (~), Protected Friend (^)
  - PHP - Package (~)



- Python - Package (~)
- C - Package (~)
- C++ - Package (~)

### 11.3.1 ActionScript Conventions

Enterprise Architect supports round trip engineering of ActionScript 2 and 3, where the following conventions are used.

#### Stereotypes

Stereotype	Applies To	Corresponds To
<b>literal</b>	Operation	A literal method referred to by a variable.
<b>property get</b>	Operation	A read property.
<b>property set</b>	Operation	A write property.

#### Tagged Values

Tag	Applies To	Corresponds To
<b>attribute_name</b>	Operation with stereotype <i>property get</i> or <i>property set</i>	The name of the variable behind this property.
<b>dynamic</b>	Class or Interface	The <i>dynamic</i> keyword.
<b>final</b>	ActionScript 3: Operation	The <i>final</i> keyword.
<b>intrinsic</b>	ActionScript 2: Class	The <i>intrinsic</i> keyword.
<b>namespace</b>	ActionScript 3: Class, Interface, Attribute, Operation	The namespace of the current element.
<b>override</b>	ActionScript 3: Operation	The <i>override</i> keyword.
<b>prototype</b>	ActionScript 3: Attribute	The <i>prototype</i> keyword.
<b>rest</b>	ActionScript 3: Parameter	The <i>rest</i> parameter ( ... ).

### Common Conventions

- Package qualifiers (ActionScript 2) and Packages (ActionScript 3) are generated when the current Package is not a **namespace root**
- An unspecified type is modeled as *var* or an empty **Type** field

### ActionScript 3 Conventions

- The *Is Leaf* property of a Class corresponds to the sealed keyword
- If a *namespace* tag is specified it overrides the *Scope* that is specified

### Learn more

- [Import Source Code](#) <sup>[2138]</sup>
- [Generate Source Code](#) <sup>[2117]</sup>
- [ActionScript Options](#) <sup>[2264]</sup>
- [Namespaces](#) <sup>[2120]</sup>

## 11.3.2 Ada 2005 Conventions

Enterprise Architect supports round trip engineering of Ada 2005, where the following conventions are used.

### Stereotypes

Stereotype	Applies To	Corresponds To
<b>adaPackage</b>	Class	A package specification in Ada 2005 without a tagged record.
<b>adaProcedure</b>	Class	A procedure specification in Ada 2005.
<b>delegate</b>	Operation	Access to a subprogram.
<b>enumeration</b>	Inner Class	An <i>enum</i> type.
<b>struct</b>	Inner Class	A record definition.
<b>typedef</b>	Inner Class	A type definition, subtype definition, access type definition, renaming.

### Tagged Values

Tag	Applies To	Corresponds To
<b>Discriminant</b>	Inner Class with stereotype <i>typedef</i>	The type's discriminant.
<b>IsAccess</b>	Parameter	Determination of whether the parameter is an access variable.
<b>InstantiatedUnitType</b>	Inner Class with stereotype <i>typedef</i>	The instantiated unit's type ( <i>Package / Procedure / Function</i> ).
<b>PartType</b>	Inner Class with stereotype <i>typedef</i>	The part type ( <i>renames</i> or <i>new</i> ).
<b>Type</b>	Inner Class with stereotype <i>typedef</i>	If Value = <i>Sub Type</i> , set subtype If Value = <i>Access</i> , set access type.

### Other Conventions

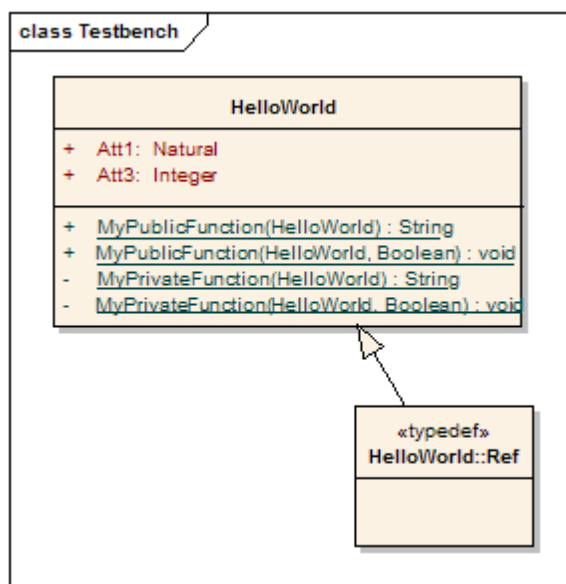
- Appropriate type of source files: Ada specification file, **.ads**
- Ada 2005 imports Packages defined as either <<adaPackage>>Class or Class, based on the settings in the **Ada 2005 options**
- A Package in the Ada specification file is imported as a Class if it contains a Tagged Record, the name of which is governed by the options **Use Class Name for Tagged Record** and **Alternate Tagged Record Name**; all attributes defined in that Tagged Record are absorbed as the Class's attributes
- A procedure / function in an Ada specification file is considered as the Class's member function if its first parameter satisfies the conditions specified in the options **Ref Param Style, Ignore Reference parameter name** and **Ref parameter name**
- The option **Define Reference for Tagged Record**, if enabled, creates a reference type for the Class, the name of which is determined by the option **Reference Type Name**; for example:

```

HelloWorld.ads
package HelloWorld is
    type HelloWorld is tagged record
        Att1: Natural;
        Att3: Integer;
    end record;

    -- Public Functions
    function MyPublicFunction (P: HelloWorld) return String;
    procedure MyPublicFunction (P1: in out HelloWorld; AFlag:
Boolean);
private
    -- Private Functions
    function MyPrivateFunction (P: HelloWorld) return String;
    procedure MyPrivateFunction (P1: in out HelloWorld; AFlag:
Boolean);

end HelloWorld;
```



### Notes

- Ada 2005 support is available in the System Engineering and Ultimate editions of Enterprise Architect

### Learn more

- [Import Source Code](#) <sup>[2138]</sup>
- [Generate Source Code](#) <sup>[2111]</sup>
- [Ada 2005 Options](#) <sup>[2264]</sup>

## 11.3.3 C Conventions

Enterprise Architect supports round trip engineering of C, where the following conventions are used:

### Stereotype

Stereotype	Applies To	Corresponds To
<b>enumeration</b>	Inner Class	An <i>enum</i> type.
<b>struct</b>	Inner Class	A <i>struct</i> type.
	Attribute	A keyword <i>struct</i> in variable definition.
<b>typedef</b>	Inner Class	A <i>typedef</i> statement, where the parent is the original type name.
<b>union</b>	Inner Class	A <i>union</i> type.

Stereotype	Applies To	Corresponds To
	Attribute	A keyword <i>union</i> in variable definition.

### Tagged Values

Tag	Applies To	Corresponds To
<b>anonymous</b>	Class also containing the Tagged Value <i>typedef</i>	The name of this class being defined only by the <i>typedef</i> statement.
<b>bodyLocation</b>	Operation	The location the method body is generated to; expected values are <b>header</b> , <b>classDec</b> or <b>classBody</b> .
<b>typedef</b>	Class with stereotype other than <i>typedef</i>	This Class being defined in a <i>typedef</i> statement.
<b>typeSynonyms</b>	Class	The <i>typedef</i> name and/or fields of this type.

### C Code Generation for UML Model

UML	C Code	Notes
A Class	A pair of C files (.h + .c)	File name is the same as Class name.
Operation (public & protected)	Function declaration in .h file and definition in .c file	
Operation (private)	Function definition in .c file only	
Operation (static)	Function definition in .c file only	Static functions will only appear in the .c file regardless of their scope.
Attribute (public & protected)	Variable definition in .h file	
Attribute (private)	Variable definition in .c file	

UML	C Code	Notes
Inner Class (without stereotype)	(N/A)	This inner Class would be ignored

### Capture #define value to be generated in C code

For example, `#define PI 3.14`.

Step	Process	See also
1	Add an attribute to the Class, with <b>Name = PI</b> and <b>Initial Value = 3.14</b> .	<a href="#">Attributes</a> <sup>[999]</sup>
2	In the General page of the Attribute Properties dialog, select the <b>Static</b> and <b>Const</b> checkboxes.	<a href="#">General Properties of Attributes</a> <sup>[100]</sup>
3	On the Tagged Values page of the Attribute Properties dialog, add a tag called <i>define</i> with the value <b>true</b> .	<a href="#">Attribute Tagged Values</a> <sup>[1005]</sup>

### Notes

- Separate conventions apply to Object Oriented programming in C

### Learn more

- [Import Source Code](#) <sup>[2138]</sup>
- [Generate Source Code](#) <sup>[211]</sup>
- [C Options](#) <sup>[2266]</sup>
- [Object Oriented Programming in C](#) <sup>[2086]</sup>

### 11.3.3.1 Object Oriented Programming In C

The following conventions are used for Object-Oriented programming in C.

To configure Enterprise Architect to support Object-Oriented programming using C, you must set the **Object Oriented Support** option to **True** on the C Specifications page of the Options dialog.

### Reference

#### **Stereotype**

Stereotype	Applies To	Corresponds To
<b>enumeration</b>	Class	An <i>enum</i> type.
<b>struct</b>	Class	A <i>struct</i> type.

Stereotype	Applies To	Corresponds To
	Attribute	A keyword <i>struct</i> in variable definition.
<b>typedef</b>	Class	A <i>typedef</i> statement, where the parent is the original type name.
<b>union</b>	Class	A <i>union</i> type.
	Attribute	A keyword <i>union</i> in variable definition.

### Tagged Values

Tag	Applies To	Corresponds To
<b>anonymous</b>	Class with stereotype of <i>enumeration, struct or union</i>	The name of this Class being defined only by the <i>typedef</i> statement.
<b>bodyLocation</b>	Operation	The location the method body is generated to; expected values are <b>header</b> , <b>classDec</b> or <b>classBody</b> .
<b>define</b>	Attribute	<i>#define</i> statement.
<b>typedef</b>	Class with stereotype of <i>enumeration, struct or union</i>	This Class being defined in a <i>typedef</i> statement.

### Topics

Topic	Detail	See also
<b>Object-Oriented C Code Generation for UML Model</b>	<p>The basic idea of implementing a UML Class in C code is to group the data variable (UML attributes) into a structure type; this structure is defined in a .h file so that it can be shared by other classes and by the client that referred to it.</p> <p>An operation in a UML Class is implemented in C code as a function; the name of the function must be a fully qualified name that consists of the operation name, as well as the Class name to indicate that the operation is for that Class.</p> <p>A delimiter (specified in the <b>Namespace Delimiter</b> option on the C Specifications page) is used to join the Class name and function (operation) name.</p> <p>The function in C code must also have a reference parameter to the Class object - you can modify the <b>Reference as Operation Parameter</b>, <b>Reference Parameter Style</b> and <b>Reference Parameter Name</b> options on the C Specifications page to support this reference parameter.</p>	<a href="#">C Specification</a> s [2266]
<b>Limitations of Object-Oriented Programming in C</b>	<ul style="list-style-type: none"> <li>No scope mapping for an attribute: an attribute in a UML Class is mapped to a structure variable in C code, and its scope (private, protected or public) is ignored</li> <li>Currently an inner Class is ignored: if a UML Class is the inner Class of another UML Class, it is ignored when generating C code</li> </ul>	

Topic	Detail	See also
	<ul style="list-style-type: none"> <li>Initial value is ignored: the initial value of an attribute in a UML Class is ignored in generated C code</li> </ul>	

#### Learn more

- [Import Source Code](#) [2138]
- [Generate Source Code](#) [2117]
- [C Options](#) [2266]

### 11.3.4 C# Conventions

Enterprise Architect supports the round trip engineering of C#, where the following conventions are used.

#### Stereotypes

Stereotype	Applies To	Corresponds To
<b>enumeration</b>	Class	An <i>enum</i> type.
<b>event</b>	Operation	An event.
<b>extension</b>	Operation	A Class extension method, represented in code by a <i>this</i> parameter in the signature.
<b>indexer</b>	Operation	A property acting as an index for this Class.
<b>partial</b>	Operation	The <i>partial</i> keyword on an operation.
<b>property</b>	Operation	A property possibly containing both read and write code.
<b>struct</b>	Class	A <i>struct</i> type.

#### Tagged Values

Tag	Applies To	Corresponds To
<b>argumentName</b>	Operation with stereotype <i>extension</i>	The name given to <i>this</i> parameter.



Tag	Applies To	Corresponds To
<b>attribute_name</b>	Operation with stereotype <i>property</i> or <i>event</i>	The name of the variable behind this property or event.
<b>className</b>	Operation with stereotype <i>extension</i>	The Class that this method is being added to.
<b>const</b>	Attribute	The <i>const</i> keyword.
<b>definition</b>	Operation with stereotype <i>partial</i>	Whether this is the declaration of the method, or the definition.
<b>delegate</b>	Operation	The <i>delegate</i> keyword.
<b>enumType</b>	Operation with stereotype <i>property</i>	The datatype that the property is represented as.
<b>extern</b>	Operation	The <i>extern</i> keyword.
<b>fixed</b>	Attribute	The <i>fixed</i> keyword.
<b>generic</b>	Operation	The generic parameters for this Operation.
<b>genericConstraints</b>	Templated Class or Interface, Operation with tag <i>generic</i>	The constraints on the generic parameters of this type or operation.
<b>Implements</b>	Operation	The name of the method this implements, including the interface name.
<b>ImplementsExplicit</b>	Operation	The presence of the source interface name in this method declaration.
<b>initializer</b>	Operation	A constructor initialization list.
<b>new</b>	Class, Interface, Operation	The <i>new</i> keyword.
<b>override</b>	Operation	The <i>override</i> keyword.

Tag	Applies To	Corresponds To
<b>params</b>	Parameter	A parameter list using the <code>params</code> keyword.
<b>partial</b>	Class, Interface	The <i>partial</i> keyword.
<b>readonly</b>	Operation with stereotype <i>property</i>	This property only defining read code.
<b>sealed</b>	Operation	The <i>sealed</i> keyword.
<b>static</b>	Class	The <i>static</i> keyword.
<b>unsafe</b>	Class, Interface, Operation	The <i>unsafe</i> keyword.
<b>virtual</b>	Operation	The <i>virtual</i> keyword.
<b>writeonly</b>	Operation with stereotype <i>property</i>	This property only defining write code.

#### Other Conventions

- Namespaces are generated for each package below a **namespace root**
- The *Const* property of an attribute corresponds to the *readonly* keyword, while the tag *const* corresponds to the *const* keyword
- The value of *inout* for the *Kind* property of a parameter corresponds to the *ref* keyword
- The value of *out* for the *Kind* property of a parameter corresponds to the *out* keyword
- Partial Classes can be modeled as two separate Classes with the *partial* tag
- The *Is Leaf* property of a Class corresponds to the *sealed* keyword

#### Learn more

- [Import Source Code](#) <sup>[2138]</sup>
- [Generate Source Code](#) <sup>[2111]</sup>
- [C# Options](#) <sup>[2268]</sup>
- [Namespaces](#) <sup>[2120]</sup>

### 11.3.5 C++ Conventions

Enterprise Architect supports round trip engineering of C++, including the **Managed C++** and **C++/CLI** extensions, where the following conventions are used.

#### Stereotypes

Stereotype	Applies To	Corresponds To
<b>enumeration</b>	Class	An <i>enum</i> type.
<b>friend</b>	Operation	The <i>friend</i> keyword.
<b>property get</b>	Operation	A read property.
<b>property set</b>	Operation	A write property.
<b>struct</b>	Class	A <i>struct</i> type.
<b>typedef</b>	Class	A <i>typedef</i> statement, where the parent is the original type name.
<b>union</b>	Class	A <i>union</i> type.

#### Tagged Values

Tag	Applies To	Corresponds To
<b>afx_msg</b>	Operation	The <i>afx_msg</i> keyword.
<b>anonymous</b>	Class also containing the Tagged Value <i>typedef</i>	The name of this Class being only defined by the <i>typedef</i> statement.
<b>attribute_name</b>	Operation with stereotype <i>property get</i> or <i>property set</i>	The name of the variable behind this property.
<b>bitfield</b>	Attribute	The size, in bits, allowed for storage of this attribute.
<b>bodyLocation</b>	Operation	The location the method body is generated to; expected values are <b>header</b> , <b>classDec</b> or <b>classBody</b> .
<b>callback</b>	Operation	A reference to the <b>CALLBACK</b> macro.
<b>explicit</b>	Operation	The <i>explicit</i> keyword.
<b>initializer</b>	Operation	A constructor initialization list.

Tag	Applies To	Corresponds To
<b>inline</b>	Operation	The <i>inline</i> keyword and inline generation of the method body.
<b>mutable</b>	Attribute	The <i>mutable</i> keyword.
<b>throws</b>	Operation	The exceptions that are thrown by this method.
<b>typedef</b>	Class with stereotype other than <i>typedef</i>	This Class being defined in a <i>typedef</i> statement.
<b>typeSynonyms</b>	Class	The <i>typedef</i> name and/or fields of this type.
<b>volatile</b>	Operation	The <i>volatile</i> keyword.

### Other Conventions

- Namespaces are generated for each package below a **namespace root**
- By Reference* attributes correspond to a pointer to the type specified
- The *Transient* property of an attribute corresponds to the *volatile* keyword
- The *Abstract* property of an attribute corresponds to the *virtual* keyword
- The *Const* property of an operation corresponds to the *const* keyword, specifying a constant return type
- The *Is Query* property of an operation corresponds to the *const* keyword, specifying the method doesn't modify any fields
- The *Pure* property of an operation corresponds to a pure *virtual* method using the "**0**" syntax
- The *Fixed* property of a parameter corresponds to the *const* keyword

### Learn more

- [Managed C++ Options](#) <sup>[2093]</sup>
- [C++/CLI Conventions](#) <sup>[2094]</sup>
- [Import Source Code](#) <sup>[2138]</sup>
- [Generate Source Code](#) <sup>[2111]</sup>
- [Namespaces](#) <sup>[2120]</sup>
- [C++ Options](#) <sup>[2269]</sup>

### 11.3.5.1 Managed C++ Conventions

The following conventions are used for managed extensions to C++ prior to C++/CLI. In order to set the system to generate managed C++ you must modify the C++ version in the **C++ Options**.

#### Stereotypes

Stereotype	Applies To	Corresponds To
<b>property</b>	Operation	The <code>__property</code> keyword.
<b>property get</b>	Operation	The <code>__property</code> keyword and a read property.
<b>property set</b>	Operation	The <code>__property</code> keyword and a write property.
<b>reference</b>	Class	The <code>__gc</code> keyword.
<b>value</b>	Class	The <code>__value</code> keyword.

#### Tagged Values

Tag	Applies To	Corresponds To
<b>managedType</b>	Class with stereotype <i>reference</i> , <i>value</i> or <i>enumeration</i> ; Interface	The keyword used in declaration of this type; expected values are <i>class</i> or <i>struct</i> .

#### Other Conventions

- The *typedef* and anonymous tags from native C++ are not supported
- The *Pure* property of an operation corresponds to the keyword `__abstract`

#### Learn more

- [C++/CLI Conventions](#) <sup>[2094]</sup>
- [C++ Options](#) <sup>[2269]</sup>
- [Import Source Code](#) <sup>[2138]</sup>
- [Generate Source Code](#) <sup>[2111]</sup>

### 11.3.5.2 C++/CLI Conventions

The following conventions are used for modeling **C++/CLI** extensions to **C++**. In order to set the system to generate managed C++/CLI you must modify the C++ version in the **C++ Options**.

#### Stereotypes

Stereotype	Applies To	Description
<b>event</b>	Operation	Defines an event to provide access to the event handler for this Class.
<b>property</b>	Operation, Attribute	This is a property possibly containing both read and write code.
<b>reference</b>	Class	Corresponds to the <i>ref class</i> or <i>ref struct</i> keyword.
<b>value</b>	Class	Corresponds to the <i>value class</i> or <i>value struct</i> keyword.

#### Tagged Values

Tag	Applies To	Description
<b>attribute_name</b>	Operation with stereotype <i>property</i> or <i>event</i>	The name of the variable behind this property or event.
<b>generic</b>	Operation	Defines the generic parameters for this Operation.
<b>genericConstraints</b>	Templated Class or Interface, Operation with tag <i>generic</i>	Defines the constraints on the generic parameters for this Operation.
<b>initonly</b>	Attribute	Corresponds to the <i>initonly</i> keyword.
<b>literal</b>	Attribute	Corresponds to the <i>literal</i> keyword.
<b>managedType</b>	Class with stereotype <i>reference</i> , <i>value</i> or <i>enumeration</i> ; Interface	Corresponds to either the <i>class</i> or <i>struct</i> keyword.

**Other Conventions**

- The *typedef* and anonymous tags are not used
- The *property get/property set* stereotypes are not used
- The *Pure* property of an operation corresponds to the keyword *abstract*

**Learn more**

- [C++ Options](#)<sup>[2269]</sup>
- [Import Source Code](#)<sup>[2138]</sup>
- [Generate Source Code](#)<sup>[2111]</sup>

**11.3.6 Delphi Conventions**

Enterprise Architect supports round trip engineering of Delphi, where the following conventions are used.

**Stereotypes**

Stereotype	Applies To	Corresponds To
<b>constructor</b>	Operation	A constructor.
<b>destructor</b>	Operation	A destructor.
<b>dispinterface</b>	Class, Interface	A dispatch interface.
<b>enumeration</b>	Class	An enumerated type.
<b>metaclass</b>	Class	A metaclass type.
<b>object</b>	Class	An object type.
<b>operator</b>	Operation	An operator.
<b>property get</b>	Operation	A read property.
<b>property set</b>	Operation	A write property.
<b>struct</b>	Class	A record type.

Tagged Values

Tag	Applies To	Corresponds To
<b>attribute_name</b>	Operation with stereotype <i>property</i> <i>get</i> or <i>property set</i>	The name of the variable behind this property.
<b>overload</b>	Operation	The <i>overload</i> keyword.
<b>override</b>	Operation	The <i>override</i> keyword.
<b>packed</b>	Class	The <i>packed</i> keyword.
<b>property</b>	Class	A property; see <i>Delphi Properties</i> for more information.
<b>reintroduce</b>	Operation	The <i>reintroduce</i> keyword.

Other Conventions

- The *Static* property of an attribute or operation corresponds to the *class* keyword
- The *Fixed* property of a parameter corresponds to the *const* keyword
- The value of *inout* for the *Kind* property of a parameter corresponds to the *Var* keyword
- The value of *out* for the *Kind* property of a parameter corresponds to the *Out* keyword

Learn more

- [Import Source Code](#) <sup>[2138]</sup>
- [Generate Source Code](#) <sup>[2117]</sup>
- [Delphi Options](#) <sup>[2271]</sup>
- [Delphi Properties](#) <sup>[2272]</sup>

### 11.3.7 Java Conventions

Enterprise Architect supports round trip engineering of Java - including **AspectJ** extensions - where the following conventions are used.

Stereotypes



Stereotype	Applies To	Corresponds To
<b>annotation</b>	Interface	An <i>annotation</i> type.
<b>enum</b>	Attributes within a Class stereotyped <i>enumeration</i>	An <i>enumerated</i> option, distinguished from other attributes that have no stereotype.
<b>enumeration</b>	Class	An <i>enum</i> type.
<b>operator</b>	Operation	An operator.
<b>property get</b>	Operation	A read property.
<b>property set</b>	Operation	A write property.
<b>static</b>	Class or Interface	The <i>static</i> keyword.

#### Tagged Values

Tag	Applies To	Corresponds To
<b>annotations</b>	Anything	The annotations on the current code feature.
<b>arguments</b>	Attribute with stereotype <i>enum</i>	The arguments that apply to this enumerated value.
<b>attribute_name</b>	Operation with stereotype <i>property get</i> or <i>property set</i>	The name of the variable behind this property.
<b>dynamic</b>	Class or Interface	The <i>dynamic</i> keyword.
<b>generic</b>	Operation	The generic parameters to this operation.
<b>parameterList</b>	Parameter	A parameter list with the ... syntax.
<b>throws</b>	Operation	The exceptions that are thrown by this method.

Tag	Applies To	Corresponds To
<b>transient</b>	Attribute	The transient keyword.

#### Other Conventions

- Package statements are generated when the current package is not a **namespace root**
- The *Const* property of an attribute or operation corresponds to the final keyword
- The *Transient* property of an attribute corresponds to the volatile keyword
- The *Fixed* property of a parameter corresponds to the final keyword

#### Learn more

- [AspectJ Conventions](#) <sup>[2098]</sup>
- [Import Source Code](#) <sup>[2138]</sup>
- [Generate Source Code](#) <sup>[2117]</sup>
- [Java Options](#) <sup>[2272]</sup>
- [Namespaces](#) <sup>[2120]</sup>

### 11.3.7.1 *AspectJ Conventions*

The following are the conventions used for supporting AspectJ extensions to Java.

#### Stereotypes

Stereotype	Applies To	Corresponds To
<b>advice</b>	Operation	A piece of advice in an AspectJ aspect.
<b>aspect</b>	Class	An AspectJ aspect.
<b>pointcut</b>	Operation	A pointcut in an AspectJ aspect.

#### Tagged Values

Tag	Applies To	Corresponds To
<b>className</b>	Attribute or operation within a Class stereotyped <i>aspect</i>	The Classes this AspectJ intertype member belongs to.

#### Other Conventions

- The specifications of a pointcut are included in the **Behavior** field of the method

Learn more

- [Java Conventions](#) <sup>[2096]</sup>
- [Import Source Code](#) <sup>[2138]</sup>
- [Generate Source Code](#) <sup>[2117]</sup>

### 11.3.8 PHP Conventions

Enterprise Architect supports the round trip engineering of PHP 4 and 5, where the following conventions are used.

Stereotypes

Stereotype	Applies To	Corresponds To
<b>property get</b>	Operation	A read property.
<b>property set</b>	Operation	A write property.

Tagged Values

Tag	Applies To	Corresponds To
<b>attribute_name</b>	Operation with stereotype <i>property get</i> or <i>property set</i>	The name of the variable behind this property.
<b>final</b>	Operations in PHP 5	The final keyword.

Common Conventions

- An unspecified type is modeled as *var*
- Methods returning a reference are generated by setting the *Return Type* to *var\**
- Reference parameters are generated from parameters with the parameter *Kind* set to *inout* or *out*

PHP 5 Conventions

- The *final* Class modifier corresponds to the *Is Leaf* property
- The *abstract* Class modifier corresponds to the *Abstract* property
- Parameter type hinting is supported by setting the *Type* of a parameter
- The value of *inout* or *out* for the *Kind* property of a parameter corresponds to a *reference* parameter

Learn more

- [Import Source Code](#) <sup>[2138]</sup>
- [Generate Source Code](#) <sup>[2111]</sup>
- [PHP Options](#) <sup>[2274]</sup>

### 11.3.9 Python Conventions

Enterprise Architect supports the round trip engineering of Python, where the following conventions are used.

Tagged Values

Tag	Applies To	Corresponds To
<b>Decorators</b>	Class, Operation	The decorators applied to this element in the source.

Other Conventions

- Model members with *Private Scope* correspond to code members with two leading underscores
- Attributes are only generated when the Initial value is not empty
- All types are reverse engineered as *var*

Learn more

- [Import Source Code](#) <sup>[2138]</sup>
- [Generate Source Code](#) <sup>[2111]</sup>
- [Python Options](#) <sup>[2273]</sup>

### 11.3.10 SystemC Conventions

Enterprise Architect supports round-trip engineering of **SystemC**, where the following conventions are used.

Stereotypes

Stereotype	Applies To	Corresponds To
<b>delegate</b>	Method	A delegate.
<b>enumeration</b>	Inner Class	An <i>enum</i> type.
<b>friend</b>	Method	A <i>friend</i> method.
<b>property</b>	Method	A property definition.

Stereotype	Applies To	Corresponds To
<b>sc_ctor</b>	Method	A SystemC constructor.
<b>sc_module</b>	Class	A SystemC module.
<b>sc_port</b>	Attribute	A port.
<b>sc_signal</b>	Attribute	A signal.
<b>struct</b>	Inner Class	A <i>struct</i> or <i>union</i> .

#### Tagged Values

Tag	Applies To	Corresponds To
<b>kind</b>	Attribute (Port)	Port kind ( <i>clocked</i> , <i>fifo</i> , <i>master</i> , <i>slave</i> , <i>resolved</i> , <i>vector</i> ).
<b>mode</b>	Attribute (Port)	Port mode ( <i>in</i> , <i>out</i> , <i>inout</i> ).
<b>overrides</b>	Method	The <i>Inheritance</i> list of a method declaration.
<b>throw</b>	Method	The exception specification of a method.

#### Other Conventions

- SystemC also inherits most of the stereotypes and Tagged Values of C++

#### SystemC Toolbox Pages





Access **Diagram | Diagram Toolbox: More tools | HDL | SystemC Constructs**

Drag these icons onto a diagram to model a SystemC design.

Page	Icon	Action
<b>SystemC</b>	<b>Module</b>	Defines a SystemC Module.

Page	Icon	Action
		An <i>sc_module</i> -stereotyped Class element.
	<b>Enumeration</b>	Defines an Enumerated Type. An Enumeration element.
	<b>Struct</b>	Defines a Structure. A <i>struct</i> -stereotyped Class element.
<b>SystemC Features</b>	<b>Port</b>	Defines a SystemC Port. An <i>sc_port</i> -stereotyped attribute.
	<b>Signal</b>	Defines a SystemC Signal. An <i>sc_signal</i> -stereotyped attribute.
	<b>Constructor</b>	Defines a SystemC Constructor. An <i>sc_ctor</i> -stereotyped method.

#### Learn more

- [Import Source Code](#)  <sup>[2138]</sup>
- [Generate Source Code](#)  <sup>[2117]</sup>
- [C++ Conventions](#)  <sup>[2090]</sup>
- [SystemC Options](#)  <sup>[2275]</sup>

### 11.3.11 VB.NET Conventions

Enterprise Architect supports round-trip engineering of Visual Basic.NET, where the following conventions are used. Earlier versions of Visual Basic are supported as a different language.

#### Stereotypes

Stereotype	Applies To	Corresponds To
<b>event</b>	Operation	An event declaration.
<b>import</b>	Operation	An operation to be imported from another library.
<b>module</b>	Class	A module.

Stereotype	Applies To	Corresponds To
<b>operator</b>	Operation	An operator overload definition.
<b>partial</b>	Operation	The <i>partial</i> keyword on an operation.
<b>property</b>	Operation	A property possibly containing both read and write code.

#### Tagged Values

Tag	Applies To	Corresponds To
<b>Alias</b>	Operation with stereotype <i>import</i>	The alias for this imported operation.
<b>attribute_name</b>	Operation with stereotype <i>property</i>	The name of the variable behind this property.
<b>Charset</b>	Operation with stereotype <i>import</i>	The <i>character set</i> clause for this import - one of the values <i>Ansi</i> , <i>Unicode</i> or <i>Auto</i> .
<b>delegate</b>	Operation	The <i>Delegate</i> keyword.
<b>enumTag</b>	Operation with stereotype <i>property</i>	The datatype that this property is represented as.
<b>Handles</b>	Operation	The <i>handles</i> clause on this operation.
<b>Implements</b>	Operation	The <i>implements</i> clause on this operation.
<b>Lib</b>	Operation with stereotype <i>import</i>	The library this import comes from.
<b>MustOverride</b>	Operation	The <i>MustOverride</i> keyword.
<b>Narrowing</b>	Operation with stereotype <i>operator</i>	The <i>Narrowing</i> keyword.
<b>NotOverrideable</b>	Operation	The <i>NotOverrideable</i> keyword.
<b>Overloads</b>	Operation	The <i>Overloads</i> keyword.

Tag	Applies To	Corresponds To
<b>Overrides</b>	Operation	The <i>Overrides</i> keyword.
<b>parameterArray</b>	Parameter	A parameter list using the <i>ParamArray</i> keyword.
<b>partial</b>	Class, Interface	The <i>Partial</i> keyword.
<b>readonly</b>	Operation with stereotype <i>property</i>	This property only defining read code.
<b>shadows</b>	Class, Interface, Operation	The <i>Shadows</i> keyword.
<b>Shared</b>	Attribute	The <i>Shared</i> keyword.
<b>Widening</b>	Operation with stereotype <i>operator</i>	The <i>Widening</i> keyword.
<b>writeonly</b>	Operation with stereotype <i>property</i>	This property only defining write code.

### Other Conventions

- Namespaces are generated for each package below a **namespace root**
- The *Is Leaf* property of a Class corresponds to the *NotInheritable* keyword
- The *Abstract* property of a Class corresponds to the *MustInherit* keyword
- The *Static* property of an attribute or operation corresponds to the *Shared* keyword
- The *Abstract* property of an operation corresponds to the *MustOverride* keyword
- The value of *in* for the *Kind* property of a parameter corresponds to the *ByVal* keyword
- The value of *inout* or *out* for the *Kind* property of a parameter corresponds to the *ByRef* keyword

### Learn more

- [Visual Basic Conventions](#) <sup>[2109]</sup>
- [Import Source Code](#) <sup>[2138]</sup>
- [Generate Source Code](#) <sup>[2117]</sup>
- [VB.NET Options](#) <sup>[2275]</sup>
- [Namespaces](#) <sup>[2120]</sup>



### 11.3.12 Verilog Conventions

Enterprise Architect supports round-trip engineering of Verilog, where the following conventions are used.

#### Stereotypes

Stereotype	Applies To	Corresponds To
<b>asynchronous</b>	Method	A concurrent process.
<b>enumeration</b>	Inner Class	An <i>enum</i> type.
<b>initializer</b>	Method	An initializer process.
<b>module</b>	Class	A module.
<b>part</b>	Attribute	A component instantiation.
<b>port</b>	Attribute	A port.
<b>synchronous</b>	Method	A sequential process.

#### Tagged Values

Tag	Applies To	Corresponds To
<b>kind</b>	Attribute (signal)	The signal kind (such as <i>register</i> , <i>bus</i> ).
<b>mode</b>	Attribute (port)	The port mode ( <i>in</i> , <i>out</i> , <i>inout</i> ).
<b>Portmap</b>	Attribute (part)	The generic / port map of the component instantiated.
<b>sensitivity</b>	Method	The sensitivity list of a sequential process.
<b>type</b>	Attribute	The range or type value of an attribute.




### Verilog Toolbox Pages

**Access** [Diagram](#) | [Diagram Toolbox: More tools](#) | [HDL](#) | [Verilog Constructs](#)

Drag these icons onto a diagram to model a Verilog design.

Page	Item	Action
<b>Verilog</b>	Module	Defines a Verilog Module. A <i>module</i> -stereotyped Class element.
	Enumeration	Defines an Enumerated Type. An enumeration element.
<b>Verilog Features</b>	Port	Defines a Verilog Port. A <i>port</i> -stereotyped attribute.
	Part	Defines a Verilog component instantiation. A <i>part</i> -stereotyped attribute.
	Attribute	Defines an attribute.
	Procedure <ul style="list-style-type: none"> <li>• Concurrent</li> <li>• Sequential</li> <li>• Initializer</li> </ul>	Defines a Verilog process: <ul style="list-style-type: none"> <li>• An <i>asynchronous</i>-stereotyped method</li> <li>• A <i>synchronous</i>-stereotyped method</li> <li>• An <i>initializer</i>-stereotyped method</li> </ul>

### Learn more

- [Import Source Code](#)  <sup>[2138]</sup>
- [Generate Source Code](#)  <sup>[2117]</sup>
- [Verilog Options](#)  <sup>[2276]</sup>

## 11.3.13 VHDL Conventions

Enterprise Architect supports round-trip engineering of VHDL, where the following conventions are used.

### Stereotypes

Stereotype	Applies To	Corresponds To
<b>architecture</b>	Class	An architecture.
<b>asynchronous</b>	Method	An asynchronous process.

Stereotype	Applies To	Corresponds To
<b>configuration</b>	Method	A configuration.
<b>enumeration</b>	Inner Class	An <i>enum</i> type.
<b>entity</b>	Interface	An entity.
<b>part</b>	Attribute	A component instantiation.
<b>port</b>	Attribute	A port.
<b>signal</b>	Attribute	A signal declaration.
<b>struct</b>	Inner Class	A record definition.
<b>synchronous</b>	Method	A synchronous process.
<b>typedef</b>	Inner Class	A <i>type</i> or <i>subtype</i> definition.

#### Tagged Values

Tag	Applies To	Corresponds To
<b>isGeneric</b>	Attribute (port)	The port declaration in a generic interface.
<b>isSubType</b>	Inner Class (typedef)	A subtype definition.
<b>kind</b>	Attribute (signal)	The signal kind (such as <i>register</i> , <i>bus</i> ).
<b>mode</b>	Attribute (port)	The port mode ( <i>in</i> , <i>out</i> , <i>inout</i> , <i>buffer</i> , <i>linkage</i> ).
<b>portmap</b>	Attribute (part)	The generic / port map of the component instantiated.
<b>sensitivity</b>	Method (synchronous)	The sensitivity list of a synchronous process.

Tag	Applies To	Corresponds To
<b>type</b>	Inner Class (typedef)	The type indication of a type declaration.
<b>typeNameSpace</b>	Attribute (part)	The type namespace of the instantiated component.

### VHDL Toolbox Pages

Access [Diagram](#) | [Diagram Toolbox: More tools](#) | [HDL](#) | [VHDL Constructs](#)

Drag these icons onto a diagram to model a VHDL design.

Page	Item	Action
<b>VHDL</b>	Architecture	Defines an architecture to be associated with a VHDL entity. <i>An <b>architecture</b>-stereotyped Class element.</i>
	Entity	Defines a VHDL entity to contain the Port definitions. <i>An <b>entity</b>-stereotyped interface element.</i>
	Enumeration	Defines an Enumerated Type. <i>An Enumeration element.</i>
	Struct	Defines a VHDL record. <i>A <b>struct</b>-stereotyped Class element.</i>
	Typedef	Defines a VHDL type or subtype. <i>A <b>typedef</b>-stereotyped Class element.</i>
<b>VHDL Features</b>	Port	Defines a VHDL Port. <i>A <b>port</b>-stereotyped attribute.</i>
	Part	Defines a VHDL component instantiation. <i>A <b>part</b>-stereotyped attribute.</i>
	Signal	Defines a VHDL signal. <i>A <b>signal</b>-stereotyped attribute.</i>

Page	Item	Action
	Procedure <ul style="list-style-type: none"> <li>• Concurrent</li> <li>• Sequential</li> <li>• Configuration.</li> </ul>	Defines a VHDL process: <ul style="list-style-type: none"> <li>• An <i>asynchronous</i>-stereotyped method</li> <li>• A <i>synchronous</i>-stereotyped method</li> <li>• A <i>configuration</i>-stereotyped method</li> </ul>

#### Learn more

- [Import Source Code](#)<sup>[2138]</sup>
- [Generate Source Code](#)<sup>[2117]</sup>
- [VHDL Options](#)<sup>[2277]</sup>

### 11.3.14 Visual Basic Conventions

Enterprise Architect supports the round trip engineering of Visual Basic 5 and 6, where the following conventions are used.

Visual Basic .Net is supported as a different language.

#### Stereotypes

Stereotype	Applies To	Corresponds To
<b>global</b>	Attribute	The <i>Global</i> keyword.
<b>import</b>	Operation	An operation to be imported from another library.
<b>property get</b>	Operation	A property <i>get</i> .
<b>property set</b>	Operation	A property <i>set</i> .
<b>property let</b>	Operation	A property <i>let</i> .
<b>with events</b>	Attribute	The <i>WithEvents</i> keyword.





#### Tagged Values

Tag	Applies To	Corresponds To
<b>Alias</b>	Operation with stereotype <i>import</i>	The alias for this imported operation.
<b>attribute_name</b>	Operation with stereotype <i>property get</i> , <i>property set</i> or <i>property let</i>	The name of the variable behind this property.
<b>Lib</b>	Operation with stereotype <i>import</i>	The library this import comes from.
<b>New</b>	Attribute	The <i>New</i> keyword.

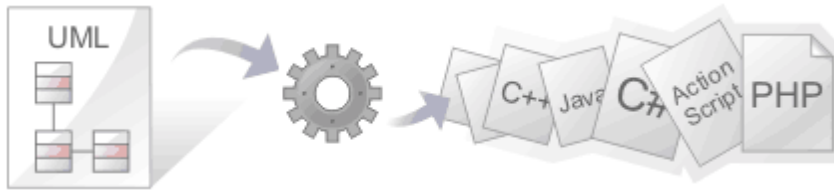
### Other Conventions

- The value of *in* for the *Kind* property of a parameter corresponds to the *ByVal* keyword
- The value of *inout* or *out* for the *Kind* property of a parameter corresponds to the *ByRef* keyword

### Learn more

- [Visual Basic .Net Conventions](#)  <sup>[2102]</sup>
- [Import Source Code](#)  <sup>[2138]</sup>
- [Generate Source Code](#)  <sup>[2111]</sup>
- [Visual Basic Options](#)  <sup>[2277]</sup>

## 11.4 Generate Source Code



Generating source code (forward engineering) takes the UML Class or Interface model elements and creates a source code equivalent for future elaboration and compilation. By forward engineering code from the model, the mundane work involved with having to key in Classes and attributes and methods is avoided, and symmetry between model and code is ensured.

### Topics

Topic	Detail	See also
<b>Languages</b>	Enterprise Architect supports code generation in each of the following languages: <ul style="list-style-type: none"> <li>• Action Script</li> <li>• C</li> <li>• C# (for .NET 1.1, .NET 2.0 and .NET 4.0)</li> <li>• C++ (standard, plus .NET managed C++ extensions)</li> <li>• Delphi</li> <li>• Java (including Java 1.5, Aspects and Generics)</li> <li>• PHP</li> <li>• Python</li> <li>• Visual Basic</li> <li>• Visual Basic .NET</li> </ul>	
<b>Elements</b>	Code is generated from Class or Interface model elements, so you must create the required Class and Interface elements to generate from.  Add attributes (which become variables) and operations (which become methods).	
<b>Settings</b>	Before you generate code, you should ensure the default settings for code generation match your requirements; set up the defaults to match your required language and preferences.  Preferences that you can define include default constructors and destructors, methods for interfaces and the Unicode options for created languages.  Languages such as Java support 'namespaces' and can be configured to specify a namespace root.	<a href="#">Source Code Options</a> <sup>[2247]</sup> <a href="#">Namespaces</a> <sup>[2120]</sup> <a href="#">ActionScript Options</a> <sup>[2081]</sup> <a href="#">C Options</a> <sup>[2266]</sup> <a href="#">C# Options</a> <sup>[2268]</sup> <a href="#">C++ Options</a> <sup>[2269]</sup> <a href="#">Delphi Options</a> <sup>[2271]</sup>



	In addition to the default settings for generating code, Enterprise Architect facilitates setting specific generation options for each of the supported languages.	<a href="#">Java Options</a> <sup>[2272]</sup> <a href="#">PHP Options</a> <sup>[2274]</sup> <a href="#">Python Options</a> <sup>[2273]</sup> <a href="#">Visual Basic Options</a> <sup>[2277]</sup> <a href="#">Visual Basic .NET Options</a> <sup>[2275]</sup>
<b>Code Template Framework</b>	The Code Template Framework (CTF) enables you to customize the way Enterprise Architect generates source code and also enables generation of languages that are not specifically supported by Enterprise Architect.	<a href="#">Code Template Framework Tool</a> <sup>[2281]</sup>
<b>Local Paths</b>	Local path names enable you to substitute tags for directory names	<a href="#">Local Paths</a> <sup>[2255]</sup> <a href="#">Local Paths Dialog</a> <sup>[2256]</sup>
<b>Behavioral Code</b>	You can also generate code from three UML behavioral modeling paradigms: <ul style="list-style-type: none"> <li>• State Machine diagrams (SW &amp; HW)</li> <li>• Interaction (Sequence) diagrams (SW)</li> <li>• Activity diagrams (SW)</li> </ul>	<a href="#">Legacy State Machine Templates</a> <sup>[2126]</sup> <a href="#">Interaction (Sequence) diagrams</a> <sup>[2133]</sup> <a href="#">Activity diagrams</a> <sup>[2134]</sup>
<b>Live Code Generation</b>	On the <b>Code Engineering</b> submenu, you have the option to update your source code instantly as you make changes to your model.	<a href="#">Code Engineering Options</a> <sup>[657]</sup>
<b>Tasks</b>	When you generate code, you perform one or more of the following tasks: <ul style="list-style-type: none"> <li>• Generate a Single Class</li> <li>• Generate a Group of Classes</li> <li>• Generate a Package</li> <li>• Update Package Contents</li> </ul>	<a href="#">Generate a Single Class</a> <sup>[2113]</sup> <a href="#">Generate a Group of Classes</a> <sup>[2114]</sup> <a href="#">Generate a Package</a> <sup>[2115]</sup> <a href="#">Update Package Contents</a> <sup>[2117]</sup>

**Notes**

- The tools provided by Enterprise Architect for code engineering and debugging are available in the Professional, Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect
- When security is enabled there are permissions for Generate Source Code and DDL and Reverse Engineer from DDL and Source Code

**Learn more**



- [Importing Source Code](#) 
- [Generate From Behavioral Models](#) 

#### Learning Center topics



- (Alt+F1) | **Enterprise Architect | Software Engineering | Generate Code**

### 11.4.1 Generate a Single Class

Before you generate code for a single Class, you:

- Complete the design of the model element (Class or Interface)
- Create Inheritance connectors to parents and Associations to other Classes that are used
- Create Inheritance connectors to Interfaces that your Class implements; the system provides an option to generate function stubs for all interface methods that a Class implements

#### Generate code for a single Class

Step	Action	See also
1	Open the diagram containing the Class or Interface for which to generate code.	
2	Right-click on the required Class or Interface and select the <b>Generate Code</b> context menu option, or press <b>F11</b> .  The Generate Code dialog displays, through which you can control how and where your source code is generated.	
3	In the <b>Path</b> field, click on the ( ... ) (Browse) button and select a path name for your source code to be generated to.	
4	In the <b>Target Language</b> field, click on the drop-down arrow and select the language to generate; this becomes the permanent option for that Class, so change it back if you are only doing one pass in another language.	
5	Click on the <b>Advanced</b> button.  The Object Options dialog displays, providing subsets of the Source Code Engineering and code language options pages on the Options dialog.	<a href="#">Source Code Options</a>   <a href="#">Language Options</a> 
6	Set any custom options (for this Class alone), then click on the <b>Close</b> button to return to the Generate Code dialog.	
7	If you are generating Java, C++, C# or C code, the <b>Use the New State Machine Template</b> option displays with the checkbox selected.	<a href="#">Legacy State</a>

Step	Action	See also
	If you want to use the Legacy Statemachine code generation templates, deselect the checkbox.	<a href="#">Machine Templates</a> <sup>[2126]</sup>
<b>8</b>	In the <b>Import(s) / Header(s)</b> fields, type any import statements, <i>#includes</i> or other header information.  Note that in the case of Visual Basic this information is ignored; in the case of Java the two import text boxes are merged; and in the case of C++ the first import text area is placed in the header file and the second in the body (.cpp) file.	
<b>9</b>	Click on the <b>Generate</b> button to create the source code.	
<b>10</b>	When complete, click on the <b>View</b> button to see what has been generated.  Note that you should set up your default viewer/editor for each language type first; you can also set up the default editor on the Code Editors page of the Options dialog ( <b>Tools   Options   Source Code Engineering   Code Editors</b> ).	<a href="#">Editing Source Code</a> <sup>[2146]</sup>

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Software Engineering | Generate Code | Generate a Single Class**

### 11.4.2 Generate a Group of Classes

In addition to being able to generate code for an individual Class, you can also select a group of Classes for batch code generation. When you do this, you accept all the default code generation options for each Class in the set.

#### Generate Class Group

Step	Detail	See also
<b>1</b>	Select a group of Classes and/or interfaces in a diagram.	
<b>2</b>	Right-click on an element in the group to display the context menu.	
<b>3</b>	Select the <b>Code Generation   Generate Selected elements</b> menu option.  If no code exists for the selected elements, the Save As dialog displays on which you specify the file path and name for each code file; enter this information and click on the <b>Save</b> button.	
<b>4</b>	The Batch Generation dialog displays, showing the status of the process as it executes (the process might be too fast to see this dialog).  If code already exists for the selected Class elements, and changes have been made to the Class name or structure, the Synchronize Element <package name>	<a href="#">Synchronize</a>

	.<element name> dialog might also display; this dialog helps synchronize the model and code.	<a href="#">Model and Code</a> [2118]
--	--	--

**Notes**

- If any of the elements selected are not Classes or interfaces the option to generate code is not available

**Learning Center**

- (Alt+F1) | **Enterprise Architect | Software Engineering | Generate Code | Generate Multiple Classes**

**11.4.3 Generate a Package**

In addition to generating source code from single Classes and groups of Classes, you can also generate code from a package. This feature provides options to recursively generate code from child packages and automatically generate directory structures based on the package hierarchy. This enables you to generate code for a whole branch of your project model in one step.

**Access** **Project Browser package context menu | Code Engineering | Generate Source Code**

**Generate code from a Package, on the Generate Package Source Code dialog**

Step	Action	See also
1	<p>In the <b>Synchronize</b> field, click on the drop-down arrow and select the appropriate synchronize option:</p> <ul style="list-style-type: none"> <li>• <b>Synchronize model and code:</b> Code for Classes with existing files is forward synchronized with that file; code for Classes with no existing file is generated to the displayed target file</li> <li>• <b>Overwrite code:</b> All selected target files are overwritten (forward generated)</li> <li>• <b>Do not generate:</b> Generate code for only those selected Classes that do not have an existing file; all other Classes are ignored</li> </ul>	
2	<p>Highlight the Classes for which to generate code; leave unselected any to not generate code for.</p> <p>If you want to display more of the information within the layout, you can resize the dialog and its columns.</p>	
3	<p>To make Enterprise Architect automatically generate directories and filenames based on the package hierarchy, select the <b>Auto Generate Files</b> checkbox; this enables the <b>Root Directory</b> field, in which you select a root directory under which the source directories are to be generated.</p> <p>By default, the <b>Auto Generate Files</b> feature <i>ignores</i> any file paths that are already associated with a Class; you can change this behavior by also selecting the <b>Retain Existing File Paths</b> checkbox.</p>	

Step	Action	See also
4	To include code for all sub-packages in the output, select the <b>Include Child Packages</b> checkbox.	
5	Click on the <b>Generate</b> button to start generating code.	

As code generation proceeds Enterprise Architect displays progress messages. If a Class requires an output filename Enterprise Architect prompts you to enter one at the appropriate time (assuming **Auto Generate Files** is not selected). For example, if the selected Classes include partial Classes, a prompt displays to enter the filename into which to generate code for the second partial Class.

For additional information on the options on the Generate Package Source Code dialog, see the following table:

Option	Action	See also
<b>Root Package</b>	Check the name of the package for which code is to be generated.	
<b>Synchronize</b>	Select options that specify how existing files should be regenerated.	
<b>Auto Generate Files</b>	Specify whether Enterprise Architect should automatically generate file names and directories, based on the package hierarchy.	
<b>Root Directory</b>	If <b>Auto Generate Files</b> is selected, display the path under which the generated directory structures are created.	
<b>Retain Existing File Paths</b>	If <b>Auto Generate Files</b> is selected, specify whether to use existing file paths associated with Classes.  If <b>Auto Generate Files</b> is unselected, Enterprise Architect generates Class code to automatically determined paths, regardless of whether source files are already associated with the Classes.	
<b>Include all Child Packages</b>	Also generate code for all Classes in all sub-packages of the target package in the list.  This option facilitates recursive generation of code for a given package and its sub-packages.	
<b>Select Objects to Generate</b>	List all Classes that are available for code generation under the target packages; only code for selected (highlighted) Classes is generated.	

Option	Action	See also
	Classes are listed with their target source file.	
<b>Select All</b>	Mark all Classes in the list as selected.	
<b>Select None</b>	Mark all Classes in the list as unselected.	
<b>Generate</b>	Start the generation of code for all selected Classes.	
<b>Cancel</b>	Exit the Generate Package Source Code dialog; no Class code is generated.	

#### Learning Center topics

- (Alt+F1) | [Enterprise Architect | Software Engineering | Generate Code | Generate a Package](#)

### 11.4.4 Update Package Contents

In addition to generating and importing code, Enterprise Architect provides the option to synchronize the model and source code, creating a model that represents the latest changes in the source code and vice versa. You can use either the model as the source, or the code as the source.

The behavior and actions of synchronization depend on the settings you have selected on the Attributes and Operations page of the Options dialog. These settings enable you to either protect or automatically discard information in the model that is not present in the code, and to prompt for a decision on code features that are not in the model. In the following two examples, the appropriate checkboxes have been selected for maximum protection of data:

- You generated some source code, but made subsequent changes to the model. When you generate code again, Enterprise Architect *adds* any new attributes or methods to the existing source code, leaving intact what already exists. This means developers can work on the source code and then generate *additional* methods as required from the model, without having their code overwritten or destroyed.
- You might have made changes to a source code file, but the model has detailed notes and characteristics you do not want to lose. By synchronizing from the source code into the model, you import additional attributes and methods but do not change other model elements.

Using the synchronization methods, it is simple to keep source code and model elements up to date and synchronized.

**Access** [Project Browser Package context menu | Code Engineering | Synchronize Package With Code \(Ctrl+Alt+M\)](#) or [Tools | Source Code Engineering | Synchronize Package Contents](#)

#### Synchronize package contents against source code

Field/Button	Action	See also
<b>Update Type</b>	Select the radio button to either <b>Forward Engineer</b> or <b>Reverse Engineer</b> the package Classes, as appropriate.	
<b>Include child packages in generation</b>	Select the checkbox to include child packages in the synchronization.	
<b>OK</b>	<p>Click on the button to start synchronization.</p> <p>Enterprise Architect uses the directory names specified when the project source was first imported/generated and updates either the model or the source code depending on the option chosen. IF:</p> <ul style="list-style-type: none"> <li>Performing forward synchronization AND</li> <li>There are differences between the model and code AND</li> <li>The <b>On forward synch, prompt to delete code features not in model</b> checkbox is selected in the Options - Attributes and Operations dialog</li> </ul> <p>THEN the Synchronize Element &lt;package name&gt;.&lt;element name&gt; dialog displays.</p> <p>Otherwise, no further action is required.</p>	<a href="#">Synchronize Model and Code</a> <small>[2118]</small>

### Notes

- Code synchronization does not change method bodies; behavioral code cannot be synchronized, and code generation only works when generating the entire file
- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Generate Source Code and DDL** permission to synchronize source code with model elements

### Learn more

- [Options - Attributes and Operations](#) [2253]
- [User Security](#) [316]
- [Generate From Behavioral Models](#) [2121]

### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Software Engineering | Generate Code | Update Packages**

#### 11.4.4.1 Synchronize Model and Code

You might either:

- Synchronize the code for a package of Classes against the model in the Project Browser, or
- Regenerate code from a batch of Classes in the model

In such processes, there might be items in the code that are not present in the model.

If you want to trap those items and resolve them manually, select the **On forward synch, prompt to delete code features not in model** checkbox in the Options - Attributes and Operations dialog, so that the Synchronize Element <package name> .<element name> dialog displays, providing options to respond to each item.

### Reference

On the Synchronize Element <package name> .<element name> dialog, select one or more items and process them as indicated below:


Button	Detail	See also
<b>Select All</b>	Highlight and select all items in the <b>Feature</b> column.	
<b>Clear All</b>	Deselect and remove highlighting from all items in the <b>Feature</b> column.	
<b>Delete</b>	Mark the selected code features to be removed from the code (the value in the <b>Action</b> column changes to <b>Delete</b> ).	
<b>Reassign</b>	Mark the selected code features to be reassigned to elements in the model.  This is only possible when an appropriate model element is present that is not already defined in the code.  The Select the Corresponding Class Feature dialog displays, from which you select the Class to reassign the feature to. Click on the <b>OK</b> button to mark the feature for reassignment.	
<b>Ignore</b>	Mark the selected code elements not present in the model to be ignored completely (the default; the value in the <b>Action</b> column remains as or changes to <b>&lt;none&gt;</b> ).	
<b>Reset to Default</b>	Reset the selected items to <b>Ignore</b> (the value in the <b>Action</b> column changes to <b>&lt;none&gt;</b> ).	
<b>OK</b>	Make the assigned changes to the items, and close the dialog.	

### Learn more

- [Update Package Contents](#) <sup>[2117]</sup>
- [Generate a Group of Classes](#) <sup>[2114]</sup>
- [Options - Attributes and Operations](#) <sup>[2253]</sup>

### 11.4.5 Namespaces

Languages such as Java support Package structures or **namespaces**. In Enterprise Architect you can specify a Package as a **namespace root**, which denotes where the namespace structure for your Class model starts; all subordinate Packages below a namespace root will form the namespace hierarchy for contained Classes and Interfaces.

To define a Package as a namespace root, right-click on the Package in the Project Browser and select the **Code Engineering | Set as Namespace Root** context menu option; the Package icon in the Project Browser changes to show a colored corner indicating this Package as a namespace root ().

Generated Java source code, for example, will automatically add a Package declaration at the beginning of the generated file, indicating the location of this Class in the Package hierarchy below the namespace root.

To clear an existing namespace root, right-click on a namespace root Package in the Project Browser and select the **Code Engineering | Clear Namespace Root**.

To view a list of namespaces, select the **Settings | Namespace Roots** menu option; the **Namespaces** dialog displays. If you double-click on a namespace in the list, the Package is highlighted in the Project Browser; alternatively, right-click on the namespace to display a context menu, and select the **Locate Package in Browser** menu option. You can also clear the selected namespace root by selecting the **Clear Namespace Attribute** option.

To omit a Package from a namespace definition, select the **Code Engineering | Suppress Namespace** context menu option; to include the Package in the namespace again, select the **Code Engineering | Show Namespace** context menu option.

#### Notes

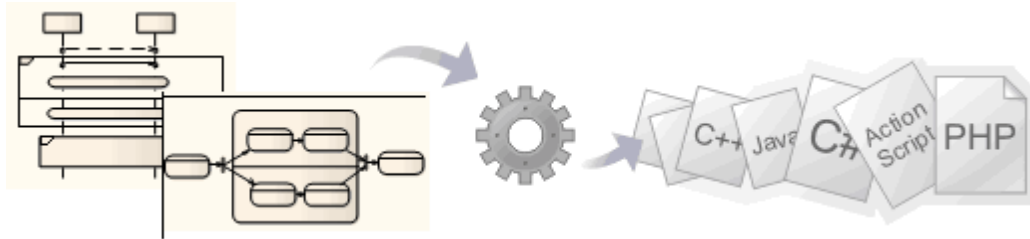
- When performing code generation, any Package name that contains whitespace characters is automatically treated as a namespace root

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Software Engineering | Generate Code | Namespaces**



## 11.5 Generate From Behavioral Models



Using Enterprise Architect's system engineering capability, you can generate code in various **software** (SW) and **hardware** (HW) languages, including C(OO), C++, C#, Java, VB.Net, VHDL, Verilog and SystemC, from these UML behavioral diagrams:

- State Machine diagrams (SW & HW)
- Interaction (Sequence) diagrams (SW)
- Activity diagrams (SW)

### Generate code from behavioral diagrams using the EAExample project

Step	Action	See also
1	Open the <i>EAExample.eap</i> file by selecting the <b>Help   Open Example Model</b> menu option.	
2	<p>From the Project Browser, select any of the following packages:</p> <p><b>Software Language Examples:</b></p> <ul style="list-style-type: none"> <li>• Project Models &gt; System Model &gt; Implementation Model (PSM) &gt; Java Model With Behaviors Generate the Account and Order classes</li> <li>• Project Models &gt; Systems Engineering Model &gt; Implementation Model &gt; Software &gt; C# Generate the DataProcessor Class</li> <li>• Project Models &gt; Systems Engineering Model &gt; Implementation Model &gt; Software &gt; C++ Generate the IO Class</li> <li>• Project Models &gt; Systems Engineering Model &gt; Implementation Model &gt; Software &gt; Java Generate the IO Class</li> <li>• Project Models &gt; Systems Engineering Model &gt; Implementation Model &gt; Software &gt; VBNet Generate the IO Class</li> </ul> <p><b>Hardware Language Examples:</b></p> <ul style="list-style-type: none"> <li>• Project Models &gt; Systems Engineering Model &gt; Implementation Model &gt; Hardware &gt; SystemC Generate the PlayBack Class</li> <li>• Project Models &gt; Systems Engineering Model &gt; Implementation Model &gt; Hardware &gt; VHDL Generate the PlayBack Class</li> <li>• Project Models &gt; Systems Engineering Model &gt; Implementation Model &gt; Hardware &gt; Verilog Generate the PlayBack Class</li> </ul>	
3	When completed, press ( <b>Ctrl+E</b> ) to open the generated source code.	

Step	Action	See also
	You should see methods generated in the code.	

### Notes

- Software code generation from behavioral models is available in the Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect
- Hardware code generation from State Machine models is available in the Systems Engineering and Ultimate editions of Enterprise Architect
- For C(OO), on the C Specifications page of the Options dialog set the **Object Oriented Support** option to **True**
- To be able to generate code from behavioral models, all behavioral constructs should be contained within a Class
- Code synchronization is not supported for behavioral code

### Learn more

- [Legacy State Machine Templates](#) <sup>[2126]</sup>
- [Interaction \(Sequence\) diagrams](#) <sup>[2133]</sup>
- [Activity diagrams](#) <sup>[2134]</sup>
- [State Machine modeling for HDLs](#) <sup>[2131]</sup>
- [Behavior](#) <sup>[1014]</sup>
- [Object Oriented Programming In C](#) <sup>[2086]</sup>

### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Software Engineering | Behavioral Code Generation**

## 11.5.1 Code Generation - State Machines

A **State Machine** illustrates how an **object** (represented by a **Class**) can change **state**, each change of state being a **transition** initiated by a **trigger** arising from an **event**, often under conditions or constraints defined as **guards**. As you model how the object changes state, you can **generate** and **build** (compile) code from it in the appropriate language and **execute** the code, visualizing the execution via the **model Simulator**.

It is also possible, in Enterprise Architect, to **combine** the State Machines of separate but related objects to see how they interact (via Broadcast Events), and to quickly create and generate code from **variants** of the model. For example, you might model the behavior of:

- The rear off-side wheel of a vehicle in rear-wheel drive and front-wheel drive modes (one State Machine)
- The steering wheel and all four drive wheels of a vehicle in 4-wheel drive mode (five State Machines)
- The wheels of an off-road vehicle and of a sports car (two Artifacts, instances of a combination of State Machines)

Of critical importance in generating and testing code for all of these options is the **Executable State Machine Artifact** element. This acts as the container and code generation unit for your State Machine

models.




### Prerequisites

- Select **Tools | Options | Objects** and select the **Port and Part type visible by default** checkbox
- Select **Tools | Options | Source Code Engineering** and, for the appropriate coding language (**Java**, **C**, **C#** or **ANSI C++**), set the **Use the new StateMachine Template** option to **True**
- If working in **C++**, select **Tools | Options | Source Code Engineering | C++** and set the **C++ Version** option to **ANSI**



This code generation method does not apply to the Legacy State Machine code generation templates developed prior to Enterprise Architect Release 11.

**Access**    **Diagram | Diagram Toolbox > Artifacts**

### Prepare your State Machine diagram(s)

Step	Action	See also
1	For each State Machine you want to model, create a Class diagram.	<a href="#">Add New Diagrams</a> 
2	From the Class Elements page of the Diagram Toolbox, drag the <b>Class</b> icon onto your diagram and give the element an appropriate name.	<a href="#">Create Elements</a> 
3	Right-click on the Class element and select the <b>New Child Diagram   State Machine</b> menu option.  Give the State Machine diagram an appropriate name.	
4	Create the State Machine model to reflect the appropriate transitions between States.	<a href="#">State Machines</a> 

### Set up the Executable State Machine Artifact

Step	Action	See also
1	Create a new Class diagram to contain the modeled State Machine(s) from which you intend to generate code.	<a href="#">Add New Diagrams</a> 
2	From the Artifacts page of the Diagram Toolbox, drag the <b>Executable StateMachine</b> icon onto the diagram to create the Artifact element. Name the element and drag its borders out to enlarge it.	<a href="#">Artifact</a> 



Step	Action	See also
3	<p>From the Project Browser, drag the (first) <b>Class</b> element containing a State Machine diagram onto the Artifact element on the diagram.</p> <p>The &lt;element name&gt;: Paste As dialog displays, with the <b>Paste as</b> field defaulted to <b>Property</b>.</p> <p>(If the dialog does not display, press <b>Ctrl</b> as you drag the Class element from the Project Browser.)</p>	<a href="#">Paste from Project Browser</a> <sup>[833]</sup>
4	Click on the <b>OK</b> button. The Class element is pasted inside the Artifact as a Part.	<a href="#">Part</a> <sup>[1383]</sup>
5	<p>Repeat steps 3 and 4 for any other Classes with State Machines that you want to combine and generate code for. These might be:</p> <ul style="list-style-type: none"> <li>• Repeat 'drops' of the same Class and State Machine, modeling parallel objects</li> <li>• Different Classes and State Machines, modeling separate interacting objects</li> </ul>	
6	<p>Right-click on the Artifact element and select the <b>Properties</b> option, and in the <b>Language</b> field click on the drop-down arrow and set the code language to the same language as is defined for the Class elements.</p> <p>You can now drag this <b>Executable StateMachine Artifact</b> element from the Project Browser onto the diagram any number of times, and modify the Parts to model variations of the system or process, or the same system or process with different programming languages.</p>	




#### Generate Code From Artifact

Step	Action	See also
1	<p>Right-click on the Executable StateMachine Artifact element and select the <b>Code Engineering   Executable StateMachine   Generate</b> menu option.</p> <p>The Executable StateMachine Code Generation dialog displays.</p>	
2	<p>In the <b>Project output directory</b> field, type or browse for the directory pathname under which to create the output files.</p> <p>During code generation, <b>all existing files in this directory are deleted</b>.</p>	
3	<p>In the <b>Location of &lt;compiler&gt; installation directory</b> field, type or browse for the path of the compiler installation directory, to be automatically mapped to the local path (displayed to the left of the field). For each programming language, these might resemble the following:</p> <ul style="list-style-type: none"> <li>• Java</li> </ul>	

Step	Action	See also
	<p>JAVA_HOME    <i>C:\Program Files (x86)\Java\jdk1.7.0_17</i></p> <ul style="list-style-type: none"> <li>• C/C++ VC_HOME    <i>C:\Program Files (x86)\Microsoft Visual Studio 9.0</i></li> <li>• C# CS_HOME    <i>C:\Windows\Microsoft.NET\Framework\v3.5</i></li> </ul>	
4	<p>Click on the <b>Generate</b> button. The code files are created appropriate to the programming language.</p> <p>The System Output window displays with an Executable State Machine Output tab, showing the progress and status of the generation.</p> <p>During code generation, an automatic <b>validation</b> function is executed to check for diagram or model errors against the UML constraints. Any errors are identified by error messages on the Executable State Machine Output tab.</p> <p>Double-click on an error message to display the modeling structure in which the error occurs, and correct the mistake before re-generating the code.</p>	
5	<p>When the code generates without error, right-click on the Artifact element and select the <b>Code Engineering   Executable Statemachine   Build</b> menu option to compile the code.</p> <p>The System Output window displays with a Build tab, showing the progress and status of the compilation. Notice that the compilation includes configuration of the simulation operation (below).</p>	

#### Execute/Simulate Code From Artifact

Step	Action	See also
1	<p>Select the two menu options:</p> <ul style="list-style-type: none"> <li>• <b>Analyzer   Simulator</b> to display the Simulation window</li> <li>• <b>Analyzer   Simulation Events</b> to display the Simulation Events window</li> </ul> <p>Dock the two windows in a convenient area of the screen.</p>	<p><a href="#">Simulation Windows</a>  <sup>[2469]</sup></p> <p><a href="#">Simulation Events Window</a>  <sup>[2504]</sup></p>
2	<p>On the diagram or Project Browser, right-click on the Artifact element and select the <b>Code Engineering   Executable Statemachine   Run</b> menu option.</p> <p>The first State Machine diagram in the series displays with the simulation of the process already started. In the Simulation window, the processing steps are indicated in this format:</p> <p>[ 03516677]      Part 1[ Class1] . I n i t i a l _ 3 6 7 _ T O _ S t a t e 4 _ 1 4 2 E f f e c t [ 03516683]      Part 1[ Class1] . S t a t e M a c h i n e _ S t a t e 4 _ E N T R Y</p>	

Step	Action	See also
	[ 03516684]      Part 1[ Class1] . StateMachine_State4 DO [ 03518375]      Blocked	
3	Click on the appropriate Simulation window toolbar buttons to step through the simulation as you prefer.  When the simulation finishes at the Exit or Terminate element, click on the <b>Stop</b> button in the Simulation window toolbar.	<a href="#">Run Model Simulation</a>  [2475]
4	Where the trace shows <b>Blocked</b> , the simulation has reached a point where a Trigger event has to occur before processing can continue. On the Simulation Events window, in the <b>Waiting Triggers</b> column, double-click on the appropriate Trigger.  When the Trigger is fired, the simulation continues to the next pause point, Trigger or exit.  <a href="http://www.sparxsystems.com/enterprise_architect_user_guide/9.3/domain_based_models/spem_toolbox_pages.html">http://www.sparxsystems.com/enterprise_architect_user_guide/9.3/domain_based_models/spem_toolbox_pages.html</a>	<a href="#">Triggers</a>  [2494] <a href="#">Waiting Triggers</a>  [2508]

**Notes**

- If you are making small changes to an existing State Machine model, you can combine the code generation, build and run operations by selecting the **Code Engineering | Executable Statemachine | Generate, Build and Run** option
- You can also generate code in **JavaScript** and **System C**

**Learn more**

- [Broadcast Events](#)  [3029]

**11.5.1.1 Legacy State Machine Templates**

Code generation operates using a set of generation templates. From Release 11.0 of Enterprise Architect, a different set of templates are available as the default for software code generation from a State Machine diagram into Java, C, ANSI C++ or C# code. You can still use the original templates, as described here, for models developed in earlier releases of Enterprise Architect, if you do not want to upgrade them for the new template facilities.

**Switch Between Legacy and Release 11 templates**

**Access**    **Tools | Options | Source Code Engineering | <language name>**

If necessary, expand the *Statemachine Engineering (for current model)* grouping and set the **Use the new Statemachine Template** option to **True** (to use the later templates) or **False** (to use the Legacy templates).

**Legacy Template Transformations**

A State Machine in a *Class* internally generates a number of constructs in software languages to provide effective execution of the States' behaviors (**do**, **entry** and **exit**) and also to code the appropriate transition's effect when necessary.

Model Objects	Code Objects	See also
<b>Enumerations</b>	<ul style="list-style-type: none"> <li>• <b>StateType</b> - comprises an enumeration for each of the States contained within the State Machine</li> <li>• <b>TransitionType</b> – comprises an enumeration for each transition that has a valid effect associated with it; for example: <i>ProcessOrder_Delivered_to_ProcessOrder_Closed</i></li> <li>• <b>CommandType</b> – comprises an enumeration for each of the behavior types that a State can contain (Do, Entry, Exit)</li> </ul>	
<b>Attributes</b>	<ul style="list-style-type: none"> <li>• <b>currState:StateType</b> - a variable to hold the current State's information</li> <li>• <b>nextState:StateType</b> - a variable to hold the next State's information, set by each State's transitions accordingly</li> <li>• <b>currTransition:TransitionType</b> - a variable to hold the current transition information; this is set if the transition has a valid effect associated with it</li> <li>• <b>transcend:Boolean</b> - a flag used to advise if a transition is involved in transcending between different State Machines (or Submachine states)</li> <li>• <b>xx_history:StateType</b> - a history variable for each State Machine/ Submachine State, to hold information about the last State from which the transition took place</li> </ul>	
<b>Operations</b>	<ul style="list-style-type: none"> <li>• <b>StatesProc</b> - a States procedure, containing a map between a State's enumeration and its operation; it de-references the current State's information to invoke the respective State's function</li> <li>• <b>TransitionsProc</b> - a Transitions procedure, containing a map between the Transition's enumeration and its effect; it invokes the Transition's effect</li> <li>• <b>&lt;&lt;State&gt;&gt;</b> - an operation for each of the States contained within the State Machine; this renders a State's behaviors based on the input CommandType, and also executes its transitions</li> <li>• <b>initializeStateMachine</b> - a function that initializes all the framework-related attributes</li> <li>• <b>runStateMachine</b> - a function that iterates through each State, and executes their behaviors and transitions accordingly</li> </ul>	<a href="#">Operation Behavior</a> <sup>[1018]</sup>

#### Notes

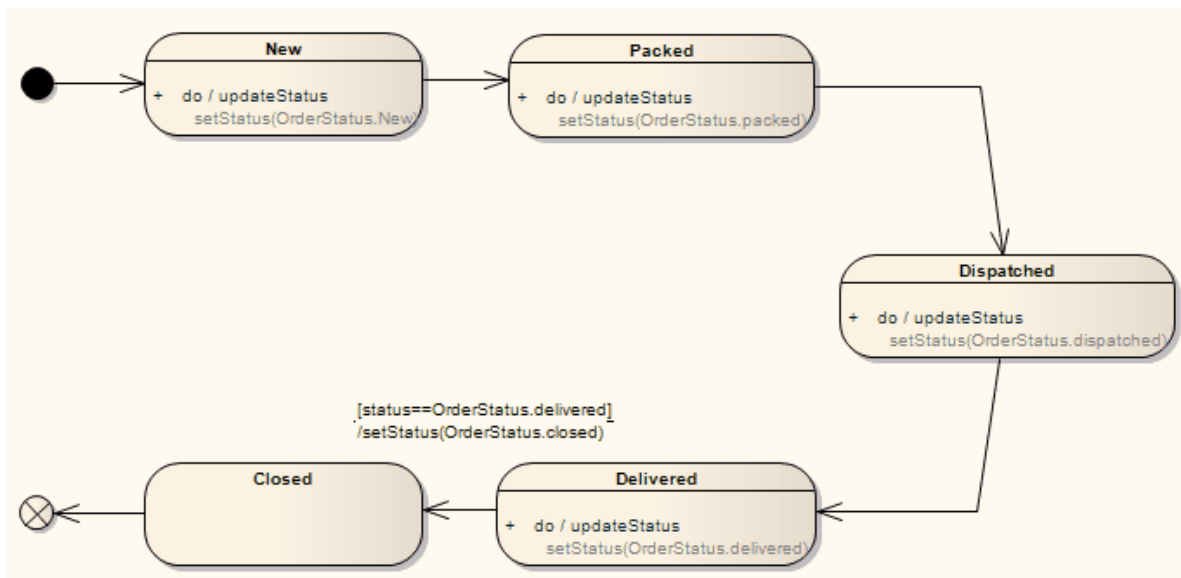
- To be able to generate code from behavioral models, all behavioral constructs should be contained within a Class

Learn more

- [State Machine Diagrams](#) <sup>[1203]</sup>
- [Java Code Generated From Legacy State Machine Template](#) <sup>[2128]</sup>
- [Generate From Behavioral Models](#) <sup>[2121]</sup>
- [State Machine Modeling For HDLs](#) <sup>[2137]</sup>

Learning Center topics

- (Alt+F1) | **Enterprise Architect | Software Engineering | Behavioral Code Generation | State Machines**

**11.5.1.1.1 Java Code Generated From Legacy State Machine Template**

```

private enum StateType : int
{
    ProcessOrder_Delivered,
    ProcessOrder_Packed,
    ProcessOrder_Closed,
    ProcessOrder_Dispatched,
    ProcessOrder_New,
    ST_NOSTATE
}
private enum TransitionType : int
{
    ProcessOrder_Delivered_to_ProcessOrder_Closed,
    TT_NOTRANSITION
}
private enum CommandType
{
    Do,
    Entry,
    Exit
}
private StateType currState;
private StateType nextState;
  
```



```

private TransitionType currTransition;
private boolean transcend;
private StateType ProcessOrder_history;
private void processOrder_Delivered(CommandType command)
{
    switch(command)
    {
        case Do:
        {
            // Do Behaviors..
            setStatus(Delivered);
            // State's Transitions
            if((status==Delivered))
            {
                nextState = StateType.ProcessOrder_Closed;
                currTransition = TransitionType.
ProcessOrder_Delivered_to_ProcessOrder_Closed;
            }
            break;
        }
        default:
        {
            break;
        }
    }
}

private void processOrder_Packed(CommandType command)
{
    switch(command)
    {
        case Do:
        {
            // Do Behaviors..
            setStatus(Packed);
            // State's Transitions
            nextState = StateType.ProcessOrder_Dispatched;
            break;
        }
        default:
        {
            break;
        }
    }
}

private void processOrder_Closed(CommandType command)
{
    switch(command)
    {
        case Do:
        {
            // Do Behaviors..
            // State's Transitions
            break;
        }
        default:
        {
            break;
        }
    }
}

private void processOrder_Dispatched(CommandType command)
{
    switch(command)
    {
        case Do:
        {
            // Do Behaviors..
            setStatus(Dispatched);

```

```

        // State's Transitions
        next State = StateType.ProcessOrder_Delivered;
        break;
    }
    default:
    {
        break;
    }
}

private void processOrder_New( CommandType command)
{
    switch( command)
    {
        case Do:
        {
            // Do Behaviors..
            set Status( new);
            // State's Transitions
            next State = StateType.ProcessOrder_Packed;
            break;
        }
        default:
        {
            break;
        }
    }
}

private void StatesProc( StateType currState, CommandType command)
{
    switch( currState)
    {
        case ProcessOrder_Delivered:
        {
            processOrder_Delivered( command);
            break;
        }

        case ProcessOrder_Packed:
        {
            processOrder_Packed( command);
            break;
        }

        case ProcessOrder_Closed:
        {
            processOrder_Closed( command);
            break;
        }

        case ProcessOrder_Dispatched:
        {
            processOrder_Dispatched( command);
            break;
        }

        case ProcessOrder_New:
        {
            processOrder_New( command);
            break;
        }
        default:
        {
            break;
        }
    }
}

private void TransitionsProc( TransitionType transition)
{
    switch( transition)
    {
        case ProcessOrder_Delivered_to_ProcessOrder_Closed:

```

```

        {
            set Status(closed);
            break;
        }
        default:
            break;
    }
}
private void initializeStateMachine()
{
    currState = StateType.ProcessOrder_New;
    nextState = StateType.ST_NOSTATE;
    currTransition = TransitionType.TT_NOTTRANSITION;
}

private void runStateMachine()
{
    while(true)
    {
        if ( currState == StateType.ST_NOSTATE )
        {
            break ;
        }

        currTransition = TransitionType.TT_NOTTRANSITION;
        StatesProc(currState, CommandType.Do);
        // then check if there is any valid transition assigned after the do
        behavior
        if ( nextState == StateType.ST_NOSTATE)
        {
            break;
        }

        if ( currTransition != TransitionType.TT_NOTTRANSITION )
        {
            TransitionsProc( currTransition );
        }
        if ( currState != nextState)
        {
            StatesProc(currState, CommandType.Exit);
            StatesProc(nextState, CommandType.Entry);
            currState = nextState ;
        }
    }
}

```

### 11.5.2 State Machine Modeling For HDLs

For efficient code generation from State Machine models into Hardware Description Languages (HDL) such as VHDL, Verilog and SystemC, apply these design practices.

In an HDL State Machine model, the following are expected:

- Designate Driving Triggers
- Establish Port–Trigger Mapping
- Active State Logic

#### How to



Topic	Detail	See also
<b>Designate Driving Triggers</b>	<ul style="list-style-type: none"> <li>• A 'change' trigger is deemed as an asynchronous trigger if the following two conditions are satisfied:</li> </ul>	<a href="#">State Machine Diagrams</a>

	<div><div><div>1203</div><div>Trigger</div><div>1350</div></div><div><div><div>1. There is a transition from the actual submachine state (which encapsulates the actual logic) triggered by it</div><div>2. The target state of that transition has a self transition triggered by the same trigger</div></div><div><div>• Asynchronous triggers should be modeled according to the following pattern:<div><div>1. The trigger should be of type Change (specification: true / false)</div><div>2. The active state (Submachine State) should have a transition trigger by it.</div><div>3. The target state of the triggered transition should have a self transition with the same trigger</div></div></div><div>• A trigger of type <i>time</i>, which triggers the transitions to the active state (Submachine State) is deemed as the <i>Clock</i>; the specification of this trigger should be specific to the target language:</div></div><table><tr><th rowspan="2">Trigger Type</th><th rowspan="2">Language</th><th colspan="2">Specification</th></tr><tr><th>Positive Edge Triggered</th><th>Negative Edge</th></tr><tr><td rowspan="3">Time</td><td>VHDL</td><td>rising_edge</td><td>falling_edge</td></tr><tr><td>Verilog</td><td>posedge</td><td>negedge</td></tr><tr><td>SystemC</td><td>positive</td><td>negative</td></tr></table></div></div>	Trigger Type	Language	Specification		Positive Edge Triggered	Negative Edge	Time	VHDL	rising_edge	falling_edge	Verilog	posedge	negedge	SystemC	positive	negative
Trigger Type	Language			Specification													
		Positive Edge Triggered	Negative Edge														
Time	VHDL	rising_edge	falling_edge														
	Verilog	posedge	negedge														
	SystemC	positive	negative														
<div>Establish Port-Trigger Mapping</div>	<div><div>After successfully modeling the different operating modes of the component, and the triggers associated with them, you must associate the triggers with the component's ports</div><div>A Dependency relationship from the Port to the associated trigger is used to signify their association</div><div><div><div>class HDL</div><div><div><div><div><div>reset</div><div>clear</div><div>clock</div></div><div><div><div>A dependency relationship is used to represent association between ports and their triggers.</div></div></div></div><div><div><div>ActiveClass</div><div><div>reset</div><div>clear</div><div>clock</div></div></div></div></div></div></div></div></div>																
<div>Active State Logic</div>	<div><div>The first two aspects, above, put in place the preliminaries required for efficient interpretation of the hardware components.</div><div>The actual State Machine logic is now modeled within the Active (Submachine) state.</div></div>																

**Notes**

- To be able to generate code from behavioral models, all behavioral constructs should be contained within a Class
- The current code generation engine supports only one clock trigger for a component














**Learn more**

- [Legacy State Machine Templates](#) 
- [Transition](#) 

**11.5.3 Code Generation - Interaction Diagrams**

During code generation from Interaction (Sequence) diagrams in a Class, Enterprise Architect applies its system engineering graph optimizer to transform the Class constructs into programmatic paradigms. Messages and Fragments are identified as one of the several action types based on their functionality, and Enterprise Architect uses the *code generation templates* to render their behavior accordingly.

**Topics**

Topic	Detail	See also
<b>Action Call</b>	A Message that invokes an operation.	<a href="#">Message (Sequence Diagram)</a>  <a href="#">Call</a>  <a href="#">Message Examples</a> 
<b>Action Create</b>	A Message with <i>Lifecycle = New</i> .	<a href="#">Message (Sequence Diagram)</a>  <a href="#">Message Properties</a> 
<b>Action Destroy</b>	A Message with Lifecycle = Delete.	<a href="#">Message (Sequence Diagram)</a>  <a href="#">Message Properties</a> 
<b>Action Loop</b>	A Combined Fragment with Type = Alt.	<a href="#">Combined Fragment</a>  <a href="#">Interaction Operators</a> 
<b>Action If</b>	A Combined Fragment with Type = loop.	<a href="#">Combined Fragment</a>  <a href="#">Interaction Operators</a> 
<b>Assign To</b>	A call message with a valid target attribute set using the <b>Assign To</b> field is rendered in the code as the target attribute of a call action.	<a href="#">Call</a>  <a href="#">Message Properties</a> 

**Notes**

- To be able to generate code from behavioral models, all behavioral constructs should be contained within a Class
- For an Interaction (Sequence) diagram, the behavioral code generation engine expects the Sequence diagram and all its associated messages and interaction fragments to be encapsulated within an interaction element

**Learn more**

- [EASL Code Generation Macros](#) <sup>[1688]</sup>
- [Interaction](#) <sup>[1314]</sup>
- [Interaction Occurrence](#) <sup>[1317]</sup>
- [Activity](#) <sup>[1279]</sup>

**Learning Center topics**



- (Alt+F1) | **Enterprise Architect | Software Engineering | Behavioral Code Generation | Interaction Diagrams**

**11.5.4 Code Generation - Activity Diagrams**

Code generation from Activity diagrams in a Class requires a validation phase, during which Enterprise Architect uses the system engineering graph optimizer to analyze the diagram and render it into various constructs from which code can be generated. Enterprise Architect also transforms the constructs into one of the various action types (if appropriate), similar to the Interaction diagram constructs.

**Topics**






Topic	Detail	See also
<b>Call Actions (Invocation Actions)</b>	<p>Used to invoke operations or behaviors in an Activity diagram; the two main variants of Call Actions supported in behavioral code generation are:</p> <ul style="list-style-type: none"> <li>• CallOperation Action - used to invoke operations, which can be within the same Class or in other Classes within the same package; if referencing operations from other Classes within the same package, you must have a target to which the request is passed</li> <li>• CallBehavior Action - used to invoke another Activity in an activity flow; the referenced Activity is expected to be within the same Class</li> </ul> <p><b>Arguments</b></p> <p>Call Actions can specify argument values corresponding to the parameters in the associated behavior or behavioral feature.</p> <p>You can add the arguments manually or create them automatically using the <b>Synchronize</b> button of the Arguments dialog.</p>	<p><a href="#">Assign Action Pins</a> <sup>[1278]</sup></p> <p><a href="#">Behavior Calls</a> <sup>[1026]</sup></p> <p><a href="#">Synchronize Arguments</a> <sup>[1027]</sup></p>

<b>CreateObjectAction</b>	Used to denote an object creation in the activity flow; you can set the result Pin of the CreateObjectAction as the object to be created, using the Assign Action Pins dialog.  The Classifier of the CreateObjectAction signifies the Classifier for which an instance is to be created.	<a href="#">Assign Action Pins</a> 
<b>DestroyObjectAction</b>	Used to denote an object deletion in the activity flow; you can set the target Pin of the DestroyObjectAction as the object to be destroyed, using the Assign Action Pins dialog.	<a href="#">Assign Action Pins</a> 
<b>Loops</b>	Enterprise Architect's system engineering graph optimizer is also capable of analyzing and identifying loops; an identified loop is internally rendered as an Action Loop, which is translated by the EASL code generation macros to generate the required code.  You can have a single loop, nested loops, and multiple levels of nested loops.	
<b>Conditional Statements</b>	To model a conditional statement, you use Decision/Merge nodes.  Alternatively, you can imply Decisions/Merges internally; the graph optimizer expects an associated Merge node for each Decision node, to facilitate efficient tracking of various branches and analysis of the code constructs within them.	

**Notes**

- To be able to generate code from behavioral models, all behavioral constructs should be contained within a Class

**Learn more**

- [EASL Code Generation Macros](#) 
- [Assign Action Pins](#) 
- [Interactions and Activities](#) 
- [Behavior Calls](#) 
- [Synchronize Arguments](#) 
- Resource:** EAExample Model > Project Models > Software Engineering > Java Model With Behaviors > TestSelectionSort

**Learning Center topics**

- (Alt+F1) | **Enterprise Architect | Software Engineering | Behavioral Code Generation | Activity Diagrams**

## 11.6 Importing Source Code



When modeling in Enterprise Architect, it is possible to **import - reverse engineer** - existing **source code** as a single directory or a directory structure, in a variety of code languages, into your **UML model**. By reverse engineering code you can examine, for example, the functionality of code libraries for reuse, or bring the UML model up to date with new code from outside the model; that is, **synchronize** the code and model. The code structures are mapped into their UML representations; for example, a Java Class is mapped into a UML Class element, variables are defined as attributes, methods modeled as operations, and interactions between the Java Classes represented by the appropriate connectors.

By examining the code in a UML model, you can identify the critical modules containing the code, providing a starting point for understanding the business and system requirements of the existing application and a framework for gaining a better overall understanding of the source code.

You have several options in performing reverse engineering, including:

- Whether comments are reverse engineered into **Notes** fields, and how they are formatted if they are
- How property methods are recognized
- Whether dependencies should be created for operation return and parameter types

It is important to note that if a system is not well designed, simply importing the source into Enterprise Architect does not turn it into an easily understandable UML model. When working with a poorly designed system it is useful to assess the code in manageable units by examining the individual model packages or elements generated from the code; for example, dragging a specific Class of interest onto a diagram and then using the **Insert Related Elements** option at one level to determine the Class's immediate relationships to other Classes. From this point it is possible to create Use Cases that identify the interaction between the source code Classes, providing an overview of the application's operation.

### Copyright Ownership

Situations that typically lend themselves to reverse engineering tend to operate on source code that:















- You have already developed
- Is part of a third-party library that you have obtained permission to use
- Is part of a framework that your organization uses
- Is being developed on a daily basis by your developers

If you are examining code that you or your organization do not own or do not have specific permission to copy and edit, you must ensure that you understand and comply with the copyright restrictions on that code before beginning the process of reverse engineering.

### Supported languages for Reverse Engineering

Topic	Link
-------	------



Action Script	<a href="#">ActionScript</a>  [2081]
Ada 2005 (Systems Engineering and Ultimate editions)	<a href="#">Ada 2005</a>  [2082]
C	<a href="#">C</a>  [2084]
C#	<a href="#">C#</a>  [2088]
C++	<a href="#">C++</a>  [2090]
CORBA IDL (MDG Technology)	<a href="http://www.sparxsystems.com/resources/mdg_tech/">www.sparxsystems.com/resources/mdg_tech/</a>
Delphi	<a href="#">Delphi</a>  [2095]
Java	<a href="#">Java</a>  [2096]
PHP	<a href="#">PHP</a>  [2099]
Python	<a href="#">Python</a>  [2100]
SystemC (Systems Engineering and Ultimate editions)	<a href="#">SystemC</a>  [2100]
Verilog (Systems Engineering and Ultimate editions)	<a href="#">Verilog</a>  [2105]
VHDL (Systems Engineering and Ultimate editions)	<a href="#">VHDL</a>  [2106]
Visual Basic	<a href="#">Visual Basic</a>  [2109]
Visual Basic .NET	<a href="#">Visual Basic .NET</a>  [2102]

### Notes

- Reverse Engineering is supported in the Professional, Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions of Enterprise Architect
- If security is enabled you must have **Reverse Engineer From DDL And Source Code** permission to reverse engineer source code and synchronize model elements against code
- Using Enterprise Architect, you can also import certain types of binary files, such as Java .jar files and .

NET PE files

- Reverse Engineering of other languages is currently available through the use of MDG Technologies listed on the MDG Technology pages of the Sparx Systems website

#### Learn more

- [Import Source Code](#)<sup>[2138]</sup>
- [Import a Directory Structure](#)<sup>[2142]</sup>
- [Source Code Options](#)<sup>[2247]</sup>
- [Insert Related Elements](#)<sup>[933]</sup>
- [Import Binary Module](#)<sup>[2143]</sup>
- [Permission List](#)<sup>[329]</sup>
- [www.sparxsystems.com/resources/mdg\\_tech/](http://www.sparxsystems.com/resources/mdg_tech/) (Online Resource)

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Software Engineering | Import Code**

### 11.6.1 Import Source Code

#### Import source code (reverse engineer)

Step	Action	See also
1	In the Project Browser, select (or add) a diagram into which to import the Classes.	
2	Right-click on the diagram background to open the context menu and either: <ul style="list-style-type: none"> <li>Select the language to import from the <b>Import from source file(s)</b> submenu</li> <li>Click on the <b>Import Language</b> drop-down arrow in the Code Generation toolbar and select the <b>Import   Import xxx files</b> menu option, where xxx represents the language to import</li> </ul>	
3	From the file browser that appears, select one or more source code files to import.	<a href="#">Notes on Source Code Import</a> <sup>[2139]</sup>
4	Click on the <b>Open</b> button to start the import process.	

As the import proceeds, Enterprise Architect provides progress information. When all files are imported, Enterprise Architect makes a second pass to resolve associations and inheritance relationships between the imported Classes.

Learning Center topics


- (Alt+F1) | **Enterprise Architect | Software Engineering | Import Source Code**

**11.6.2 Notes on Source Code Import**

Enterprise Architect enables you to import code into your project, in a range of programming languages.

Topics

Topic	Detail	See also
<b>Introduction</b>	Enterprise Architect supports most constructs and keywords for each coding language.  If there is a particular feature you require support for that you feel is missing, please contact Sparx Systems.  You must select the appropriate type of source file for the language, as the source code to import.	<a href="#">Sparx Systems</a> (Online resource)
<b>ActionScript</b>	Appropriate type of source file: <b>.as</b> code file.	<a href="#">ActionScript</a> [2081]
<b>C</b>	Appropriate type of source file: <b>.h</b> header files and/or <b>.c</b> files.  When you select a header file, Enterprise Architect automatically searches for the corresponding <b>.c</b> implementation file to import, based on the options for extension and search path specified in the C options.  Enterprise Architect does not expand macros that have been used, these must be added into the internal list of Language Macros.	<a href="#">C</a> [2084]  <a href="#">C options</a> [2266]  <a href="#">Language Macros</a> [2257]
<b>C++</b>	Appropriate type of source file: <b>.h</b> header file.  Enterprise Architect automatically searches for the <b>.cpp</b> implementation file based on the extension and search path set in the C++ options; when it finds the implementation file, it can use it to resolve parameter names and method notes as necessary.  When importing C++ source code, Enterprise Architect ignores function pointer declarations.  To import them into your model you could create a <i>typedef</i> to define a function pointer type, then declare function pointers using that type; function pointers declared in this way are imported as attributes of the function pointer type.  Enterprise Architect does not expand macros that have been used; these must be added into the internal list of Language Macros.	<a href="#">C++</a> [2090]  <a href="#">C++ options</a> [2269]  <a href="#">Language Macros</a> [2257]
<b>C#</b>	Appropriate type of source file: <b>.cs</b> .	<a href="#">C#</a> [2088]  <a href="#">C# Options</a> [2268]
<b>Delphi</b>	Appropriate type of source file: <b>.pas</b> .	<a href="#">Delphi</a> [2095]

<b>Java</b>	<p>Appropriate type of source file: <b>.java</b>.</p> <p>Enterprise Architect supports the <b>AspectJ</b> language extensions.</p>  <p>Aspects are modeled using Classes with the stereotype <i>aspect</i>; these aspects can then contain attributes and methods as for a normal Class.</p> <p>If an <i>intertype</i> attribute or operation is required, you can add a tag <i>className</i> with the value being the name of the Class it belongs to.</p> <p><i>Pointcuts</i> are defined as operations with the stereotype <i>pointcut</i>, and can occur in any Java Class, Interface or aspect; the details of the <i>pointcut</i> are included in the behavior field of the method.</p> <p><i>Advice</i> is defined as an operation with the stereotype <i>advice</i>; the <i>pointcut</i> this advice operates on is in the <b>behavior</b> field and acts as part of the method's unique signature.</p> <p><i>afterAdvice</i> can also have one of the Tagged Values <b>returning</b> or <b>throwing</b>.</p>	<a href="#">Java</a> <sup>[2096]</sup> <a href="#">Java Options</a> <sup>[2272]</sup>
<b>PHP</b>	<p>Appropriate type of source file: <b>.php, .php4, or .inc</b>.</p> <p>Nested if condition syntax is enabled.</p>	<a href="#">PHP</a> <sup>[2099]</sup>
<b>Python</b>	<p>Appropriate type of source file: <b>.py</b>.</p>	<a href="#">Python</a> <sup>[2100]</sup>
<b>Visual Basic</b>	<p>Appropriate type of source file: <b>.cls</b> Class file.</p>	<a href="#">Visual Basic</a> <sup>[2109]</sup>
<b>Visual Basic .NET</b>	<p>Appropriate type of source file: <b>.vb</b> Class file.</p>	<a href="#">Visual Basic .NET</a> <sup>[2102]</sup>

### Notes

- When reverse engineering attributes with parameter substitutions (templated attributes):
  - If a Class with proper template parameter definitions is found, an Association connector is created and its parameter substitutions are configured
  - An Association connector is also created if a matching entry is defined as a Collection Class or in

the **Additional Collection Classes** option (for C#, C++ and Java); for an example, see *Example Use of Collection Classes*

#### Learn more

- [Import Source Code](#) <sup>[2138]</sup>
- [Parameterized Classes \(Templates\)](#) <sup>[1365]</sup>
- [Set Collection Classes](#) <sup>[2259]</sup>
- [Example Use of Collection Classes](#) <sup>[2260]</sup>

### 11.6.3 Import Resource Script

Enterprise Architect supports the import and export of Microsoft Windows™ **Resource Scripts** (as **.rc** files), which contain the Win32 dialog definitions (those with the stereotype «*win32Dialog*») for an application's graphical user interface. Dialog resources are imported and exported for a specific location, which defaults to the location of the current computer system.

**Access**    **Right-click on a Package | Code Engineering | Import Resource Script**  
**Right-click on a Screen element | Code Engineering | Synchronize with Code (F7)**

#### Import dialog resources from a .rc file

Field/Button/Option	Action	See also
<b>Resource File</b>	Click on the ( ... ) Browse button and locate the .rc file to import the screen elements(s) from.	
<b>Resource ID</b>	Either: <ul style="list-style-type: none"> <li>• Leave the default value <b>All</b> to import all screen elements from the file, or</li> <li>• Click on the drop-down arrow and select the <b>screen ID</b> of a specific dialog to import</li> </ul>	
<b>Language</b>	Click on the drop-down arrow and select the language version (such as <b>English - United States</b> ) of the dialog(s) to import.	
<b>Import</b>	Click on this button to import the screens from the resource file.  The progress of the import is reported in the field underneath the <b>Language</b> field.	

#### Export a dialog to a .rc file

Field/Button/ Option	Action	See also
<b>Screen ID</b>	Defaults from the Win32UI <b>ID</b> Tagged Value of the selected Screen element.  (If the dialog does not have this ID, open the Win32UI page of the element's Properties dialog and provide a value for the <b>ID</b> tag.)	
<b>Resource File</b>	Click on the ( ... ) Browse button and locate the .rc file into which to export the screen elements(s).  If the element was previously imported, this field defaults to the source file.	
<b>Language</b>	Click on the drop-down arrow and select the language version (such as <b>English - United States</b> ) of the exported dialog.	
<b>Export</b>	Click on this button to export the screens from the resource file.  The progress of the export is reported in the field underneath the <b>Language</b> field.	

**Notes**

- New dialogs are exported to an **existing** .rc file
- In an export to an existing .rc file, no dialogs are ever deleted from the file, even when they are deleted from the model
- In an import, no dialogs are deleted from the model even when omitted from the original .rc file

**11.6.4 Import a Directory Structure**

You can import from all source files in a complete directory structure, which enables you to import or synchronize multiple files in a directory tree in one pass.

Enterprise Architect creates the necessary packages and diagrams during the import process.

**Access** [Project Browser package context menu | Code Engineering | Import Source Directory](#)

**Import a directory structure, using the Import Source Directory dialog**

Step	Action	See also
<b>1</b>	Select the options you require; you can configure: <ul style="list-style-type: none"> <li>• The source directory</li> <li>• The source type</li> <li>• The file extensions to look at</li> <li>• Whether to recurse sub directories</li> </ul>	

Step	Action	See also
	<ul style="list-style-type: none"> <li>Whether to create a diagram for each package</li> <li>Whether to import additional files as described in the Import Component Types dialog</li> <li>Whether to exclude private members from libraries being imported from the model</li> <li>Whether to <b>Synchronize</b> or <b>Overwrite</b> existing Classes when found; if a model Class is found matching the one in code: <ul style="list-style-type: none"> <li><b>Synchronize</b> updates the model Class to include the details from the one in code, which preserves information not represented in code, such as the location of Classes in diagrams</li> <li><b>Overwrite</b> deletes the model Class and generates a new one from code, which deletes and does not replace the additional information</li> </ul> </li> <li>Whether to create a package for every directory, namespace or file; this might be restricted depending on the source type selected</li> <li>How to handle Classes not found during the import (prompt for action enables you to review Classes individually)</li> <li>What is shown on diagrams created by the import</li> </ul>	<a href="#">Classes not found during Import</a> <sup>[2144]</sup>
2	Click on the <b>OK</b> button to start.	

#### Learning Center topics

- (Alt+F1) | [Enterprise Architect | Software Engineering | Import Code | Import Source Directory](#)

### 11.6.5 Import Binary Module

Enterprise Architect enables you to reverse-engineer certain types of binary modules.

**Access** [Right-click on a package within the Project Browser | Code Engineering | Import Binary Module](#)  
or  
[Tools | Source Code Engineering | Import Binary Module](#)

#### Topics

Topic	Detail	See also
<b>Usage</b>	<p>Currently the permitted types are as follows:</p> <ul style="list-style-type: none"> <li>Java Archive (.jar)</li> <li>.Net PE file (.exe, .dll) - Native Windows DLL and EXE files are <b>not</b> supported, only PE files containing .NET assembly data</li> <li>Intermediate Language file (.il)</li> </ul> <p>Enterprise Architect creates the necessary packages and diagrams during the</p>	

	<p>import process; selecting the <b>Do not import private members</b> checkbox excludes private members from libraries from being imported into the model.</p> <p>When importing .Net files, you can import via reflection or via disassembly, or let Enterprise Architect decide the best method - this might result in both types being used.</p> <p>The reflection-based importer relies on a .Net program, and requires the .Net runtime environment to be installed.</p> <p>The disassembler-based importer relies on a native Windows program called <i>lldasm.exe</i>, which is a tool provided with the MS .Net SDK; the SDK can be downloaded from the Microsoft website.</p> <p>A choice of import methods is available because some files are not compatible with reflection (such as <i>mscorlib.dll</i>) and can only be opened using the disassembler; however, the reflection-based importer is generally much faster.</p> <p>You can also configure:</p> <ul style="list-style-type: none"> <li>• Whether to <b>Synchronize</b> or <b>Overwrite</b> existing Classes when found; if a model Class is found matching the one in the file: <ul style="list-style-type: none"> <li>• <b>Synchronize</b> updates the model Class to include the details from the one in the file, which preserves information not represented in the file, such as the location of Classes in diagrams</li> <li>• <b>Overwrite</b> deletes the model Class and generates a new one from the file, which deletes and does not replace the additional information</li> </ul> </li> <li>• Whether to create a diagram for each package</li> <li>• What is shown on diagrams created by the import</li> </ul>	
--	---	--

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Software Engineering | Import Code | Import Compiled Java Libraries**
- (Alt+F1) | **Enterprise Architect | Software Engineering | Import Code | Import from .Net Assemblies**

### 11.6.6 Classes Not Found During Import

When reverse engineering from your code, there might be times when Classes are deliberately removed from your source code.

The **Import Source Directory** functionality keeps track of the Classes it expects to synchronize with and, on the Import Directory Structure dialog, provides options for how to handle the Classes that weren't found.

You can select the appropriate option to make Enterprise Architect, at the end of the import, ignore the missing Classes, automatically delete them or prompt you to manage them.

On the Import Directory Structure dialog, if you select the **Prompt For Action** radio button to manually review missing Classes, a dialog displays on which you specify the handling for each Class that was missing in the imported code.

By default, all Classes are marked for deletion; to keep one or more Classes, select them and click on the **Ignore** button.

[Learn more](#)



- [Import a Directory Structure](#) 

## 11.7 Editing Source Code

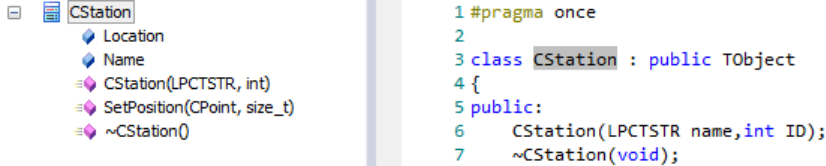
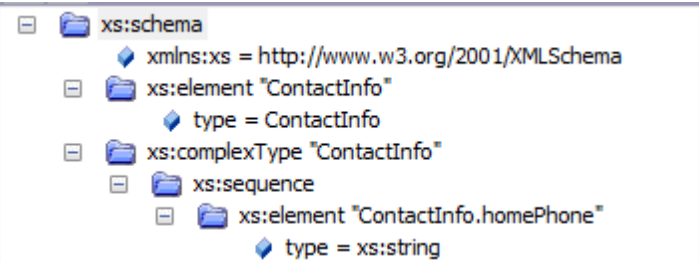
Use the Source Code editors to view and edit any source code files for an element. If a Class is selected in a diagram or the Project Browser, the Source Code editor shows the source code for that Class, provided it has already been *generated*. For C++ two tabs display, to show the implementation and header files.

A number of options change the way the Source Code editor works. They can be altered using the Options dialog (select the **Tools | Options | Source Code Engineering | Code Editors** menu option). There are also many options for developing the code, available through the Code Editor context menu, Toolbar or function keys.

There are several variations on the Source Code editor, with different access methods. The variants are discussed in the *Compare Editors* topic.

**Access** **Project Browser element context menu | View Source Code (F12)** or **(Ctrl+E)**  
**Diagram element context menu | View Source Code (F12)**  
**Element | Show Source Viewer (Alt+7)**  
**Tools | Open Source File (Ctrl+Alt+O)** (files not imported to or created in Enterprise Architect)

### Topics

Topic	Detail	See also
<b>Source Code editor</b>	<p>By default the Source Code editor is set to:</p> <ul style="list-style-type: none"> <li>• Parse all opened files, and show a tree of the results</li> <li>• Show line numbers</li> </ul>  <p>If you are editing an XML file, the structure tree is presented in a folder hierarchy rather than a Class structure hierarchy, as follows:</p> 	
<b>File Parsing</b>	<p>The Source Code editor parses files for a number of reasons; the first is to enable it to jump to the location in the file at which the currently selected item is found (but not when editing external files).</p> <p>Additionally, parsing displays a structure tree showing an overview of the file in a similar fashion to the main Project Browser; you can also select anything</p>	

Topic	Detail	See also
	in that and jump to the appropriate line in the editor.	

### Notes

- There are no *context menu options* to display XML or DDL.sql files; select the appropriate object and use the keyboard keys (**F12**, **Alt+7**)
- When you select an element to view source code, if the element does not have a generation file (that is, code has not been or cannot be generated, such as for a Use Case), Enterprise Architect checks whether the element has a link to either an operation or an attribute of another element - if such a link exists, and that other element has source code, the code for *that* element displays
- You can also locate the directory containing a source file that has been created in or imported to Enterprise Architect, and edit it or its related files using an *external* editor such as Notepad or Visual Studio; right-click on the element in the Project Browser and select the **Open Source Directory** context menu option (**Ctrl+Alt+Y**)

### Learn more

- [Generate Source Code](#) <sup>[2117]</sup>
- [Languages Supported](#) <sup>[2147]</sup>
- [Compare Editors](#) <sup>[2148]</sup>
- [Code Editor Context Menu](#) <sup>[2152]</sup>
- [Code Editor Toolbar](#) <sup>[2149]</sup>
- [Code Editor Functions](#) <sup>[2157]</sup>
- [Code Editor Key Bindings](#) <sup>[2167]</sup>
- [Options - Code Editors](#) <sup>[2250]</sup>
- [Connect to Element Feature](#) <sup>[1110]</sup>

### Learning Center topics

- (**Alt+F1**) | **Enterprise Architect | Software Engineering | Code Editor**

## 11.7.1 Languages Supported

The Source Code Editors can display code in a wide range of languages, as listed here. For each language, the editor highlights - in colored text - the standard code syntax.

- Ada (.ada, .ads, .adb)
- ActionScript (.as)
- BPEL Document (.bpel)
- C++ (.h, .hh, .hpp, .c, .cpp, .cxx)
- C# (.cs)
- DDL Structured Query Language (.sql)
- Delphi/Pascal (.pas)
- Diff/Patch Files (.diff, .patch)
- Document Type Definition (.dtd)

- DOS Batch Files (.bat)
- DOS Command Scripts (.cmd)
- HTML (.html)
- Interface Definition Language (.idl, .odl)
- Java (.java)
- Javascript (.javascript)
- JScript (.js)
- Modified Backus-Naur Form Grammar (.mbnf)
- PHP (.php, .php4, .inc)
- Python (.py)
- Standard Generalized Markup Language (.sgml)
- SystemC (.sc)
- Visual Basic 6 (.bas)
- VB.NET (.vb)
- VBScript (.vbs)
- Verilog (.v)
- VHDL Hardware Description Language (.vhd)
- Visual Studio Resource Configuration (.rc)
- XML (eXtensible Markup Language) (.xml)

#### Learn more

- [Function Details \(Syntax Highlighting\)](#) <sup>[2157]</sup>
- [Options - Code Editors](#) <sup>[2250]</sup>
- [Language Options](#) <sup>[2262]</sup>
- [Generate BPEL 1.1](#) <sup>[1893]</sup>
- [Generate BPEL 2.0](#) <sup>[1916]</sup>
- [Generate DDL](#) <sup>[2380]</sup>
- [Create Web Style Templates](#) <sup>[2747]</sup>
- [Generate XSD](#) <sup>[2417]</sup>

### 11.7.2 Compare Editors

Enterprise Architect provides four principal code editor variants, available through a number of access paths. The most direct access options are identified in the descriptions below.

The first three code editor variants listed have the same display format, option toolbar, context menu options and internal function keys. They differ in their method of access and display mechanism.

#### Editor Variants

Variant	Function Key/ Menu Option	Description
<b>Source Code</b>	<b>Alt+7</b>	Displays the contents of the source file for a selected Class

Variant	Function Key/ Menu Option	Description
<b>window (Dockable)</b>	<b>Element   Show Source Viewer</b>	(except if the language is C++, when the window displays a tab for the Header file and a tab for the Implementation file).  If you select a different Class, the window <b>changes</b> to show the code for the new Class (unless the first Class calls the second, in which case the window scrolls down to the second Class's code instead).
<b>Source Code View</b>	<b>F12</b> <b>Ctrl+E</b> <b>Class context menu   View Source Code</b>	Displays the code on a tab of the Diagram View; the tab label shows the file name and extension (such as .java); again, for C++, there are two tabs for the Header and Implementation files.  You can display the source code for other Classes on <b>additional</b> tabs, by reselecting the menu option/keys on the next Class.
<b>Internal Editor, External Source Code</b>	<b>Ctrl+Alt+O</b> <b>Tools   Open Source File</b>	Use this option if you intend to edit external code, XML or DDL files (that is, code not imported to or generated in Enterprise Architect).  Displays an external browser, then opens the specific selected code file as a tab of the Diagram View (for C++, <b>not</b> two code files); otherwise this is identical to the <b>F12</b> option.
<b>External Editor, Internal or External Source Code</b>	<b>Ctrl+Alt+Y</b> <b>Class context menu   Open Source Directory</b>	Displays an external file browser, open to the directory containing the selected Class's source files; you can open the files in Notepad, Visual Studio or other tools you might have on your system.

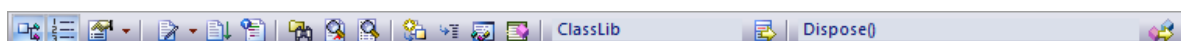
#### Learn more

- [Code Editor Context Menu](#) <sup>[2152]</sup>
- [Code Editor Toolbar](#) <sup>[2149]</sup>
- [Code Editor Functions](#) <sup>[2157]</sup>
- [Code Editor Key Bindings](#) <sup>[2167]</sup>
- [Options - Code Editors](#) <sup>[2250]</sup>





### 11.7.3 Code Editor Toolbar




When you are reviewing the code for a part of your model in the Source Code editor, you can access a wide range of display and editing functions from the editor toolbar.

#### Code Editor Toolbar



Toolbar Options

Icon/Option	Action	See also
<b>Structure Tree</b>	Click on this icon to show or hide the element hierarchy panel (the left panel of the Source Code editor).	
<b>Line Numbers</b>	Click on this icon to show or hide the line numbers against the lines of code.	
<b>Source Code Engineering Properties</b>	<p>Click on the drop-down arrow to display a menu of options to select individual Source Code Engineering pages of the Options dialog, from which you can configure display and behavior options for source code engineering:</p> <ul style="list-style-type: none"> <li>• <b>Syntax Highlighting Options</b></li> <li>• <b>Code Editor Options</b></li> <li>• <b>Code Engineering Options</b></li> <li>• <b>Code Editor Key Bindings</b></li> </ul>	<p><a href="#">Editor Language Properties</a>  <sup>[2251]</sup></p> <p><a href="#">Options - Code Editors</a>  <sup>[2250]</sup></p> <p><a href="#">Source Code Options</a>  <sup>[2247]</sup></p> <p><a href="#">Code Editor Key Bindings</a>  <sup>[2167]</sup></p>
<b>Editor Functions</b>	<p>Click on the drop-down arrow to display a menu providing access to a range of code editing functions:</p> <ul style="list-style-type: none"> <li>• <b>Open Corresponding File (Ctrl+Shift+O)</b> - opens the header or implementation file associated with the currently-open file</li> <li>• <b>Go to Matching Brace (Ctrl+E)</b> - for a selected opening or closing brace, highlights the corresponding closing or opening brace in the pair</li> <li>• <b>Go to Line (Ctrl+G)</b> - displays a dialog on which you select the number of the line to highlight; click on the <b>OK</b> button to move the cursor to that line</li> <li>• <b>Cursor History Previous (Ctrl+-)</b> - the Source Code viewer keeps a history of the previous 50 cursor positions, creating a record when the cursor is moved either more than 10 lines away from its previous position, or in a find-and-replace operation; the menu option moves the cursor to the position in the immediately-previous cursor history record</li> <li>• <b>Cursor History Next (Ctrl+Shift+-)</b> - if you have moved to an earlier cursor position, this option moves the cursor to the position in the immediately-following cursor history record</li> <li>• <b>Find (Ctrl+F)</b> - displays a dialog in which you define a text string and search options to locate that text string in the code</li> <li>• <b>Replace (Ctrl+R)</b> - displays a dialog in which you define a text string and search options to locate that text string in the code and replace it with another text string; the dialog has options to locate and replace each occurrence as you decide, or to replace all occurrences immediately</li> <li>• <b>Record Macro</b> - records your next keystrokes to be saved as a macro</li> </ul>	



Icon/Option	Action	See also
	<ul style="list-style-type: none"> <li>• <b>Stop Recording and Save Macro</b> - stops recording the keystrokes and displays the Save Macro dialog on which you specify a name for the macro</li> <li>• <b>Play Macro</b> - displays the Open Macro dialog from which you select and execute a saved macro, to repeat the saved keystrokes</li> <li>• <b>Toggle Line Comment (Ctrl+Shift+C)</b> - comments out (//) or re-establishes the code for each <b>full line</b> in which text is highlighted</li> <li>• <b>Toggle Stream Comment (Ctrl+Shift+X)</b> - inserts a stream comment (<i>/* */</i>) at the cursor position (comments out only the <b>highlighted characters and lines</b>), or re-establishes the commented text as code</li> <li>• <b>Toggle Whitespace Characters (Ctrl+Shift+W)</b> - shows or hides the spacing characters: --&gt; (tab space) and . (character space)</li> <li>• <b>Toggle EOL Characters (Ctrl+Shift+L)</b> - shows or hides the end-of-line characters: CR (carriage return) and LF (line feed)</li> <li>• <b>Open Containing Folder</b> - opens the file browser at the folder containing the code file; you can open other files in your default external editor for comparison and parallel work</li> </ul>	
<b>Save Source and Resynchronize Class</b>	Click on this icon to save the source code and resynchronize the code and the Class in the model.	
<b>Code Templates</b>	Click on this icon to access the Code Templates Editor, to edit or create code templates for code generation.	<a href="#">Code Templates Editor</a> 
<b>Find in Project Browser</b>	For a selected line of code, click on this icon to highlight the corresponding structure in the Project Browser. If there is more than one possibility the Possible Matches dialog displays, listing the occurrences of the structure from which you can select the required one.	
<b>Search in Files</b>	Click on this icon to search for the selected object name in associated files, and display the results of the search in the File Search window. You can refine and refresh the search by specifying criteria on the Find in Files window toolbar.	<a href="#">Search in Files</a> 
<b>Search in Model</b>	Click on this icon to search for the selected text throughout the model, and display the results of the search in the Model Search window.	<a href="#">Model Search</a> 
<b>Go to Declaration</b>	Click on this icon to locate the <b>declaration</b> of a symbol in the source code.	

Icon/Option	Action	See also
<b>Go to Definition</b>	Click on this icon to locate the <b>definition</b> of a symbol in the source code (applicable to languages such as C++ and Delphi, where symbols are declared and defined in separate files).	
<b>Autocomplete List</b>	Click on this icon to display the autocompletion list of possible values; double-click on a value to select it.	
<b>Parameter Information</b>	When the cursor is between the parentheses of an operation's parameter list, click on this icon to display the operation's signature, highlighting the current parameter.	
<b>Find Current Class in Project Browser</b>	Click on this icon to display the name of the currently-selected Class in the code, and highlight that name in the Project Browser; if there is more than one possibility the Possible Matches dialog displays, listing the occurrences of the Class from which you can select the required one.	
<b>Find Member</b>	Click on this icon to display the name of the currently-selected attribute or method in the code, and highlight that name in the Project Browser; if there is more than one possibility the Possible Matches dialog displays, listing the occurrences of the feature from which you can select the required one.	

### Notes

- The **Record Macro** option disables Intelli-sense while the macro is being recorded
- You can **assign key strokes** to execute the macro, instead of using the toolbar drop-down and Open Macro dialog

### Learn more

- [Intelli-sense](#)  <sup>[2160]</sup>
- [Editor Language Properties](#)  <sup>[2251]</sup>

## 11.7.4 Code Editor Context Menu

When working on a file with a code editor, you can perform a number of code search and editing operations to review the contents of the file. These options are available through the editor context menu, and can vary depending on which code editor you are using.

**Access** **Right-click on the code text string you are working on**



Options

Option	Action	See also
<b>Go to Declaration</b>	Locate and highlight the declaration of a symbol in the source code.	
<b>Go to Definition</b>	Locate and highlight the definition of a symbol in the source code (applicable to languages such as C++ and Delphi, where symbols are declared and defined in separate places).	
<b>Search for '&lt;string&gt;'</b>	Display a submenu providing options to locate the selected text string in a range of locations.	
	<b>Find in Project Browser</b>	Highlight the object containing the selected text in the Project Browser.
	<b>Search in Open Files</b>	Search for the selected text string in associated <b>open</b> files and display the results of the search in the Find in Files window. You can refine and refresh the search by specifying criteria on the Find in Files window toolbar.  <a href="#">Search In Files</a> [2164]
	<b>Search in Files</b>	Search for the selected text string in all associated files ( <b>closed or open</b> ), and display the results of the search in the Find in Files window. You can refine and refresh the search by specifying criteria on the Find in Files window toolbar.  Shortcut key: <b>F12</b>  <a href="#">Model Search</a> [700] <a href="#">Pre-Defined Searches</a> [700]
	<b>Search in Model</b>	Perform an <i>Element Name</i> search in the Model Search facility, and display the results on the Model Search tab.  <a href="#">Script Editor</a> [2798]
	<b>Search in Scripts</b>	(Available while working in the Script Editor.)  Open the Find in Files window, set the <b>Search Path</b> field to <b>Search in Scripts</b> and the <b>Search Text</b> field to the selected text, then search all scripts for the text string and display the results of the search.  You can refine and refresh the search by specifying criteria on the Find in Files window toolbar.  <a href="#">Search In Files</a> [2164]
	<b>EA User Guide</b>	Display the description of the code item in the <i>Enterprise Architect User Guide</i> .

Option	Action		See also
	<b>Google</b>	Display the results of a Google search on the text.	
	<b>MSDN</b>	Display the results of a search on the text in the Microsoft Developer Network (MSDN).	
	<b>Sun Java SE</b>	Display the results of a search on the text in the Sun Microsystems 'Sun Search' facility.	
	<b>Wikipedia</b>	Display any entry on the object on the Wikipedia web site.	
	<b>Koders</b>	Display the results of a search for the text string on <i>Koders.com</i> .	
<b>Search Intelli-sense   &lt;list of query names&gt;</b>	Perform a search on the specified string in any Intelli-sense libraries listed in the active Execution Analyzer script, and display the results in the Find in Files window, Intelli-sense Search tab.  Shortcut key: <b>Shift + F12</b>		<a href="#">Search Intelli-sense</a> <sup>[2166]</sup>
<b>Set Debugger to Line</b>	(If the debugger is executing and has reached a breakpoint.) Move the execution point to the current line. Check that you do not skip over any code or declarations that affect the next section of code being debugged.		
<b>Display Variable</b>	(If the debugger is executing.) Open the Locals window and highlight the local variable for the current point in the code.		<a href="#">View the Local Variables</a> <sup>[2234]</sup>
<b>Show in String Viewer</b>	Display the full contents of a variable string in the String Viewer.		<a href="#">View Content Of Long Strings</a> <sup>[2235]</sup>
<b>Create Use Case for '&lt;string&gt;'</b>	Display the Create Use Case For Method dialog, through which you create a Use Case for the method containing the text string.		<a href="#">Create Use Case For Method</a> <sup>[2155]</sup>
<b>Breakpoint</b>	Display a submenu of options for creating a recording marker on the selected line of code. The recording markers you can add include: <ul style="list-style-type: none"> <li>• Breakpoint</li> <li>• Start Recording Marker</li> <li>• End Recording Marker</li> <li>• Stack Auto Capture Marker</li> <li>• Method Auto Record Marker</li> <li>• Tracepoint</li> </ul>		<a href="#">Set Record Markers</a> <sup>[2537]</sup>

Option	Action	See also
<b>Testpoints</b>	Display options to add a new <b>Testpoint</b> , show the Testpoints Manager ( <b>Testpoints window</b> ) or edit an existing Testpoint if one or more are already defined at the selected location.  (The sub-options depend on the type of code file you are reviewing.)	<a href="#">Testpoint Editor</a> <sup>[2564]</sup>  <a href="#">The Testpoints Window</a> <sup>[2578]</sup>
<b>Open (Close) IME</b>	Open (or close) the Input Method Editor, so that you can enter text in a selected foreign language script, such as Japanese. You set the keyboard language using the <b>Windows Control Panel - Regional and Language Options</b> facility.	
<b>Line Numbers</b>	(Script Editor only.) Show or hide the code line numbers on the left hand side of the editor screen.	
<b>Undo</b> <b>Cut</b> <b>Copy</b> <b>Paste</b> <b>Delete</b> <b>Select All</b>	These six options provide simple functions for editing the code.	

### Notes

- The options in the lower half of the **Search for '<string>'** submenu (after **Search in Scripts**) are configurable; you can add new search tools or remove existing ones by editing the *searchProviders.xml* file in the *Sparx Systems > EA > Config* folder - this file is in OpenSearch description document format

### Learn more

- [OpenSearch Description Document Format](#) (Online Resource)
- [Editing Source Code](#)<sup>[2146]</sup>
- [Compare Editors](#)<sup>[2148]</sup>

#### 11.7.4.1 Create Use Case for Method

Using the code editor context menu, you can create a Use Case element for a method that you select from the code. You can also:

- Link the Use Case directly to the method
- Add the parent Class to a diagram (if it is not already in the selected diagram) and/or add the Use Case element to the diagram
- Block from display any attributes or methods that are not also the targets of feature links

Create a Use Case for a method, through the code editor

Step	Action	See also
1	(If you want to depict the Use Case and its link to the method <i>in a diagram</i> ) click on the diagram name in the Project Browser.	
2	In the code editor, right click on either the method name or any part of the method body, and select the <b>Create Method for &lt;methodname&gt;</b> context menu option.  The Create Use Case for Method dialog displays.	<a href="#">Code Editor Context Menu</a> [2152]
3	The basic function of this dialog is to create a Use Case for the selected method: <ul style="list-style-type: none"> <li>If this is all that is required, click on the <b>OK</b> button; the Use Case element is created in the Project Browser, in the same package as the parent Class for the method, and with the same name as the method</li> <li>If you intend to make the relationship tangible, continue with the procedure</li> </ul>	
4	To create a Trace connector linking the Use Case to the method, select the <b>Link Use Case to Method</b> check box.	
5	To add the method's parent Class to the diagram, if it is not already there, select the <b>Add Class to Diagram</b> check box.	
6	To add the newly-created Use Case to the diagram, select the <b>Add Use Case to Diagram</b> check box; this would now show the Use Case, Class and Trace connector on the diagram.	
7	To only show the features (attributes and methods) of the parent Class that are the targets of 'link to feature' relationships, select the <b>Display only linked features in Class</b> checkbox.  The Class might contain any number of attributes and methods, but those without a 'link to feature' relationship are hidden.	<a href="#">Connect to Element Feature</a> [1110]
8	Click on the <b>OK</b> button to create and depict the Use Case and relationship; if you selected all options, the diagram now contains linked elements resembling the following: <div data-bbox="319 1765 1117 2004" data-label="Diagram"> <pre> classDiagram     usecase memoryCancel     class ClassLib {         memoryCancel(bool*)     }     memoryCancel --&gt; ClassLib : «trace»   </pre> </div>	

Step	Action	See also

### 11.7.5 Code Editor Functions

The common Code Editor provides a variety of functions to assist with the code editing process, including:

- Syntax Highlighting
- Bookmarks
- Cursor History
- Brace Matching
- Automatic Indentation
- Commenting Selections
- Scope Guides
- Zooming
- Line Selection
- intelli-sense
- Find and Replace
- Find in Files

A range of these functions is available through keyboard key combinations and/or context menu options.

You can customize several of the Code Editor features by setting properties in the Code Editor configuration files; for example, by default the line containing the cursor is always highlighted, but you can turn the highlighting off.

#### Learn more

- [Function Details](#) <sup>[2157]</sup>
- [Intelli-sense](#) <sup>[2160]</sup>
- [Find and Replace](#) <sup>[2161]</sup>
- [Search in Files](#) <sup>[2164]</sup>
- [Code Editor Key Bindings](#) <sup>[2167]</sup>
- [Code Editor Context Menu](#) <sup>[2152]</sup>

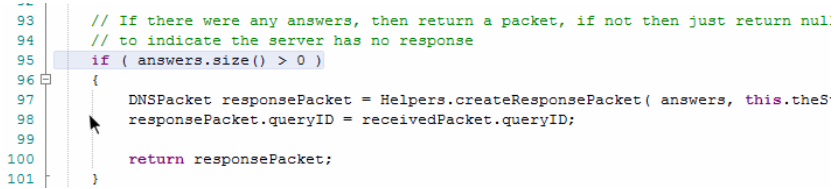
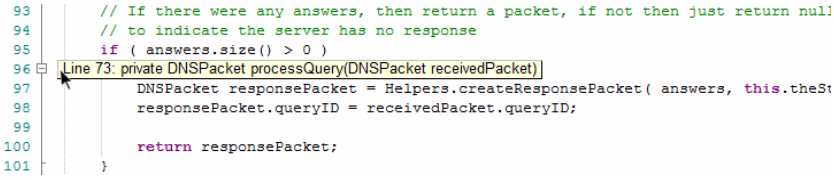
#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Software Engineering | Code Editor | Locate in Source**
- (Alt+F1) | **Enterprise Architect | Software Engineering | Code Editor | Using Intelli-sense**

#### 11.7.5.1 Function Details

Topic	Detail	See also
<b>Syntax Highlighting</b>	The Code Editor highlights - in colored text - the standard code syntax of all language file formats supported by Enterprise Architect	<a href="#">Languages Supported</a> <sup>[2147]</sup>

Topic	Detail	See also
	<pre> 1 #pragma once 2 #include "afxwin.h" 3 #include "afxcmn.h" 4 5 6 // CToolBox dialog 7 8 class CToolBox : public CDialog 9 { 10     DECLARE_DYNAMIC(CToolBox) 11     CRect m_rect; 12     int m_offset; </pre> <p>You can define how the Code Editor implements syntax highlighting for each language, through the Code Editors page of the Options dialog.</p>	<a href="#">Options - Code Editors</a> <a href="#">Editor Language Properties</a> <a href="#">Language Options</a>
<b>Bookmarks</b>	<p>Bookmarks denote a line of interest in the document; you can toggle them on and off for a particular line by pressing <b>Ctrl+F2</b>.</p> <p>Additionally, you can press <b>F2</b> and <b>Shift+F2</b> to navigate to the next or previous bookmark in the document.</p> <p>To clear all bookmarks in the code file, press <b>Ctrl+Shift+F2</b>.</p>	
<b>Cursor History</b>	<p>The Code Editor Control keeps a history of the previous 50 cursor positions; an entry in the history list is created when:</p> <ul style="list-style-type: none"> <li>The cursor is moved more than 10 lines from its previous position</li> <li>The cursor is moved in a <i>find/replace</i> operation</li> </ul> <p>You can navigate to an earlier point in the cursor history by pressing <b>Ctrl+-</b>, and to a later point by pressing <b>Ctrl+Shift+-</b>.</p>	
<b>Brace Matching</b>	<p>When you place the cursor over a brace or bracket, the Code Editor highlights its corresponding partner; you can then navigate to the matching brace by pressing <b>Ctrl+E</b>.</p> <pre> 28     function ProtectedFunctionTest: boolean; 29     procedure ProtectedProcedureTest(a: WideString); </pre>	
<b>Automatic Indentation</b>	<p>For each supported language, the Code Editor adjusts the indentation of a new line according to the presence of control statements or scope block tokens in the lines leading up to the cursor position.</p> <pre> 358 { 359     for(size_t t = 0; t &lt; Stations.size(); t++) 360     { 361         if(Stations[t]-&gt;Location == loc) 362             return Stations[t]; 363     } 364     return NULL; 365 } </pre>	

Topic	Detail	See also
	<p>The levels of indent are indicated by pale horizontal lines.</p> <p>You can also manually indent selected lines and blocks of code by pressing <b>Tab</b>; to unindent the selected code, press <b>Shift+Tab</b>.</p>	
<b>Commenting Selections</b>	<p>For languages that support comments, the Code Editor can comment entire selections of code.</p> <p>The Code Editor recognizes two types of commenting:</p> <ul style="list-style-type: none"> <li>Line Commenting - entire lines are commented from the start (for example:  <code>// This is a comment</code>)</li> <li>Stream Commenting - sections of a line are commented from a specified start point to a specified end point (for example:  <code>/* This is a comment */</code>)</li> </ul> <p>You can toggle comments on the current line or selection by pressing:</p> <ul style="list-style-type: none"> <li><b>Ctrl+Shift+C</b> for line comments, or</li> <li><b>Ctrl+Shift+X</b> for stream comments</li> </ul>	
<b>Scope Guides</b>	<p>If the cursor is placed over an indentation marker, the Code Editor performs a 'look back' to find the line that started the scope at that indentation level; if the line is found and is currently on screen, it is highlighted in light blue.</p>  <pre> 93 // If there were any answers, then return a packet, if not then just return null 94 // to indicate the server has no response 95 if ( answers.size() &gt; 0 ) 96 { 97     DNSPacket responsePacket = Helpers.createResponsePacket( answers, this.theS 98     responsePacket.queryID = receivedPacket.queryID; 99 100     return responsePacket; 101 } </pre> <p>Alternatively if the line is off screen, a calltip is displayed advising of the line number and contents:</p>  <pre> 93 // If there were any answers, then return a packet, if not then just return null 94 // to indicate the server has no response 95 if ( answers.size() &gt; 0 ) 96 Line 73: private DNSPacket processQuery(DNSPacket receivedPacket) 97     DNSPacket responsePacket = Helpers.createResponsePacket( answers, this.theS 98     responsePacket.queryID = receivedPacket.queryID; 99 100     return responsePacket; 101 } </pre>	
<b>Zooming</b>	<p>You can zoom into and out of the contents of the Code Editor using:</p> <ul style="list-style-type: none"> <li><b>Ctrl + keypad +</b> and</li> <li><b>Ctrl + keypad -</b></li> </ul> <p>Zoom can be restored to 100% using <b>Ctrl + keypad /</b>.</p>	

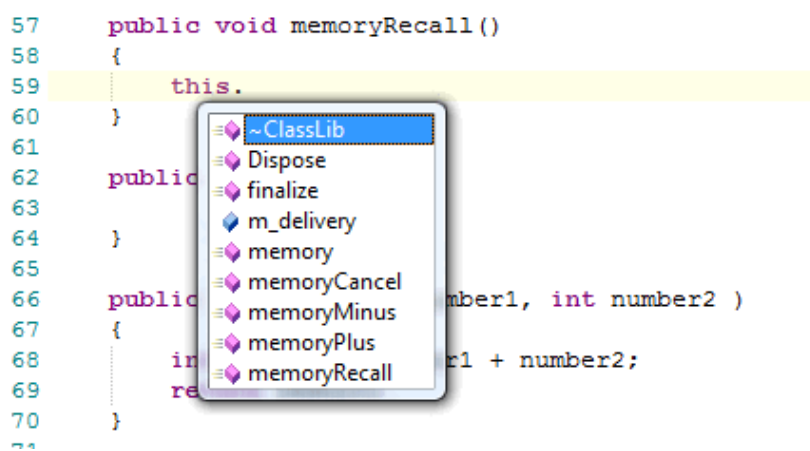
Topic	Detail	See also
<b>Line Selection</b>	<p>If you want to move the cursor to a specific line of code, press <b>Ctrl+G</b> and, in response to the prompt, type in the line number.</p> <p>Press the <b>OK</b> button; the editor displays the specified line of code with the cursor at the left.</p>	

### 11.7.5.2 Intelli-sense

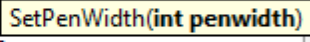
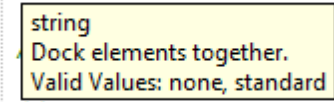
Intelli-sense is a feature that provides choices of code items and values as you type. Not all code editors use Intelli-sense; for example, Intelli-sense is disabled while you record a macro in the Source Code Viewer.

Intelli-sense provides you with context-based assistance through autocompletion lists, calltips and mouseover information.

#### Topics

Topic	Detail	See also
<b>Autocompletion List</b>	<p>An autocompletion list provides a list of possible completions for the current text; the list is automatically invoked when you enter an accessor token (such as a period or pointer accessor) after an object or type that contains members.</p>  <p>You can also invoke the autocompletion list manually by pressing (<b>Ctrl+Space</b>); the Code Editor then searches for matches for the word leading up to the invocation point.</p> <p>Select an item from the list and press (<b>Enter</b>) or (<b>Tab</b>) to insert the item into the code; to dismiss the autocompletion list, press (<b>Escape</b>).</p>	
<b>Calltips</b>	<p>Calltips display the current method's signature when you type the parameter list token (for example, opening parenthesis); if the method is overloaded, the calltip displays arrows that you can use to navigate through the different method signatures</p>	



Topic	Detail	See also
	<pre> 20      //PostDraw Adornments 21      //Stereotyped Static Adornments 22      //Add Stakeholder's STAKE 23      setpenwidth( 24      // Add a the 25      startpath( 26      moveto(25,37); 27      lineto(25,52); 28      endpath(); 29      strokepath(); 30      //Add tip </pre> 	
<b>Mouseover Information</b>	<p>You can display supporting documentation for code elements (for example, attributes and methods) by hovering the cursor over the element in question.</p> <pre> 11      dockable = "none"; 12      string 13      Dock elements together. 14      Valid Values: none, standard 15      //PreDraw Derived Attribute 1 </pre> 	

#### Learn more

- [Code Editor Functions](#) <sup>[2157]</sup>
- [Editing Source Code](#) <sup>[2146]</sup>

### 11.7.5.3 Find and Replace

Each of Enterprise Architect's code editors facilitates searching for and replacing terms in the editor, through the Find and Replace dialog.

#### Access

Highlight the required text string and press:

- **(Ctrl+F)** for the *find* controls only, or
- **(Ctrl+R)** for both *find* and *replace* controls

In each instance, the **Find what** field is populated with the text currently selected in the editor. If no text is selected in the editor, the **Find what** field is populated with the word at the current cursor position. If no word exists at the current cursor position, the last searched-for term is used.

#### Basic Operations

##### Commands

Command	Action	See also
<b>Find Next</b>	Locate and highlight the next instance (relative to the current cursor position) of the text specified in the <b>Find what</b> field.	
<b>Replace</b>	Replace the current instance of the text specified in the <b>Find what</b> field with the text specified in the <b>Replace with</b> field, and then locate and highlight the next instance (relative to the current cursor position) of the text specified in the <b>Find what</b> field.	
<b>Replace All</b>	Automatically replace all instances of the text specified in the <b>Find what</b> field with the text specified in the <b>Replace with</b> field.	

### Options

Option	Action	See also
<b>Match Case</b>	Specify that the case of each character in the text string in the <b>Find what</b> field is significant when searching for matches in the code.	
<b>Match whole word</b>	Specify that the text string in the <b>Find what</b> field is a complete word and should not be matched with instances of the text that form part of a longer string.  For example, searches for ARE should not match those letters in instances of the words AREA or ARENA.	
<b>Search up</b>	Perform the search from the current cursor position up to the start of the file, rather than in the default direction of current cursor position to end of file.	
<b>Use Regular Expressions</b>	Evaluate specific character sequences in the <b>Find what</b> and <b>Replace with</b> fields as <i>Regular Expressions</i> (see below).	

### Topics

Topic	Detail	See also
<b>Regular Expressions</b>	A Regular Expression is a formal definition of a search pattern, which can be used to match specific characters, words or patterns of characters.  For the sake of simplicity, the Code Editor's 'find and replace' mechanism supports only a subset of the standard Regular Expression grammar.  Text in the <b>Find what</b> and <b>Replace with</b> fields is only interpreted as a Regular Expression if the <b>Use Regular Expressions</b> checkbox is selected	

Topic	Detail	See also																		
	in the Find and Replace dialog.																			
<b>Meta sequences</b>	<p>If the <b>Use Regular Expressions</b> checkbox is selected, most characters in the <b>Find what</b> field are treated as literals (that is, they match only themselves).</p> <p>The exceptions are called <i>metasequences</i>; each <i>metasequence</i> recognized in the Code Editor <b>Find and Replace</b> dialog is described in the following table:</p> <table><tr><th>Metasequence</th><th>Description</th></tr><tr><td>.</td><td>Represents any single character. For example: <b>.at</b> is matched to <b>cat</b>, <b>hat</b>, <b>bat</b> and <b>rat</b>.</td></tr><tr><td>\&lt;</td><td>Indicates that the text is the start of a word. For example: <b>\&lt;cat</b> is matched to <b>catastrophe</b> and <b>concatenate</b>.</td></tr><tr><td>\&gt;</td><td>Indicates that the text is the end of a word. For example: <b>hat\&gt;</b> is matched to <b>that</b> and <b>chat</b>, but not <b>hat</b>.</td></tr><tr><td>( ... )</td><td>Indicates alternative single characters that can be match can be specific - ( <b>chr</b> ) - or in an alphabetical or numerical range - ( <b>a-m</b> ). For example: ( <b>hc</b> ) at is matched to <b>hat</b> and <b>cat</b> but not <b>bat</b>. For example: ( <b>a-m</b> ) Class is matched to any name in the <b>mClass</b>.</td></tr><tr><td>( ^... )</td><td>Indicates alternative single characters that should be excluded from the match. The characters can be specific - ( <b>^chr</b> ) - or in an alphabetical or numerical range - ( <b>^a-m</b> ). For example: ( <b>^hc</b> ) at is matched to <b>rat</b> and <b>bat</b> but <b>hat</b> is excluded. For example: ( <b>^a-m</b> ) Class is matched to any name in the <b>zClass</b> but <b>aClass</b> to <b>mClass</b> are excluded.</td></tr><tr><td>^</td><td>Matches the start of a line.</td></tr><tr><td>\$</td><td>Matches the end of a line.</td></tr><tr><td>*</td><td>Matches the preceding character (or character set) 0 or more times. For example: <b>ba*t</b> is matched to <b>bt</b>, <b>bat</b>, <b>baat</b>, <b>baaat</b> and <b>baaaat</b>.</td></tr></table>	Metasequence	Description	.	Represents any single character. For example: <b>.at</b> is matched to <b>cat</b> , <b>hat</b> , <b>bat</b> and <b>rat</b> .	\<	Indicates that the text is the start of a word. For example: <b>\&lt;cat</b> is matched to <b>catastrophe</b> and <b>concatenate</b> .	\>	Indicates that the text is the end of a word. For example: <b>hat\&gt;</b> is matched to <b>that</b> and <b>chat</b> , but not <b>hat</b> .	( ... )	Indicates alternative single characters that can be match can be specific - ( <b>chr</b> ) - or in an alphabetical or numerical range - ( <b>a-m</b> ). For example: ( <b>hc</b> ) at is matched to <b>hat</b> and <b>cat</b> but not <b>bat</b> . For example: ( <b>a-m</b> ) Class is matched to any name in the <b>mClass</b> .	( ^... )	Indicates alternative single characters that should be excluded from the match. The characters can be specific - ( <b>^chr</b> ) - or in an alphabetical or numerical range - ( <b>^a-m</b> ). For example: ( <b>^hc</b> ) at is matched to <b>rat</b> and <b>bat</b> but <b>hat</b> is excluded. For example: ( <b>^a-m</b> ) Class is matched to any name in the <b>zClass</b> but <b>aClass</b> to <b>mClass</b> are excluded.	^	Matches the start of a line.	\$	Matches the end of a line.	*	Matches the preceding character (or character set) 0 or more times. For example: <b>ba*t</b> is matched to <b>bt</b> , <b>bat</b> , <b>baat</b> , <b>baaat</b> and <b>baaaat</b> .	
Metasequence	Description																			
.	Represents any single character. For example: <b>.at</b> is matched to <b>cat</b> , <b>hat</b> , <b>bat</b> and <b>rat</b> .																			
\<	Indicates that the text is the start of a word. For example: <b>\&lt;cat</b> is matched to <b>catastrophe</b> and <b>concatenate</b> .																			
\>	Indicates that the text is the end of a word. For example: <b>hat\&gt;</b> is matched to <b>that</b> and <b>chat</b> , but not <b>hat</b> .																			
( ... )	Indicates alternative single characters that can be match can be specific - ( <b>chr</b> ) - or in an alphabetical or numerical range - ( <b>a-m</b> ). For example: ( <b>hc</b> ) at is matched to <b>hat</b> and <b>cat</b> but not <b>bat</b> . For example: ( <b>a-m</b> ) Class is matched to any name in the <b>mClass</b> .																			
( ^... )	Indicates alternative single characters that should be excluded from the match. The characters can be specific - ( <b>^chr</b> ) - or in an alphabetical or numerical range - ( <b>^a-m</b> ). For example: ( <b>^hc</b> ) at is matched to <b>rat</b> and <b>bat</b> but <b>hat</b> is excluded. For example: ( <b>^a-m</b> ) Class is matched to any name in the <b>zClass</b> but <b>aClass</b> to <b>mClass</b> are excluded.																			
^	Matches the start of a line.																			
\$	Matches the end of a line.																			
*	Matches the preceding character (or character set) 0 or more times. For example: <b>ba*t</b> is matched to <b>bt</b> , <b>bat</b> , <b>baat</b> , <b>baaat</b> and <b>baaaat</b> .																			

Topic	Detail	See also								
	<table><tr><th>Metasequence</th><th>Description</th></tr><tr><td></td><td>For example: <b>b( ea ) *t</b> is matched to <b>bt, bet, bat, beat, be</b></td></tr><tr><td><b>+</b></td><td>Matches the preceding character (or character set) 1 or more times. For example: <b>ba+t</b> is matched to <b>bat, baat</b> and <b>baaat</b> but not <b>ba</b>. For example: <b>b( ea ) +t</b> is matched to <b>bet, bat, beat, beet</b> but not <b>b</b>.</td></tr></table> <p>If a single character metasequence is preceded by a backslash (<b>\</b>) it is treated as a literal character.<b>c\ ( at\ )</b> matches <b>c( at )</b> as the brackets are treated literally.</p> <p>When the <b>Use Regular Expressions</b> checkbox is selected, a metasequence helper menu is available to the right of both of the <b>Find what</b> and <b>Replace with</b> fields; selecting a metasequence from this menu inserts the metasequence into the field, replacing or wrapping the currently selected text as appropriate.</p>	Metasequence	Description		For example: <b>b( ea ) *t</b> is matched to <b>bt, bet, bat, beat, be</b>	<b>+</b>	Matches the preceding character (or character set) 1 or more times. For example: <b>ba+t</b> is matched to <b>bat, baat</b> and <b>baaat</b> but not <b>ba</b> . For example: <b>b( ea ) +t</b> is matched to <b>bet, bat, beat, beet</b> but not <b>b</b> .			
Metasequence	Description									
	For example: <b>b( ea ) *t</b> is matched to <b>bt, bet, bat, beat, be</b>									
<b>+</b>	Matches the preceding character (or character set) 1 or more times. For example: <b>ba+t</b> is matched to <b>bat, baat</b> and <b>baaat</b> but not <b>ba</b> . For example: <b>b( ea ) +t</b> is matched to <b>bet, bat, beat, beet</b> but not <b>b</b> .									
<b>Tagged Regions</b>	<p>When 'find and replacing' with Regular Expressions, up to nine sections of the original term can be substituted into the replacement term.</p> <p>The metasequences <b>\(</b> and <b>\)</b> denote the start and the end of a <i>tagged region</i>; the section of the matched text that falls within the tagged region can be included in the replacement text with the metasequence <b>\n</b> (where <i>n</i> is the tagged region number between 1 and 9).</p> <p>For example:</p> <table><tr><td><b>Find what:</b></td><td><b>\( ( A- Z a- z ) +\ )' s t h i n g s</b></td></tr><tr><td><b>Replace with</b></td><td><b>i t e m s t h a t b e l o n g t o \1</b></td></tr><tr><td>Original text:</td><td>These are all Michael's things.</td></tr><tr><td>Replaced text:</td><td>These are all items that belong to Michael</td></tr></table>	<b>Find what:</b>	<b>\( ( A- Z a- z ) +\ )' s t h i n g s</b>	<b>Replace with</b>	<b>i t e m s t h a t b e l o n g t o \1</b>	Original text:	These are all Michael's things.	Replaced text:	These are all items that belong to Michael	
<b>Find what:</b>	<b>\( ( A- Z a- z ) +\ )' s t h i n g s</b>									
<b>Replace with</b>	<b>i t e m s t h a t b e l o n g t o \1</b>									
Original text:	These are all Michael's things.									
Replaced text:	These are all items that belong to Michael									

#### 11.7.5.4 Search in Files

File Text Searches are provided by the Search window and from within the Code Editors, to search files for data names and structures.

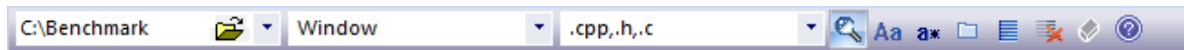
You can use the Search window to search for text in code files and scripts; you can select to display the results of the search in one of two formats:

- List View - each result line consists of the file path and line number, followed by the line text; multiple lines from one file are listed as separate entries
- Tree View - each result line consists of the file path that matches the search criteria, and the number of lines matching the search text within that file; you can expand the entry to show the line number and text of each line

**Access** **Edit | Find in Files** (Ctrl+Shift+Alt+F) > **File Search**

### Search Toolbar

You can use the toolbar options in the Search window to control the search operation. The state of all buttons persists over time to always reflect your previous search criteria.



### Options

Field/Option	Action	See also
<b>Search Path</b> list box	<p>Specify the folder to search.</p> <p>You can type the path to search directly into the text box, or click on the folder icon to browse for the path.</p> <p>Any paths you enter are automatically saved in the drop-down list, up to a maximum of ten; paths added after that overwrite the oldest path in the list.</p> <p>A fixed option in the drop-down list is <b>Search in Scripts</b>, which sets the search to operate on all local and user-defined scripts in the Scripts tab of the Scripting window; this option disables the <b>Search File Types</b> list box.</p>	
<b>Search Text</b> list box	<p>Specify the text to look for.</p> <p>You can type the text directly into the text box or click on the drop-down arrow to select from a previous entry in the list. The search text you enter is automatically saved in the list when you click on the <b>Search</b> button.</p> <p>The list box saves up to ten search queries. Search queries added after that overwrite the oldest query in the list.</p>	
<b>Search File Types</b> list box	<p>Limit the search to specific types of files. You can select multiple file types in a string, separated by either a comma or a semi-colon as shown in the image above.</p>	
<b>Search</b> button	<p>Begin the search.</p> <p>During the course of the search all other buttons in the toolbar are disabled. You can cancel the search at any time by clicking on the <b>Search</b> button again.</p> <p>If you switch any of the toggle buttons below, you must run the search again to change the output.</p>	
<b>Case Sensitivity</b> button	<p>Toggle the case sensitivity of the search. The tooltip message identifies the current status of the button.</p>	

Field/Option	Action	See also
<b>Word Match</b> button	Toggle between searching for any match and searching for only those matches that form an entire word. The tooltip message identifies the current status of the button.	
<b>SubFolders</b> button	Toggle between limiting the search to a single path and including all subfolders under that path. The tooltip message identifies the current status of the button.	
<b>Result View</b> button	Select the presentation format of the search results - List View or Tree View format.	
<b>Clear Results</b> button	Clear the results.	
<b>Clear Search Criteria</b> button	Remove all the entries in the <b>Search Path</b> , <b>Search Text</b> and <b>Search File Types</b> list boxes, if required.	
<b>Help</b> button	Display this Help topic.	

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect** | **Software Engineering** | **Code Editor** | **Search for Options**

#### 11.7.5.4.1 Search Intelli-sense

When editing a code file with the **code editor**, you can perform a search for a text string in any **Intelli-sense libraries** listed in the active Execution Analyzer script.

**Access**   **Edit** | **Find in Files** (Ctrl+Shift+Alt+F) > **Intelli-sense Search**

#### Intelli-sense Libraries

Intelli-sense libraries are information databases that can be used to perform searches and queries on a code base. The databases are created from source code directories using a specific language such as C++, Java or C#, and are available using shortcut keys and context menus in Enterprise Architect code editors.

The libraries are created, updated, removed or added in the **Analyzer Script Editor**. A typical scenario for using this feature would be create libraries for your source code and that of any libraries you are referencing. The libraries for your source can then be updated frequently with code changes, while the external libraries that remain mostly static will not get updated often. Libraries can be searched in a similar way to the File Search tool, but with the following differences:

- Multiple directories can be searched at once by having multiple libraries
- A query can be run in a fraction of the time required for a File Search
- All files are indexed based on equivalent UML constructs, allowing intelligent searches producing

meaningful results in a modeling setting

### **The Intelli-sense Search Control**

The Search Control is located as a tab of the Find in Files window. From this control you view the results of Intelli-sense searches initiated from within the code editor. This control can be used to run, create and maintain queries. Each query is named and the names of all queries become available as sub-menu items in the **Search Intelli-sense** context menu displayed in the code editors.

Using the Intelli-sense Search tab toolbar you can run, create and maintain Intelli-sense queries. Queries are saved as either **Personal** or **Model** queries. Model queries are available to all users of the model; currently, a set of common Model queries are supplied by Sparx Systems and it is not expected that you will have to create others. All queries work on any Intelli-sense **library** created by users.

To view a query, select it from the list and use the gray arrow button at the far right of the toolbar to open the query editor.

### **11.7.6 Code Editor Key Bindings**

Key	Description
<b>Ctrl+G</b>	Move cursor to a specified line
↓	Move cursor down one line
<b>Shift+↓</b>	Extend selection down one line
<b>Ctrl+↓</b>	Scroll down one line
<b>Alt+Shift+↓</b>	Extend rectangular selection down one line
↑	Move cursor up one line
<b>Shift+↑</b>	Extend selection up one line
<b>Ctrl+↑</b>	Scroll up one line
<b>Alt+Shift+↑</b>	Extend rectangular selection up one line
<b>Ctrl+ (</b>	Move cursor up one paragraph
<b>Ctrl+Shift+ (</b>	Extend selection up one paragraph
<b>Ctrl+ )</b>	Move cursor down one paragraph
<b>Ctrl+Shift+ )</b>	Extend selection down one paragraph
←	Move cursor left one character
<b>Shift+←</b>	Extend selection left one character
<b>Ctrl+←</b>	Move cursor left one word
<b>Ctrl+Shift+←</b>	Extend selection left one word

Key	Description
<b>Alt+Shift+←</b>	Extend rectangular selection left one character
<b>→</b>	Move cursor right one character.
<b>Shift+→</b>	Extend selection right one character
<b>Ctrl+→</b>	Move cursor right one word
<b>Ctrl+Shift+→</b>	Extend selection right one word
<b>Alt+Shift+→</b>	Extend rectangular selection right one character
<b>Ctrl+/<b></b></b>	Move cursor left one word part
<b>Ctrl+Shift+/<b></b></b>	Extend selection left one word part
<b>Ctrl+\<b></b></b>	Move cursor right one word part
<b>Ctrl+Shift+\<b></b></b>	Extend selection right one word part
<b>Home</b>	Move cursor to the start of the current line
<b>Shift+Home</b>	Extend selection to the start of the current line
<b>Ctrl+Home</b>	Move cursor to the start of the document
<b>Ctrl+Shift+Home</b>	Extend selection to the start of the document
<b>Alt+Home</b>	Move cursor to the absolute start of the line
<b>Alt+Shift+Home</b>	Extend rectangular selection to the start of the line
<b>End</b>	Move cursor to the end of the current line
<b>Shift+End</b>	Extend selection to the end of the current line
<b>Ctrl+End</b>	Move cursor to the end of the document
<b>Ctrl+Shift+End</b>	Extend selection to the end of the document
<b>Alt+End</b>	Move cursor to the absolute end of the line
<b>Alt+Shift+End</b>	Extend rectangular selection to the end of the line
<b>Page Up</b>	Move cursor up a page
<b>Shift+Page Up</b>	Extend selection up a page
<b>Alt+Shift+Page Up</b>	Extend rectangular selection up a page
<b>Page Down</b>	Move cursor down a page





Key	Description
<b>Shift+Page Down</b>	Extend selection down a page
<b>Alt+Shift+Page Down</b>	Extend rectangular selection down a page
<b>Delete</b>	Delete character to the right of the cursor
<b>Shift+Delete</b>	Cut selection
<b>Ctrl+Delete</b>	Delete word to the right of the cursor
<b>Ctrl+Shift+Delete</b>	Delete until the end of the line
<b>Insert</b>	Toggle overtype
<b>Shift+Insert</b>	Paste
<b>Ctrl+Insert</b>	Copy selection
<b>Backspace</b>	Delete character to the left of the cursor
<b>Shift+Backspace</b>	Delete character to the left of the cursor
<b>Ctrl+Backspace</b>	Delete word to the left of the cursor
<b>Ctrl+Shift+Backspace</b>	Delete from the start of the line to the cursor
<b>Alt+Backspace</b>	Undo delete
<b>Tab</b>	Indent cursor one tab
<b>Ctrl+Shift+I</b>	Indent cursor one tab
<b>Shift+Tab</b>	Unindent cursor one tab
<b>Ctrl+keypad( + )</b>	Zoom in
<b>Ctrl+keypad( - )</b>	Zoom out
<b>Ctrl+keypad( / )</b>	Restore Zoom
<b>Ctrl+Z</b>	Undo
<b>Ctrl+Y</b>	Redo
<b>Ctrl+X</b>	Cut selection
<b>Ctrl+C</b>	Copy selection
<b>Ctrl+V</b>	Paste

Key	Description
<b>Ctrl+L</b>	Cut line
<b>Ctrl+Shift+L</b>	Delete line
<b>Ctrl+T</b>	Transpose line
<b>Ctrl+Shift+T</b>	Copy line
<b>Ctrl+A</b>	Select entire document
<b>Ctrl+D</b>	Duplicate selection
<b>Ctrl+U</b>	Convert selection to lowercase
<b>Ctrl+Shift+U</b>	Convert selection to uppercase
<b>Ctrl+E</b>	Move cursor to matching brace
<b>Ctrl+Shift+E</b>	Extend selection to matching brace
<b>Ctrl+Shift+C</b>	Toggle line comment on selection
<b>Ctrl+Shift+X</b>	Toggle stream comment on selection
<b>Ctrl+F2</b>	Toggle bookmark
<b>F2</b>	Go to next bookmark
<b>Shift+F2</b>	Go to previous bookmark
<b>Ctrl+Shift+F2</b>	Clear all bookmarks in current file
<b>Ctrl+Shift+W</b>	Toggle whitespace characters
<b>Ctrl+Shift+L</b>	Toggle EOL characters
<b>Ctrl+Spacebar</b>	Invoke autocomplete
<b>Ctrl+-</b>	Go backwards in cursor history
<b>Ctrl+Shift+-</b>	Go forwards in cursor history
<b>F12</b>	Start/Cancel search for keyword in file(s).
<b>Ctrl+F</b>	Find text
<b>Ctrl+R</b>	Replace text

**Notes**

- In addition to these keys, you can assign (**Ctrl+Alt+<n>**) key combinations to macros that you define within the Source Code Editor

**Learn more**

- [Editor Language Properties](#)  2251
- [Source Code Viewer Toolbar](#)  2149

## 11.8 Code Build & Debug



This section discusses the integrated Model Driven Development Environment (MDDE).

### Topics

Topic	Detail	See also
<b>Abstract</b>	<p>Model Driven Development provides a more robust, accessible and faster development cycle than traditional coding-driven cycles.</p> <p>A well constructed model, intimately linked with source code build, run, debug, test and deploy capabilities provides a rich, easily navigated and easily understood target architecture.</p> <p>Traceability, linkage to Use Cases, Components and other model artifacts, plus the ability to readily record and document pre-existing or recently developed code, make Enterprise Architect's development environment uniquely powerful.</p> <p>Enterprise Architect incorporates industry standard intelligent editing, debuggers and modeling languages.</p>	<a href="#">Getting Started</a> <sup>[2173]</sup> <a href="#">Analyzer Scripts</a> <sup>[2175]</sup> <a href="#">Editing Source Code</a> <sup>[2146]</sup>
<b>The MDDE</b>	<p>The MDDE provides tools to design, build and debug an application:</p> <ul style="list-style-type: none"> <li>• UML technologies and tools to model software</li> <li>• Code generation tools to generate/reverse engineer source code</li> <li>• Tools to import source code and binaries</li> <li>• Code editors that support different programming languages</li> <li>• Intelli-sense to aid coding</li> <li>• Analyzer scripts that enable a user to describe how to build, debug, test and deploy the application</li> </ul> <pre> pApp = new CBCGPAAppointmentDemo (     ▲ 2 of 2 ▼ CBCGPAAppointmentDemo::CBCGPAAppointmentDemo(COleDateTime&amp; dtStart,     COleDateTime&amp; dtFinish, CString&amp; strText, COLORREF clrBackground, COLORREF     clrForeground, COLORREF clrDuration)     RGB (165, 222, 99),     CLR_DEFAULT,     RGB(128, 0, 128) ); </pre>	<a href="#">Development Tools</a> <sup>[2074]</sup> <a href="#">MDG Technologies</a> <sup>[1475]</sup> <a href="#">Modeling Basics</a> <sup>[750]</sup> <a href="#">Generate Source Code</a> <sup>[2111]</sup> <a href="#">Import Source Code</a> <sup>[2138]</sup> <a href="#">Import Binary Module</a> <sup>[2143]</sup> <a href="#">Editing Source Code</a> <sup>[2146]</sup> <a href="#">Intelli-sense</a> <sup>[2160]</sup> <a href="#">Managing Analyzer Scripts</a> <sup>[2175]</sup>

### Notes

- The Enterprise Architect debugger cannot attach to 64-bit processes; you can only debug 32-bit

processes

## 11.8.1 Getting Started

To quickly start development in the Model Driven Development Environment, check through the following topics:

- [Prerequisites](#) <sup>[2173]</sup>
- [General Workflow](#) <sup>[2173]</sup>
- [Workspace Layout and Tools](#) <sup>[2174]</sup>

### 11.8.1.1 Prerequisites

Before using the Model Driven Development Environment:

- You should be using the correct edition: Enterprise Architect Professional, Corporate or extended editions
- You should be connected to the required model
- Relevant source code files should be linked to Classes in the model (sufficient to be editable from the internal code editors) either:
  - Using MDA Transforms to generate stub code files, and internal editing to write code
  - Reverse engineering source code from existing files, or
  - Importing existing source code from external files
- Required External Frameworks or compilers should be installed and operable
- An Analyzer Script should be configured to link to the compilers in the External Framework

#### Learn more

- [Analyzer Scripts](#) <sup>[2175]</sup>
- [Importing Source Code](#) <sup>[2136]</sup>
- [Editing Source Code](#) <sup>[2146]</sup>
- [Add Build Commands](#) <sup>[2180]</sup>
- [Debugging](#) <sup>[2222]</sup>

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Software Engineering | Import Code**
- (Alt+F1) | **Enterprise Architect | Software Engineering | Code Editor | Using Intelli-sense**
- (Alt+F1) | **Enterprise Architect | Software Engineering | Code Editor | Browse to Source File**

### 11.8.1.2 General Workflow

In working with the Model Driven Development Environment, the build-debug workflow is a cyclic process. Beginning with configuring and setting up scripts, each iteration of the cycle can include:

- Model - Edit - Build - Debug - Test - Visual Analysis - Profile

When the cycle is complete, you document and deploy the solution.




#### Code Synchronization:

It is important that the model and source code are kept synchronized for the Visual Execution Analyzer to

produce useful results; use the Code Generation tools to synchronize your model after any design changes or code editing.

Always build the application prior to any Execution Analysis session - debugging, recording or profiling.

#### Learn more

- [Managing Analyzer Scripts](#) 
- [Build Application](#) 
- [Update Package Contents](#) 

### **11.8.1.3 Workspace Layout and Tools**

This topic identifies the workspace layouts that you can select for debugging and Visual Execution Analysis, and the tools these layouts make available in the Model Driven Development Environment.

#### Workspace Layouts

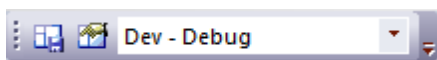
You can choose from many predefined workspace layouts, depending on the tasks you perform. When you are familiar with the environment and controls available to you, you can also define your own workspace layouts.

You can select from the following debug/execution analysis layouts:

- Dev - Debug
- VEA - Profile
- VEA - Record
- VEA - Simulation
- VEA - Test Points

You select the workspace layout from either the:

- Workspace Toolbar or
- Workspaces tab of the Workspaces and Commands dialog



#### Tools

Whether you open the appropriate windows through a workspace layout or select them individually through their menu options, the following tools and facilities are of particular relevance to coding, building and debugging:

- Code engineering
- Source Code Editors
- Breakpoints and Markers
- Intelli-sense
- Execution Analyzer
- Debug window
- Locals window

- Call Stack window

#### Learn more

- [Workspace Layouts Toolbar](#)<sup>[149]</sup>
- [Manage Workspace Layout](#)<sup>[163]</sup>
- [Code Engineering](#)<sup>[2073]</sup>
- [Editing Source Code](#)<sup>[2146]</sup>
- [Intelli-sense](#)<sup>[2160]</sup>
- [Analyzer Scripts](#)<sup>[2175]</sup>
- [Build Application](#)<sup>[2221]</sup>
- [Breakpoint and Marker Management](#)<sup>[2224]</sup>
- [Debugger Facilities](#)<sup>[2231]</sup>

### 11.8.2 Analyzer Scripts

*Analyzer Scripts* are used to record information on compiling and running the application under development.

An Analyzer Script can be defined for a package-tree containing Classes with links to the code. The information you provide in a script controls the actions in, and provides support for:

- Building
- Debugging
- Testing
- Execution
- Deployment

#### Learn more

- [Analyzer Script Editor](#)<sup>[2179]</sup>
- [Managing Analyzer Scripts](#)<sup>[2175]</sup>
- [Add Build Commands](#)<sup>[2180]</sup>
- [Clean](#)<sup>[2181]</sup>
- [Debug](#)<sup>[2222]</sup>
- [Testing](#)<sup>[2182]</sup>
- [Add Run Command](#)<sup>[2217]</sup>
- [Add Deploy Command](#)<sup>[2218]</sup>
- [Configure Recording Detail](#)<sup>[2219]</sup>
- [Set Up Simulation Script](#)<sup>[2472]</sup>
- [Visual Execution Analyzer](#)<sup>[2527]</sup>

#### 11.8.2.1 Managing Analyzer Scripts


The Execution Analyzer window enables you to manage all Analyzer scripts in the model. You use the window toolbar buttons or script context menu options to control script tasks. Scripts are listed by package; the list only shows packages that have Analyzer scripts defined against them.

Each user can activate one script at any time, independent of other users of the same model; one user







activating a script does not impact the currently active scripts for other users or affect the scripts available to them.

**Access** **Analyzer | Execution Analyzer (Shift+F12)**

**Toolbar Options:**

Toolbar Button	Action	See also
	List the Analyzer windows; select and display the required window.	<a href="#">Debugging</a> <sup>[2222]</sup> <a href="#">Breakpoint and Marker Management</a> <sup>[2224]</sup> <a href="#">View the Call Stack</a> <sup>[2239]</sup> <a href="#">View the Local Variables</a> <sup>[2234]</sup> <a href="#">View Variables in Other Scopes</a> <sup>[2237]</sup> <a href="#">Inspect Process Memory</a> <sup>[2242]</sup> <a href="#">The Recording History</a> <sup>[2533]</sup> <a href="#">The Testpoints Window</a> <sup>[2578]</sup> <a href="#">Profiling</a> <sup>[2555]</sup> <a href="#">Model Simulation</a> <sup>[2463]</sup>
	Add new Analyzer Script.  A prompt displays to browse for and select the parent package, followed by the Analyzer Script Editor.	<a href="#">Analyzer Script Editor</a> <sup>[2179]</sup>
	Export Scripts.  Export one or more Analyzer Scripts to an XML file, which can be used to import the scripts into another model.  The Execution Analyzer: Export dialog displays, from which you select the script or scripts to export, followed by a prompt for the target file name and location.	
	Import Scripts.  Import one or more Analyzer Scripts into the current model from a previously exported XML file.  The Browse Project dialog displays, on which you select the package into which to import the scripts, followed by a prompt for the source file name and location.	



Toolbar Button	Action	See also
	Execute the Build command of the active script.	<a href="#">Add Commands</a> <sup>[2180]</sup>
	Cancel the Build command currently in progress.	
	Execute the Run command of the active script.	<a href="#">Add Run Command</a> <sup>[2217]</sup>
	Execute the Test command of the active script.	<a href="#">Add Testing Command</a> <sup>[2182]</sup>
	Execute the Deploy command of the active script.	<a href="#">Add Deploy Command</a> <sup>[2218]</sup>
	Display the Help topic for this window.	

#### Context Menu Options:

Right-click on the required script or package to display the context menus.

Option	Action	See also
<b>Add New Script</b>	Add a new script to the selected package. The Analyzer Scripts Editor displays, showing the Build page.	<a href="#">Analyzer Script Editor</a> <sup>[2179]</sup>
<b>Paste Script</b>	Paste a copied script from the Enterprise Architect clipboard into the selected package. You can paste the copied script several times; each copy has the suffix <b>Copy</b> . To rename the copied script, press <b>(F2)</b> and overwrite the script name.	
<b>Export Scripts</b>	Export scripts from the selected package. The Execution Analyzer: Export dialog displays, from which you select the script or scripts to export, followed by a prompt for the target file name and location.	
<b>Import Scripts</b>	Import scripts from a .XML file into the selected package. A prompt displays for the source file name and location.	

Option	Action	See also
<b>Select In Project Browser</b>	Highlight the selected package in the Project Browser. Display the Project Browser, which is now expanded to show the highlighted package.	
<b>Build</b>	Execute the Build command of the selected script.	<a href="#">Build Application</a> <sup>[2221]</sup>
<b>Clean</b>	Execute the Clean command of the selected script.	<a href="#">Clean</a> <sup>[2181]</sup>
<b>Rebuild</b>	Execute the Clean and Build commands of the selected script.	<a href="#">Build Application</a> <sup>[2221]</sup>
<b>Debug</b>	Execute the Debug command of the selected script.	<a href="#">Set up a Debug script</a> <sup>[2185]</sup>
<b>Run</b>	Execute the Run command of the selected script.	<a href="#">Add Run Command</a> <sup>[2217]</sup>
<b>Test</b>	Execute the Test command of the selected script.	<a href="#">Add Testing Command</a> <sup>[2182]</sup>
<b>Deploy</b>	Execute the Deploy command of the selected script.	<a href="#">Add Deploy Command</a> <sup>[2218]</sup>
<b>Start Simulation</b>		
<b>Edit</b>	Open the selected script in the Analyzer Scripts Editor.	<a href="#">Analyzer Script Editor</a> <sup>[2179]</sup>
<b>Copy</b>	Copy the selected script to the Enterprise Architect clipboard.	
<b>Paste</b>	Paste the most-recently copied script to the same package as the selected script.  You can paste the copied script several times; each copy has the suffix <b>Copy</b> .  To rename the copied script, press <b>(F2)</b> and overwrite the script name.	
<b>Delete</b>	Delete the selected script; there is no prompt for confirmation.  To delete a <i>package</i> from the Execution Analyzer window, delete the scripts from the package; when the last script is deleted, the package is no longer listed.	

Option	Action	See also
<b>Set as Model Default</b>	Set the selected script as the default script for the model. The icon to the left of the script changes color; any previously-set default script reverts to normal.	
<b>Help</b>	Display the Help topic for this window.	

### Learning Center topics


- (Alt+F1) | **Enterprise Architect | Build and Debug | Overview | Analyzer Scripts**

#### 11.8.2.2 Analyzer Script Editor

Analyzer Scripts are associated with a Package. When you create an Analyzer Script you can define a number of actions - many features, such as Debugging, Testing, Profiling and Simulation, depend on and function according to how the active script has been configured.

The Execution Analyzer Script Editor has a number of pages for configuring various scripts; if you plan to use any of the features of the Execution Analyzer, you must complete at least the Build and Debug pages.

Task	Page	Actions	See also
<b>Build</b>	Build	Enter script or command to build the application.	<a href="#">Add Build Commands</a> <sup>[2180]</sup>
	Clean	Enter script or command to clean the previous build .	<a href="#">Clean</a> <sup>[2181]</sup>
<b>Test</b>	Test	Enter script or command to test the application.	<a href="#">Test</a> <sup>[2182]</sup>
	Testpoints	Specify where the output from a Testpoint run is sent.	<a href="#">Testpoint Output</a> <sup>[2184]</sup>
<b>Debug</b>	Platform	Specify the debugging platform, the application to be debugged, and the mode of debugging (attach to process or run ).	<a href="#">Debug</a> <sup>[2222]</sup>
	Tracepoints	Specify where the output from Tracepoints encountered during a debug session are sent.	<a href="#">Tracepoint Output</a> <sup>[2215]</sup>
	Workbench	Specify required information for using the Object Workbench control.	<a href="#">Workbench Setup</a> <sup>[2216]</sup>
<b>Run</b>	Run	Enter a script or command to run the application.	<a href="#">Add Run Command</a> <sup>[2217]</sup>
<b>Deploy</b>	Deploy	Enter a script or command to deploy the project.	<a href="#">Add Deploy Command</a> <sup>[2218]</sup>
<b>Recording</b>	Recording	Exclude areas of the application from being recorded; you can specify wildcard expressions to exclude operations, Classes and modules.  Here you can also enable recording of instances.	<a href="#">Configure Recording Detail</a> <sup>[2219]</sup>

Task	Page	Actions	See also
<b>Simulation</b>	Simulation	Complete the configuration for Simulation Control.	<a href="#">Model Simulation</a> 

### 11.8.2.3 Add Build Commands



This topic describes how you specify the commands to build the project or package, using the Execution Analyzer Script Editor.

The Build page enables you to enter commands to build your project. You can use Enterprise Architect Local Paths and environment variables in composing your command line(s).

**Access** **Analyzer | Execution Analyzer (Shift+F12) : New** (Toolbar icon) > **Build > Build Analyzer | Execution Analyzer (Shift+F12)** : locate and double-click on required script > **Build > Build**

#### Reference

On the Execution Analyzer Script Editor:

Field	Action	See also
<b>Execute Command As:</b>	<p><b>Batch File</b></p> <p>Use this option to create a build script. The script is executed in a system command window. Environment variables can be accessed by commands in this script.</p> <p><b>Process</b></p> <p>Use this option to run a single program.</p> <p>The command should specify the path to the program, plus any command line arguments; if the executable path or any arguments contain spaces, they must be surrounded by quotes.</p> <p>When this option is selected for Java, you can build all java files in child directories by adding the <code>%r</code> token. The token should immediately precede the files to be built. Enterprise Architect will then execute the command in each child directory. (see example below)</p>	<a href="#">Analyzer Script Editor</a> 
<b>Build Script</b>	Write your script in the large text box, using the standard Windows Command Line commands; the format and content of this section depends on the actual compiler, make system, linker and so on that you use to build your project.	<a href="#">Local Paths</a> 

Field	Action	See also
	<p>You can specify, for example, compiler and linker options, and the names of output files; you can also wrap up all these commands into a convenient batch file and call that here instead.</p> <p><b>Examples:</b></p> <p><b>Visual Studio:</b></p> <pre>"C:\Program Files (x86)\Microsoft Visual Studio 9.0\Common7\IDE\devenv.com" /Rebuild Debug RentalSystem.sln</pre> <p><b>Using a Local Path:</b></p> <pre>"%VsCompPath%\devenv.exe" /build Debug Subway.sln</pre> <p><b>Java:</b></p> <pre>C:\Program Files (x86)\Java\jdk1.6.0_22\bin\javac.exe -g -cp "%classpath%" %*.java</pre> <p><b>Using a Local Path:</b></p> <pre>"%JAVA%\bin\javac.exe" -g -cp "%classpath%" %*.java</pre>	
<b>Default Directory</b>	Type in or browse for the default directory path into which the Build commands are to load the built application files.	
<b>Parse Output</b>	<p>This enables you to select a method for automatically parsing the compiler output.</p> <p>If you select this option, output from the script is logged in the System Output window; Enterprise Architect parses the output according to the syntax you specify.</p>	<a href="#">The Output Window</a> <sup>169</sup>

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Build and Debug | Overview | Build**
- (Alt+F1) | **Enterprise Architect | Build and Debug | Build and Run | Build Scripts - Native**
- (Alt+F1) | **Enterprise Architect | Build and Debug | Build and Run | Build Scripts - .NET**
- (Alt+F1) | **Enterprise Architect | Build and Debug | Build and Run | Build Scripts - Java**

#### 11.8.2.4 Clean

This topic explains how you enter a command to clean your project prior to a build; for example, to delete cached items used for building the project so that the next build recreates from scratch.

In Java you might delete all Class files, jar files and .war files in the project; Visual Studio projects, however, might issue a call to the Visual Studio Clean operation instead.

When you execute the **Rebuild** menu option on a script, the command you specify here is executed followed by the Build command.

**Access** **Analyzer | Execution Analyzer (Shift+F12) : New** (Toolbar icon) > **Build > Clean**  
**Analyzer | Execution Analyzer (Shift+F12) : locate and double-click on required script > Build > Clean**

### Topics

Topic	Detail	See also
<b>Usage</b>	Enter the command to be executed when you select <b>Clean</b> from the script context menu.	
<b>Example</b>	<code>devenv.com / Clean Debug MyProject.sln</code>	

### Learn more

- [Add Build Commands](#) 

## 11.8.2.5 Testing

The Execution Analyzer supports testing using xUnit testing applications and Testpoints.

This section describes how to create commands for performing unit testing and Testpoint testing on your code.

### Learn more

- [Add Testing Command](#) 
- [Testpoints Output](#) 

### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Execution Analysis | xUnit Testing**
- (Alt+F1) | **Enterprise Architect | Execution Analysis | Testpoint Testing**







### 11.8.2.5.1 Add Testing Command

This topic explains how to configure the Test Page of an Analyzer Script for performing unit testing on your code.

In order for this to be possible, you must first create a Build script for the appropriate package.

**Access** **Analyzer | Execution Analyzer (Shift+F12) : New** (Toolbar icon) > **Test > Test**  
**Analyzer | Execution Analyzer (Shift+F12) : locate and double-click on required script > Test > Test**

Topics

Topic	Detail	See also
<b>Usage</b>	<p>A sample script would contain a line to execute the testing tool of your choice, with the filename of the executable produced by the <b>Build</b> command as the option.</p> <p>Testing could be integrated with any test tool using the command line provided, but in these examples you can see how to integrate <i>NUnit</i> and <i>JUnit</i> testing with your source code.</p> <p>Enterprise Architect provides inbuilt MDA Transforms from source to Test Case, plus the ability to capture <i>xUnit</i> output and use it to go directly to a test failure; <i>xUnit</i> integration with your model is a powerful means of delivering solid and well-tested code as part of the complete model-build-test-execute-deploy life-cycle.</p> <p>NUnit and JUnit must be downloaded and installed prior to their use; Enterprise Architect does not include these products in the base installer.</p>	<p><a href="#">Managing Analyzer Scripts</a>  [2175]</p> <p><a href="#">Add Build Commands</a>  [2180]</p> <p><a href="#">JUnit Transformation</a>  [2037]</p> <p><a href="#">NUnit Transformation</a>  [2038]</p> <p><a href="#">Unit Testing</a>  [2573]</p>
<b>Execute Command As:</b>	<p><b>Process</b></p> <p>Enter the path to a program or batch file to run followed by any parameters.</p> <p><b>Batch File</b></p> <p>When using this option you can enter multiple commands, which are then executed as a single script in a command console; you have access to any environment variables available in a standard command console.</p>	
<b>Example</b>	<p><b>NUnit</b></p> <pre>" C: \ P r o g r a m F i l e s \ N U n i t \ b i n \ n u n i t - c o n s o l e . e x e " p i n \ d e b u g \ C a l c u l a t o r . e x e "</pre> <p><b>JUnit</b></p> <pre>j a v a j u n i t . t e x t u i . T e s t r u n n e r %N</pre> <p>The command listed in this field is executed as if from the command prompt; as a result, if the executable path or any arguments contain spaces, they must be surrounded in quotes.</p>	
<b>Default Directory</b>	Preset to the Build default directory.	<a href="#">Add Commands</a>  [2180]
<b>Parse Output</b>	<p>Selecting an Output Parser causes output from the command(s) to be captured to the Output window; the <b>Output Parser</b> field specifies the syntax to be expected.</p> <p>Double-clicking on a result in the Output window opens the corresponding code segment in Enterprise Architect's code</p>	<a href="#">Locate Compiler</a>

Topic	Detail	See also
	window.	<a href="#">Errors in Code</a> <sup>[2221]</sup>
<b>Build First</b>	Select to ensure that the package is recompiled each time you run the test.  If you include the string %N in your test script it is replaced by the fully namespace-qualified name of the currently selected Class when the script is executed.	

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Build and Debug | Test**

#### 11.8.2.5.2 Testpoints Output

The Testpoints page of the Analyzer Script allows you to configure the output of a Testpoint run.

**Access** **Analyzer | Execution Analyzer (Shift+F12) : New** (Toolbar icon) > **Test > Testpoints**  
**Analyzer | Execution Analyzer (Shift+F12) : locate and double-click on required script > Test > Testpoints**

Topic	Detail	See also
<b>Output</b>	You can select from two options: <ul style="list-style-type: none"> <li>• <b>Screen</b> (Default) - The output is directed to the Testpoints tab of the System Output window</li> <li>• <b>File</b> - The output is directed to file</li> </ul>	<a href="#">Testpoint Management</a> <sup>[2577]</sup>
<b>Folder</b>	Enter the folder to use for Testpoint log files.	
<b>Filename</b>	Enter the name to use for the Testpoints log files.	
<b>Overwrite</b>	When this option is selected, the file specified is overwritten each time a Testpoint run is performed.	
<b>Auto Number</b>	When this option is selected, the Testpoint output is composed of the filename you specify and the number of the Test run; each time you perform a Test run the number is incremented.	
<b>Prefix trace output with function</b>	When this option is selected, any trace statements executed during the Testpoint run are prefixed with the current function call.	



Learning Center topics

- (Alt+F1) | **Enterprise Architect | Execution Analysis | Testpoints | Introducing Testpoints**

**11.8.2.6 Set up a Debug script**

The process of configuring the Debug section of an Analyzer Script is quick and easily accomplished, yet provides access to all the features of Enterprise Architect's Visual Execution Analyzer, namely:

- Debugging
- Sequence diagram recording
- Testpoint management and
- Process profiling

All you need to do is select the appropriate platform for your project and enter some basic details. The debugger platforms you can use include:

- Java
- Java Debug Wire Protocol (JDWP)
- Microsoft .NET Debugger
- Microsoft C++ and Native Code (C, VB) Debugger
- The PHP Debugger
- The GNU Debugger (GDB)

**Access** **Analyzer | Execution Analyzer (Shift+F12) : New** (Toolbar icon) > **Debug > Platform Analyzer | Execution Analyzer (Shift+F12) : locate and double-click on required script > Debug > Platform**

Notes

- An Analyzer script is not necessary for debugging script languages such as JavaScript (Mozilla), VBScript and JScript (Microsoft)

Learn more

- [Operating System Specific Requirements](#) <sup>[2186]</sup>
- [Microsoft C++ and Native \(C, VB\)](#) <sup>[2189]</sup>
- [Java](#) <sup>[2191]</sup>
- [.NET](#) <sup>[2200]</sup>
- [The PHP Debugger](#) <sup>[2204]</sup>
- [The GNU Debugger \(GDB\)](#) <sup>[2209]</sup>
- [The Android Debugger](#) <sup>[2210]</sup>
- [Java JDWP Debugger](#) <sup>[2213]</sup>
- [Script Debugging](#) <sup>[2801]</sup>






Learning Center topics

- (Alt+F1) | **Enterprise Architect | Build and Debug | Overview | Debug**

### 11.8.2.6.1 Operating System Specific Requirements

The Enterprise Architect debugger is able to operate on a number of different platforms. Each of these has different requirements before debugging will be successful. The following table describes the different requirements.

#### Topics

Topic	Detail	See also
<b>Prerequisites</b>	Creation of an Analyzer script and configuration of the <b>Debug</b> command in that script.	
<b>Supported Platforms</b>	<p>Enterprise Architect supports debugging on these platforms:</p> <p><b>.NET</b></p> <ul style="list-style-type: none"> <li>Microsoft™ .NET Framework 1.1 and later, including .NET 4</li> <li>Language support: C, C#, C++, J#, Visual Basic</li> <li>Target Platform on 64 bit systems must be either x86 or x64; do not use the <b>Any CPU</b> configuration</li> </ul> <p><b>Java</b></p> <ul style="list-style-type: none"> <li>Java SE Development Kit from Oracle™. (version 5.0 minimum) (either 32-bit or 64-bit JDK)</li> <li>Java EE Development Kit from Oracle™. (version 5.0 minimum) (either 32-bit or 64-bit JDK)</li> </ul> <p>Debugging is implemented through the Java Virtual Machine Tools Interface (JVMTI), which is part of the Java Platform Debugger Architecture (JPDA); the JPDA was introduced in Java SE version 5.0.</p> <p><b>Windows for Native Applications</b></p> <p>Enterprise Architect supports debugging native code (C, C++ and Visual Basic) compiled with the Microsoft™ compiler where an associated PDB file is available.</p> <p><b>PHP</b></p> <p>Enterprise Architect enables you to perform local and remote debugging with PHP.exe.</p> <p><b>GNU Debugger (GDB)</b></p> <p>Enterprise Architect supports debugging using the GNU Debugger, which enables you to debug your applications under Linux either locally or remotely.</p> <p>Requires GDB version 7.0 or above.</p> <p>Source code file path must not contain spaces.</p>	<p><a href="#">.NET</a>  [2200]</p> <p><a href="#">Java</a>  [2191]</p> <p><a href="#">Microsoft C++ and Native (C, VB)</a>  [2189]</p> <p><a href="#">The PHP Debugger</a>  [2204]</p> <p><a href="#">The GNU Debugger</a>  [2209]</p>

### Notes

- Debugging under Windows Vista (x64) - if you encounter problems debugging with Enterprise Architect on a 64-bit platform, you should build a Win32 platform configuration in Visual Studio; that is, do not specify **ANY-CPU**, specify **WIN32**

#### 11.8.2.6.1.1 UAC-Enabled Operating Systems

The Microsoft operating systems *Windows Vista* and *Windows 7* provide User Account Control (UAC) to manage security for applications.

The Enterprise Architect Visual Execution Analyser is UAC-compliant, and users of UAC-enabled systems can perform operations with the Visual Execution Analyser and related facilities under accounts that are members of only the *Users* group.

However, when attaching to processes running as services on a UAC-enabled operating system, it might be necessary to log in as an Administrator.

### How to

To log in as an Administrator, follow the step below:



Step	Action	See also
1	Before you run Enterprise Architect, right-click on the Enterprise Architect icon on the desktop and select the <b>Run as administrator</b> option.	

Alternatively, to edit or create a link to Enterprise Architect and configure the link to run as an Administrator

Step	Action	See also
1	Right-click on the Enterprise Architect icon and select the <b>Properties</b> menu option. The Enterprise Architect Properties dialog displays.	
2	Click on the <b>Advanced</b> button. The Advanced Properties dialog displays.	
3	Select the <b>Run as administrator</b> checkbox.	
4	Click on the <b>OK</b> button, and again on the Enterprise Architect Properties dialog.	

### 11.8.2.6.1.2 WINE Debugging

#### Configure Enterprise Architect to debug under WINE

Step	Action	See also
1	At the command line, run <code>\$ winecfg</code> .	
2	Set the library overrides for <code>dbghelp</code> to <b>(native, builtin)</b> , and accept the warning about overriding this DLL.	
3	Set <code>dbghelp</code> to <b>native</b> by using <code>winecfg</code> .	
4	Copy the application source code plus executable(s) to your bottle. The path must be the same as the compiled version; that is:  If Windows source = <code>C: \ Sour ce\ Sampl eApp</code> , under Crossover it must be <code>C: \ Sour ce\ Sampl eApp</code>	
5	Copy any Side-By-Side assemblies that are used by the application.	
6	Import the source code into Enterprise Architect (optional).	
7	Create a build script on a package. Set the path of the application on the Debug tab, and set the <b>Use Debugger</b> field to <b>Microsoft Native</b> .	<a href="#">Add Build Commands</a>  <sup>[2180]</sup>
8	Open the Profiler ( <b>Analyzer   Profiler</b> ).	<a href="#">Profiling</a>  <sup>[2555]</sup>
9	Click on the <b>Launch</b> button (first button on the Profiler window).	
10	If the sample didn't start, click on the <b>Sampling</b> button (third button on the Profiler window).	
11	Once you have finished profiling, shut down the application ( <i>not</i> Enterprise Architect).	
12	View the Sampler report by clicking the <b>View Report</b> button (fifth button on the Profiler window).	

### Access Violation Exceptions

Due to the manner in which WINE handles direct drawing and access to DIB data, an additional option is provided on the drop-down menu on the Debug window toolbar to ignore or process access violation exceptions thrown when your program directly accesses DIB data.

Select this option to catch genuine (unexpected) access violations; deselect it to ignore expected violations.

As the debugger cannot distinguish between expected and unexpected violations, you might have to use trial and error to capture and inspect genuine program crashes.

### Notes

- If WINE crashes, the back traces might not be correct
- If you are using MFC remember to copy the debug side-by-side assemblies to the `C:\windows\winsxs` directory
- To add a windows path to WINE, modify the Registry entry:

```
HKEY_LOCAL_MACHINE\System\Current Control Set\Control\Session  
Manager\Environment
```

#### **11.8.2.6.2 Microsoft C++ and Native (C, VB)**

You can debug native code only if there is a corresponding PDB file for the executable; you normally create the PDB file as a result of building the application.

The build should include full debug information and there should be no optimizations set.

The script must specify two things to support debugging:

- The path to the executable
- Microsoft Native as the debugging platform

### Learn more

- [General Setup](#) 

### Learning Center topics

- (Alt+F1) | **Enterprise Architect** | **Build and Debug** | **Debug** | **Microsoft Native**

#### **11.8.2.6.2.1 General Setup**


This is the general setup for debugging Microsoft Native Applications (C++, C, Visual Basic). You have two options when debugging:

- Debug an application
- Attach to an application that is running

### Option 1 - Debug an application

Field	Action	See also
<b>Debugger</b>	Select <b>Microsoft Native</b> as the debugging platform.	
<b>x64</b>	Select this checkbox if you are debugging a <b>64-bit</b> application. Deselect the checkbox if you are debugging a <b>32-bit</b> application.	
<b>Mode</b>	Select the <b>Run</b> radio button.	
<b>Default Directory</b>	This is set as the default directory for the process being debugged.	
<b>Application Path</b>	Select and enter either the full or the relative path to the application executable. <ul style="list-style-type: none"> <li>If the path contains spaces, specify the full path; do not use a relative path</li> <li>If the path contains spaces, the path must be enclosed by quotes</li> </ul>	
<b>Command Line Arguments</b>	Parameters to pass to the application at startup	
<b>Show Console</b>	Create a console window for the debugger; not applicable for attaching to a process.	
<b>Symbol Search Paths</b>	Specify any additional paths to locate debug symbols for the debugger; separate the paths with a semi-colon.	

#### Option 2 - Attach to an application that is running

Field	Action	See also
<b>Debugger</b>	Select <b>Microsoft Native</b> as the debugging platform.	
<b>x64</b>	Select this checkbox if you are debugging a <b>64-bit</b> application. Deselect the checkbox if you are debugging a <b>32-bit</b> application.	
<b>Mode</b>	Select the <b>Attach to Process</b> radio button.	<a href="#">Debug Another Process</a>  2243

Field	Action	See also
<b>Symbol Search Paths</b>	Specify any additional paths to locate debug symbols for the debugger.  You could specify a symbol server here if you prefer; separate the paths with a semi-colon or comma.	

#### 11.8.2.6.2.2 Debug Symbols

##### Topics

Topic	Detail	See also
<b>Abstract</b>	<p>For applications built using <i>Microsoft Platform SDK</i>, Debug Symbols are written to an application PDB file when the application is built.</p> <p>The <i>Debugging Tools for Windows</i>, an API used by the Visual Execution Debugger, uses these symbols to present meaningful information to Execution Analyzer controls.</p> <p>These symbols can easily get out of date and cause errant behaviour - the debugger might highlight the wrong line of code in the editor whilst at a breakpoint; it is therefore best to ensure the application is built prior to any debugging or recording session.</p> <p>The debugger must inform the API how to reconcile addresses in the image being debugged; it does this by specifying a number of paths to the API that tell it where to look for PDB files.</p> <p>For system DLLs (<i>kernel32</i>, <i>mfc90ud</i>) for which no debug symbols are found, the Call Stack shows some frames with module names and addresses only.</p> <p>You can supplement the symbols translated by passing additional paths to the API; you pass additional symbol paths in a semi-colon separated list in the Debug tab.</p>	

#### 11.8.2.6.3 Java

This section describes how to set up Enterprise Architect for debugging Java applications and Web Servers.

##### Learn more

- [General Setup for Java](#) <sup>2191</sup>
- [Advanced Techniques](#) <sup>2194</sup>
- [Working with Java Web Servers](#) <sup>2195</sup>

#### 11.8.2.6.3.1 General Setup for Java

The general setup for debugging Java Applications supports two options:

- Debug an Application
- Attach to an application that is running

Option 1 - Debug an Application

Field	Action	See also
Debugger	Select <b>Java</b> .	
x64	Select this checkbox if you are debugging a <b>64-bit</b> application. Deselect the checkbox if you are debugging a <b>32-bit</b> application.	
Mode	Select <b>Run</b> .	
Default Directory	This path is added to the class path property when the Java Virtual Machine is created.	
Application Class	Identify the fully qualified Class name to debug; the Class must have a method declared with the following signature:  <pre>public static void main( String( ) );</pre> <div><div>Application Class</div><div>samples.Collector</div></div> <div><div>Command Line Arguments:</div><div>"param1" param2 "param3" param4</div></div>	
Command Line Arguments	Specify any parameters to be passed to the main method of the Application Class above.  Parameters containing spaces should be surrounded with double quotes.	
Java Virtual Machine Options	Specify command line options for Virtual Machine creation.  You also must provide a parameter ( <b>JRE</b> ) as the path to be searched for the <i>jvm.dll</i> ; this is the DLL supplied as part of the Java runtime environment or Java JDK from Sun Microsystems™.  The JRE parameter can be either: <ul style="list-style-type: none"><li>• An Enterprise Architect-defined <b>Local Path</b></li><li>• An <b>absolute file path</b> (with no double quotes) to the installation folder of the Java JDK to be used for debugging</li></ul> The JRE parameter must point to the installation folder for the Java JDK. A JDK installation is necessary for debugging to succeed. The JRE should not point to the installation of the public Java Runtime Environment, if that is installed. Environment variables can be used when specifying the VM startup options, such as <b>classpath</b> .  For example, using:	



Field	Action	See also
	<ul style="list-style-type: none"> <li>An Enterprise Architect <b>Local Path</b> <i>JAVA</i> and an environment variable <i>classpath</i>:   <div>Java Virtual Machine Options:</div> <pre>JRE=%JAVA%,-Djava.class.path=%classpath%;;</pre> </li> <li>Or an <b>absolute path</b> to the JDK installation directory and an environment variable <i>classpath</i>:   <div>Java Virtual Machine Options:</div> <pre>JRE=C:\Program Files (x86)\Java\jdk1.7.0,-Djava.class.path=%classpath%</pre> </li> </ul> <p>In these two examples, the debugger will create a virtual machine using the JDK located at the value of the <b>JRE</b> parameter.</p> <p>If no classpath is specified, the debugger always creates the virtual machine with a class path property equal to any path contained in the environment variable plus the path entered in the default working directory of this script.</p> <p>If source files and .class files are located under different directory trees, the classpath property <b>MUST</b> include both root path(s) to the source and root path(s) to binary class files.</p>	

### Option 2 - Attach to Virtual Machine

There is very little to specify when attaching to a VM; however, the VM must have the Sparx Systems debugging agent loaded.

Field	Action	See also
<b>Debugger</b>	Select <b>Java</b>	<a href="#">Attach to Virtual Machine</a> <sup>[2194]</sup> <a href="#">Debug Tomcat Server</a> <sup>[2195]</sup>
<b>Mode</b>	Select <b>Attach to Virtual Machine</b>	<a href="#">Debug Another Process</a> <sup>[2243]</sup>

### Learning Center topics

- (Alt+F1) | **Enterprise Architect** | **Build and Debug** | **Debug** | **Java**

### 11.8.2.6.3.2 Advanced Techniques

In addition to the standard Java debugging techniques, you can:


- [Attach to a Virtual Machine](#) <sup>[2194]</sup>
- [Debug Internet Browser Java Applets](#) <sup>[2194]</sup>

#### Topics

Topic	Detail	See also
<b>Abstract</b>	<p>You can debug a Java application by attaching to process that is hosting a Java Virtual Machine; you might want to do this for attaching to a webserver such as Tomcat or JBOSS.</p> <p>The <i>Java Virtual Machine Tools Interface</i> from Sun Microsystems, is the API used by Enterprise Architect; it allows a debugging agent to be specified when the JVM is created.</p> <p>To debug a running JVM from Enterprise Architect, the Sparx Systems' debugging agent must have been specified as a startup option to the JVM when it was started; how this accomplished for products such as Tomcat and JBOSS should be researched from that product's own documentation.</p> <p>For <i>java.exe</i>, the command line option to load the Enterprise Architect debugging agent depending on your environment could be:</p> <ul style="list-style-type: none"> <li>• - agent path: "c:\program files\sparx systems\ea\VEA\x86\SSJavaProfiler32"</li> <li>• - agent path: "c:\program files (x86)\sparx systems\ea\VEA\x86\SSJavaProfiler32"</li> <li>• - agent path: "c:\program files (x86)\sparx systems\ea\VEA\x64\SSJavaProfiler64"</li> </ul> <p>The the appropriate option will depend on your operating system and whether you are working on a 32-bit application or a 64-bit application.</p> <p>Alternatively, if you add the appropriate VEA directory to your PATH environment variable you can choose to use:</p> <ul style="list-style-type: none"> <li>• - agent lib: SSJavaProfiler32</li> <li>• - agent lib: SSJavaProfiler64</li> </ul> <p>It is not necessary to configure an Analyzer Script when you attach to a Virtual Machine; you can just use the <b>Attach</b> button on one of the Analyzer toolbars.</p> <p>If you configure an Analyzer Script, there are only two things that must be selected:</p> <ul style="list-style-type: none"> <li>• Select <b>Java</b> as the debugging platform</li> <li>• Choose the <b>Attach to Virtual Machine</b> option</li> </ul>	<a href="#">Debug Another Process</a> <sup>[2243]</sup>

This topic describes the configuration requirements and procedure for debugging Java Applets running in a browser from Enterprise Architect.

#### Attach to the browser process hosting the Java Virtual Machine (JVM) from Enterprise Architect

Step	Action	See also
1	Ensure binaries for the applet code to be debugged have been built with debug information.	
2	Configure the JVM using the Java Control Panel.	
3	In the Java Applet Runtime Settings panel, click on the <b>View</b> button.	
4	On the installed version to use, include one of the following options in the <b>Runtime Parameters</b> field depending on your environment and whether you are working on a 32-bit application or a 64-bit application: <ul style="list-style-type: none"> <li>- agent path: "c:\program files\sparx systems\ea\VEA\x86\SSJavaProfiler32"</li> <li>- agent path: "c:\program files (x86)\sparx systems\ea\VEA\x86\SSJavaProfiler32"</li> <li>- agent path: "c:\program files (x86)\sparx systems\ea\VEA\x64\SSJavaProfiler64"</li> </ul>	
5	In this field add the required Class paths.  At least one of these paths should include the root path of the source files to use in debugging.	
6	Set breakpoints.	<a href="#">Breakpoint and Marker Management</a> 
7	Launch the browser.	
8	Attach to the browser process from Enterprise Architect.	

#### 11.8.2.6.3.3 Working with Java Web Servers

If you are debugging **Java web servers** such as **JBOSS** and **Apache Tomcat** (both Server configuration and Windows Service configuration) in Enterprise Architect, apply these configuration requirements and procedures.

##### Attach to process hosting the Java Virtual Machine from Enterprise Architect

Step	Action	See also
1	Build binaries for the web server code to be debugged, with debug information.	
2	Launch the server with the Virtual Machine startup option described in <i>Server Configuration</i> , below.	

Step	Action	See also
3	Import source code into the Enterprise Architect Model, or synchronize existing code.	
4	Set breakpoints.	<a href="#">Breakpoint and Marker Management</a>
5	Launch the client.	
6	Attach to the process from Enterprise Architect.	

### Server Configuration

The configuration necessary for the web servers to interact with Enterprise Architect must address the following two essential points:

- Any VM to be debugged, created or hosted by the server must have the Sparx Systems Agent command line option specified or in the VM startup option (that is:
  - agent l i b: SSJavaPr of i l e r 32 or - agent l i b: SSJavaPr of i l e r 64)
- The CLASSPATH, however it is passed to the VM, must specify the root path to the Package source files

The Enterprise Architect debugger uses the `java.class.path` property in the VM being debugged, to locate the source file corresponding to a breakpoint occurring in a Class during execution; for example, a Class to be debugged is called:

a . b . C

This is located in physical directory:

C: \ s o u r c e \ a \ b

So, for debugging to be successful, the CLASSPATH must contain the root path:

c : \ s o u r c e

### Analyzer Script Configuration

Using the Debug tab of the Build Script dialog, create a script for the code you have imported and:

- Select the **Attach to process** radio button and, in the field below it, type **attach**
- In the **Use Debugger** field, click on the drop-down arrow and select **Java**

All other fields are unimportant; the **Directory** field is normally used in the absence of any Class path property.

**Run the Debugger**

Step	Action	See also
1	Run the server and check that the server process has loaded the Sparx Systems Agent:  DLL SSJavaPr of i l e r 32. DLL or SSJavaPr of i l e r 64  Use <b>Process Explorer</b> or similar tools to prove that the server process has loaded the agent.	
2	In Enterprise Architect, open the source code and set some breakpoints.	<a href="#">Breakpoint and Marker Management</a> <sup>[2224]</sup>
3	Click on the <b>Run Debug</b> button in Enterprise Architect. The Attach To Process dialog displays.	<a href="#">Debug Another Process</a> <sup>[2243]</sup>
4	Select the server process hosting the application.	
5	Click on the <b>OK</b> button. A confirmation message displays in the Debug window, stating that the process has been attached.	

The breakpoints could show a question mark. In this case the Class might not have been loaded yet by the VM. If the question mark remains even after you are sure the Class containing the breakpoint has been loaded, then either:

- The binaries being executed by the server are not based on the source code
- The debugger cannot reconcile the breakpoint to a source file (check Class paths), or
- The JVM has not loaded the Sparx Systems agent

**Learn more**

- [General Setup for Java](#) <sup>[2191]</sup>
- [JBoss Server](#) <sup>[2198]</sup>
- [Apache Tomcat Server](#) <sup>[2199]</sup>
- [Apache Tomcat Windows Service](#) <sup>[2199]</sup>

In this JBoss example, for a 32-bit application, the source code for a simple servlet is located in the directory location:

```
C:\Benchmark\Java\JBoss\Inventory
```

The binaries executed by JBoss are located in the JAW.EAR file in this location:

```
C:\JBoss\03b-dao\build\distribution
```

The Enterprise Architect debugger has to be able to locate source files during debugging; to do this it also uses the CLASSPATH, searching in any listed path for a matching JAVA source file, so the CLASSPATH must include a path to the root of the Package for Enterprise Architect to find the source during debugging.

This is an excerpt from the command file that executes the JBoss server; the Class to be debugged is at:

```
com/Inventory/dto/carDTO
```

Therefore, the root of this path is included in the **JBoss\_CLASSPATH**.

### Example Code

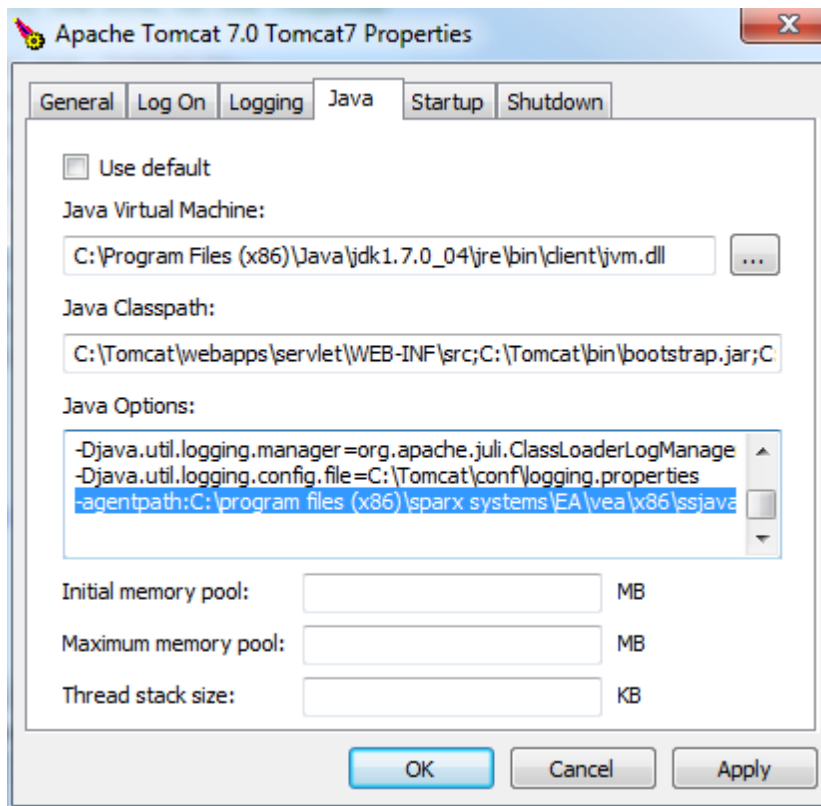
```
RUN. BAT
-----
set SOURCE=C:\Benchmark\Java\JBoss\Inventory

set JAVAC_JAR=%JAVA_HOME%\lib\tools.jar

if "%JBoss_CLASSPATH%" == ""
(
    set JBoss_CLASSPATH=%SOURCE%; %JAVAC_JAR%; %RUNJAR%;
)
else
(
    set JBoss_CLASSPATH=%SOURCE%; %JBoss_CLASSPATH%; %JAVAC_JAR%; %
    RUNJAR%;
)

set JAVA_OPTS=%JAVA_OPTS% -agent path: "c:\program files\sparx
systems\vea\x86\ssjavaprofiler32"
```

The Apache Tomcat Server can be configured for debugging using the Java debugger in Enterprise Architect. This example shows the configuration dialog for Apache Tomcat 7.0 on a PC running Windows 7.



These three points are important:

- The **Java Virtual Machine** specifies the runtime from an installation of the Java JDK
- The source path to any servlet to be debugged is added to **Java Classpath**; in this case we add the path to the Tomcat servlet:

```
c:\tomcat\webapps\servlet\WEB-INF\src
```

- The **Java Options** include the path to the Sparx Systems debugging agent:

```
-agentpath:c:\program files (x86)\sparx systems\vea\x86\ssjavaProfiler32
```

### Learn more

- [Apache Tomcat Windows Service](#) <sup>2199</sup>

### Configuration

For users running Apache Tomcat as a Windows™ service, it is important to configure the service to enable interaction with the Desktop; failure to do so causes debugging to fail within Enterprise Architect.

Log on as:

- ☒ Local System account
- ☒ Allow service to interact with desktop

Select the **Allow service to interact with desktop** checkbox.

#### Learn more

- [Working with Java Web Servers](#) <sup>[2195]</sup>
- [Apache Tomcat Server](#) <sup>[2199]</sup>
- [JBOSS Server](#) <sup>[2198]</sup>

### 11.8.2.6.4 .NET

This section describes how to configure Enterprise Architect for debugging .NET applications. It covers:

- [General Setup](#) <sup>[2200]</sup>
- [Debugging Managed code in an Unmanaged Application](#) <sup>[2201]</sup>
- [Debug COM Interop](#) <sup>[2202]</sup>
- [Debug ASP .NET](#) <sup>[2203]</sup>

#### 11.8.2.6.4.1 General Setup for .NET

This is the general setup for debugging Microsoft .NET applications. You have two options when debugging:

- Debug an Application
- Attach to an application that is running

#### Option 1 - Debug an application

Field	Action	See also
<b>Debugger</b>	Select <b>Microsoft .NET</b> as the debugging platform.	<a href="#">Debugging an Unmanaged Application</a> <sup>[2201]</sup> <a href="#">Debug ASP.NET</a> <sup>[2203]</sup>
<b>x64</b>	Select this checkbox if you are debugging a <b>64-bit</b> application. Deselect the checkbox if you are debugging a <b>32-bit</b> application.	
<b>Mode</b>	Select the <b>Run</b> radio button.	
<b>Default Directory</b>	This is set as the default directory for the process being debugged.	
<b>Application Path</b>	Select and enter either the full or the relative path to the application executable. <ul style="list-style-type: none"> <li>• If the path contains spaces, specify the full path; do not</li> </ul>	



Field	Action	See also
	use a relative path <ul style="list-style-type: none"> <li>If the path contains spaces, the path must be enclosed by quotes</li> </ul>	
<b>Command Line Arguments</b>	Parameters to pass to the application at startup.	
<b>Show Console</b>	Create a console window for the debugger; not applicable to attaching to a process.	
<b>Symbol Search Paths</b>	Specify any additional paths to locate debug symbols for the debugger; separate the paths with a semi-colon.	

#### Option 2 - Attach to an application that is running

Field	Action	See also
<b>Debugger</b>	Select <b>Microsoft .NET</b> as the debugging platform.	<a href="#">Debugging an Unmanaged Application</a> [2201]
<b>x64</b>	Select this checkbox if you are debugging a <b>64-bit</b> application. Deselect the checkbox if you are debugging a <b>32-bit</b> application.	<a href="#">Debug ASP.NET</a> [2203] <a href="#">Debug Another Process</a> [2243]
<b>Mode</b>	Select the <b>Attach to Process</b> radio button.	

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect** | **Build and Debug** | **Debug** | **Microsoft .NET**

#### **11.8.2.6.4.2 Debugging an Unmanaged Application**

##### Topics

Topic	Detail	See also
<b>Usage</b>	If you are debugging managed code using an unmanaged application, the debugger might fail to detect the correct version of the Common Language Runtime (CLR) to load.  You should specify a config file if you don't already have one for the debug application specified in the <b>Debug</b> command of your script.	<a href="http://www.msdn2.microsoft.com/en-us/library/9w519wzk.aspx">http://www.msdn2.microsoft.com/en-us/library/9w519wzk.aspx</a> (Online Resource)


Topic	Detail	See also
	<p>The config file should reside in the same directory as your application, and take the format:</p> <pre>name.exe.config</pre> <p>where <i>name</i> is the name of your application.</p> <p>The version of the CLR you specify should match the version loaded by the managed code invoked by the debuggee.</p> <p>The following is a sample config file:</p> <pre>&lt;configuration&gt;   &lt;startup&gt;     &lt;requiredRuntime       version="version" /&gt;   &lt;/startup&gt; &lt;/configuration&gt;</pre> <p><i>Version</i> is the version of the CLR targeted by your plugin or COM code.</p>	

#### 11.8.2.6.4.3 Debug COM Interop

Enterprise Architect enables you to debug .NET managed code executed using COM in either a Local or an In-Process server.

This feature is useful for debugging Plugins and ActiveX components.

#### How to

Step	Action	See also
1	Create a package in Enterprise Architect and import the code to debug.	<a href="#">Software Engineering</a> 
2	Ensure the COM component is built with debug information.	
3	Create a Script for the Package.	
4	In the Debug   Platform page, you can select to either attach to an unmanaged process or specify the path to an unmanaged application to call your managed code.	
5	Add breakpoints in the source code to debug.	

#### Topics

Topic	Detail	See also
<b>Attach to an Unmanaged Process</b>	<p>If you are using:</p> <ul style="list-style-type: none"> <li>• An In-Process COM server, attach to the client process</li> <li>• A Local COM Server, attach to the server process</li> </ul> <p>Click on the Debug window <b>Run</b> button (or press ( <b>F6</b> ) ) to display a list of processes from which you can choose.</p>	<a href="#">Debug Another Process</a> <sup>[2243]</sup>

**Notes**

- Detaching from a COMinterop process you have been debugging terminates the process; this is a known issue for Microsoft .NET Framework, and information on it can be found on many of the MSDN .NET blogs

**11.8.2.6.4.4 Debug ASP .NET****Topics**

Topic	Detail	See also
<b>Usage</b>	<p>Debugging for web services such as ASP requires that the Enterprise Architect debugger is able to attach to a running service.</p> <p>Begin by ensuring that the directory containing the ASP .NET service project has been imported into Enterprise Architect and, if required, the web folder containing the client web pages.</p> <p>If your web project directory resides under the website hosting directory, you can import from the root and include both ASP code and web pages at the same time.</p> <p>It is necessary to launch the client first, as the ASP .NET service process might not already be running; load the client using your browser - this ensures that the web server is running.</p> <p>In the debug setup you must select the <b>Attach to process</b> radio button, and then type the keyword <b>Attach</b> in the script; this keyword causes the debugger to prompt you for a process at runtime.</p> <p>Click on the Debug window <b>Run</b> button to start the debugger; the Attach To Process dialog displays.</p> <p>The name of the process varies across Microsoft operating systems, as explained in the <i>ASP .NET SDK</i>; for example, under Windows Vista the name of the IIS process is <i>w3wp.exe</i>.</p> <p>On Windows XP, the name of the process resembles <i>aspnet_wp.exe</i>, although the name could reflect the version of the .NET framework that it is supporting.</p> <p>There can be multiple ASP.NET processes running under XP; you must ensure that you attach to the correct version, which would be the one hosting the .NET framework</p>	<a href="#">Run the Debugger</a> <sup>[2231]</sup> <a href="#">Breakpoint and Marker Management</a> <sup>[2224]</sup> <a href="#">Debug Another Process</a> <sup>[2243]</sup>

Topic	Detail	See also
	<p>version that your application runs on; check the <i>web.config</i> file for your web service to verify the version of .NET framework it is tied to.</p> <p>The Debug window <b>Stop</b> button should be enabled and any breakpoints should be red, indicating they have been bound.</p>	

You can set breakpoints at any time in the web server code. You can also set breakpoints in the ASP web page(s) if you imported them.

#### Notes

- Some breakpoints might not have bound successfully, but if none at all are bound (indicated by being dark red with question marks) something has gone out of sync; try rebuilding and re-importing source code

#### 11.8.2.6.5 The PHP Debugger

The Enterprise Architect PHP Debugger enables you to debug PHP.exe scripts. This section discusses basic setup and the various debugging scenarios that are commonly encountered; the scenarios concern themselves with the mapping of file paths, which is critical to the success of a remote debugging session.

- Script Setup
- Local Windows Machine (Apache Server)
- Local Windows Machine (PHP.exe)
- Remote Linux Machine (Apache Server)
- Remote Linux Machine (PHP.exe)

#### Topics

Topic	Detail	See also	
<b>Script Setup</b>	An Analyzer Script is a basic requirement for debugging in Enterprise Architect; you create a script using the toolbar of the Execution Analyzer.		
	Select <b>PHP.XDebug</b> as the debugging platform; when you select this platform the property page displays the following connection settings:		
	Field	Value	Meaning
	port	9000	The port that Enterprise Architect listens on for the
	host	localhost	The adaptor that Enterprise Architect listens on for from PHP.
localpath	%LOCAL%	Specifies the local file path to be mapped to a remote  This is a remote debugging setting; for local debugging	

Topic	Detail	See also									
	<table> <tr> <th>Field</th><th>Value</th><th>Meaning</th></tr> <tr> <td></td><td></td><td>The value is a placeholder; you should edit it to fi</td></tr> <tr> <td><b>remotepath</b></td><td><b>%REMOTE%</b></td><td>Specifies the remote file path that a local file path This is a remote debugging setting; for local deb The value is a placeholder; you should edit it to fi</td></tr> </table>	Field	Value	Meaning			The value is a placeholder; you should edit it to fi	<b>remotepath</b>	<b>%REMOTE%</b>	Specifies the remote file path that a local file path This is a remote debugging setting; for local deb The value is a placeholder; you should edit it to fi	
Field	Value	Meaning									
		The value is a placeholder; you should edit it to fi									
<b>remotepath</b>	<b>%REMOTE%</b>	Specifies the remote file path that a local file path This is a remote debugging setting; for local deb The value is a placeholder; you should edit it to fi									
<b>Local Machine Apache Server</b>	<p>In this situation, consider the following configuration:</p> <ul style="list-style-type: none"> <li>• O/S: Windows7</li> <li>• Network computer name: <i>MyPC</i></li> <li>• Network share MyShare mapped to <i>c:\myshare</i></li> <li>• Source files in Enterprise Architect have been imported from <i>c:\myshare\apache\myapp\scripts</i></li> <li>• Apache document root is set to <i>//MyPC/MyShare/apache</i></li> </ul> <p>In this scenario an Analyzer Script for the connection parameters might be configured as follows:</p> <ul style="list-style-type: none"> <li>• <b>host:</b> localhost</li> <li>• <b>port:</b> 9000</li> <li>• <b>localpath:</b> <i>c:\myshare\apache\</i></li> <li>• <b>remotepath:</b> <i>MyPC/MyShare/apache/</i></li> </ul>										
<b>Local Machine PHP.EXE</b>	<p>In this scenario an Analyzer Script for the connection parameters might be configured as follows, as file paths always map to same physical path:</p> <ul style="list-style-type: none"> <li>• <b>host:</b> localhost</li> <li>• <b>port:</b> 9000</li> <li>• <b>localpath:</b></li> <li>• <b>remotepath:</b></li> </ul>										
<b>Remote Linux Machine Apache Server</b>	<p>In this situation consider the following configuration:</p> <ul style="list-style-type: none"> <li>• Local Machine <ul style="list-style-type: none"> <li>• O/S: Windows7</li> <li>• Source files in Enterprise Architect have been imported from <i>c:\myshare\apache\myapp\scripts</i></li> </ul> </li> <li>• Remote Machine <ul style="list-style-type: none"> <li>• O/S: Linux</li> <li>• Apache document root is set to <i>home/apache/htdocs</i></li> </ul> </li> </ul>										

Topic	Detail	See also
	<ul style="list-style-type: none"> <li>Source files in Apache are located at <i>home/apache/htdocs/myapp/scripts</i></li> </ul> <p>In this scenario an Analyzer Script for the connection parameters might be configured as follows:</p> <ul style="list-style-type: none"> <li><b>host:</b> localhost</li> <li><b>port:</b> 9000</li> <li><b>localpath:</b> <i>c:\myshare\apache\</i></li> <li><b>remotepath:</b> <i>home/apache/htdocs/</i></li> </ul>	
<b>Remote Linux Machine PHP.exe</b>	<p>In this situation consider the following configuration:</p> <ul style="list-style-type: none"> <li>Local Machine <ul style="list-style-type: none"> <li>O/S: Windows7</li> <li>Source files in Enterprise Architect have been imported from <i>c:\myshare\apache\myapp\scripts</i></li> </ul> </li> <li>Remote Machine <ul style="list-style-type: none"> <li>O/S: Linux</li> <li>Source files in Apache located at <i>home/myapp/scripts</i></li> </ul> </li> </ul> <p>In this scenario an Analyzer Script for the connection parameters might be configured as follows:</p> <ul style="list-style-type: none"> <li><b>host:</b> localhost</li> <li><b>port:</b> 9000</li> <li><b>localpath:</b> <i>c:\myshare\apache\</i></li> <li><b>remotepath:</b> <i>home/</i></li> </ul>	

#### Learn more

- [PHP Debugger - System Requirements](#) <sup>[2206]</sup>
- [PHP Debugger Checklist](#) <sup>[2207]</sup>

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Build and Debug | Debug | PHP**
- (Alt+F1) | **Enterprise Architect | Build and Debug | Debug | PHP Samples**

#### **11.8.2.6.5.1 PHP Debugger - System Requirements**

This topic identifies the system requirements and operating systems for the Enterprise Architect PHP debugger.

#### **System Requirements:**

- Enterprise Architect version 9

- PHP version 5.3 or above
- PHP zend extension XDebug 2.1 or above
- For web servers such as Apache, a server version that supports the PHP version

#### **Supported Operating Systems:**

- Client (Enterprise Architect)
  - Microsoft Windows XP and above
  - Linux running Crossover Office
- Server (PHP)
  - Microsoft Windows XP and above
  - Linux

#### **11.8.2.6.5.2 PHP Debugger Checklist**

This topic provides a supplementary checklist and troubleshooting guide for debugging PHP scripts in Enterprise Architect.

#### **Topics**

Topic	Detail		See also
<b>System Requirements</b>	<ul style="list-style-type: none"> <li>• Apache HTTP Web Server version 2.2</li> <li>• PHP version 5.3 or above</li> <li>• XDebug version 2.1.1</li> </ul>		<a href="http://httpd.apache.org/">http://httpd.apache.org/</a> <a href="http://windows.php.net/">http://windows.php.net/</a> <a href="http://www.xdebug.org/download.php">http://www.xdebug.org/download.php</a>
<b>Checklist</b>	<b>Enterprise Architect</b>	<ul style="list-style-type: none"> <li>• The model has an Analyzer Script configured to use the PHP XDebug platform</li> <li>• PHP source code has been imported into the model (for recording and testpoints)</li> <li>• When the PHP XDebug platform is selected from the Analyzer Script dialog, default runtime settings are listed in the <b>Connection</b> field               <ul style="list-style-type: none"> <li>• <code>local path: %LOCAL%</code></li> <li>• <code>remote path: %REMOTE%</code></li> </ul> </li> </ul> <p>Either define local paths for these default variables or edit the script to provide actual paths.</p> <p><b>For example:</b> local source, remote source</p> <p><i>localpath:c:\code samples\vea\php\sample</i>  <i>remotepath:web server/sample</i></p> <p>where:</p> <ul style="list-style-type: none"> <li>• <i>web server</i> is a network or local share</li> <li>• <i>sample</i> is a folder below share</li> </ul>	

Topic	Detail		See also
	<b>PHP</b>	<p>In order to debug PHP scripts in Enterprise Architect, it is a requirement that the PHP is configured properly to load the XDebug extension.</p> <p>Settings similar to those below should be used:</p> <ul style="list-style-type: none"> <li>• [xdebug]</li> <li>• xdebug.extended_info=1</li> <li>• xdebug.idekey=ea</li> <li>• xdebug.remote_enable=1</li> <li>• xdebug.remote_handler=dbgp</li> <li>• xdebug.remote_autostart=1</li> <li>• xdebug.remote_host=X.X.X.X</li> <li>• xdebug.remote_port=9000</li> <li>• xdebug.show_local_vars=1</li> </ul> <p>The IP address X.X.X.X refers to and should match the host specified in the model Analyzer Script.</p> <p>The IP address is the address XDebug connects with and the same address the Enterprise Architect PHP agent listens on.</p>	
	<b>Apache</b>	<p>For debugging using Apache, the following lines should be present in the Apache configuration file, <i>httpd.conf</i>:</p> <pre>LoadModule php5_module "php_home/ php5apache2_2.dll" AddHandler application/x-httpd-php . php PHPLibraryDir "php_home"</pre> <p>where <code>php_home</code> is the PHP installation path (the path where <code>php.ini</code> and <code>apache.dll</code> exist).</p>	
<b>Troubleshooting</b>		<p>To prevent both PHP and Apache timeouts during a debugging session, the settings below might require modification.</p> <p>These settings were used while developing the PHP Debugging agent in Enterprise Architect.</p>	
	<b>PHP</b>	<p><b>File: <code>php.ini</code></b></p> <pre>; EA prevent PHP timeouts when debugging PHP extensions max_execution_time = 0  ; EA prevent web server timeouts when debugging PHP extensions max_input_time = -1  ; EA log errors display_errors = On  ; EA display startup errors display_startup_errors = On</pre>	



Topic	Detail	See also
	<b>Apache</b> <b>File: httpd.conf</b> ; EA prevent timeouts while debugging php extensions Timeout 60000	

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect** | **Build and Debug** | **Debug** | **PHP Samples**
- (Alt+F1) | **Enterprise Architect** | **Build and Debug** | **Debug** | **PHP**

#### 11.8.2.6.6 The GNU Debugger (GDB)

When debugging your applications you can use the **GNU Debugger (GDB)**, which is portable and runs on Unix-like systems such as Linux, as well as on Windows. The GDB works for many programming languages including Ada, Java, C, C++ and Objective-C. Using the GDB, you can debug your applications either locally or remotely.

**Access** **Analyzer** | **Execution Analyzer (Shift+F12)** : **New** (Toolbar icon) > **Debug** > **Platform Analyzer** | **Execution Analyzer (Shift+F12)** : locate and double-click on required script > **Debug** > **Platform**

#### Set up the GNU Debugger

Task	Details	See also												
<b>Set up Script</b>	<p>An Analyzer Script is a basic requirement for debugging in Enterprise Architect; you create a script using the Execution Analyzer toolbar.</p> <p>On the Platform page of the Execution Analyzer Script Editor, in the <b>Debugger</b> field click on the drop-down arrow and select <b>GDB</b>.</p>	<a href="#">Managing Analyzer Scripts</a> <small>2175</small>												
<b>Define Connection Settings</b>	<p>The property panel displays a number of connection settings for which you provide values.</p> <table> <tr> <th>Field</th><th>Value</th><th>Meaning</th></tr> <tr> <td><b>path</b></td><td><b>&lt;path&gt;</b></td><td>The complete file path of the GDB executable; you only specify this if the GDB cannot be found in the system path.</td></tr> <tr> <td><b>source</b></td><td><b>&lt;path&gt;, &lt;path&gt;</b></td><td>The path in which the debugger will search for source files, if they do not reside in the executable directory.</td></tr> <tr> <td><b>remote</b></td><td><b>F</b></td><td>Set for remote debugging; otherwise leave blank.</td></tr> </table>	Field	Value	Meaning	<b>path</b>	<b>&lt;path&gt;</b>	The complete file path of the GDB executable; you only specify this if the GDB cannot be found in the system path.	<b>source</b>	<b>&lt;path&gt;, &lt;path&gt;</b>	The path in which the debugger will search for source files, if they do not reside in the executable directory.	<b>remote</b>	<b>F</b>	Set for remote debugging; otherwise leave blank.	
Field	Value	Meaning												
<b>path</b>	<b>&lt;path&gt;</b>	The complete file path of the GDB executable; you only specify this if the GDB cannot be found in the system path.												
<b>source</b>	<b>&lt;path&gt;, &lt;path&gt;</b>	The path in which the debugger will search for source files, if they do not reside in the executable directory.												
<b>remote</b>	<b>F</b>	Set for remote debugging; otherwise leave blank.												

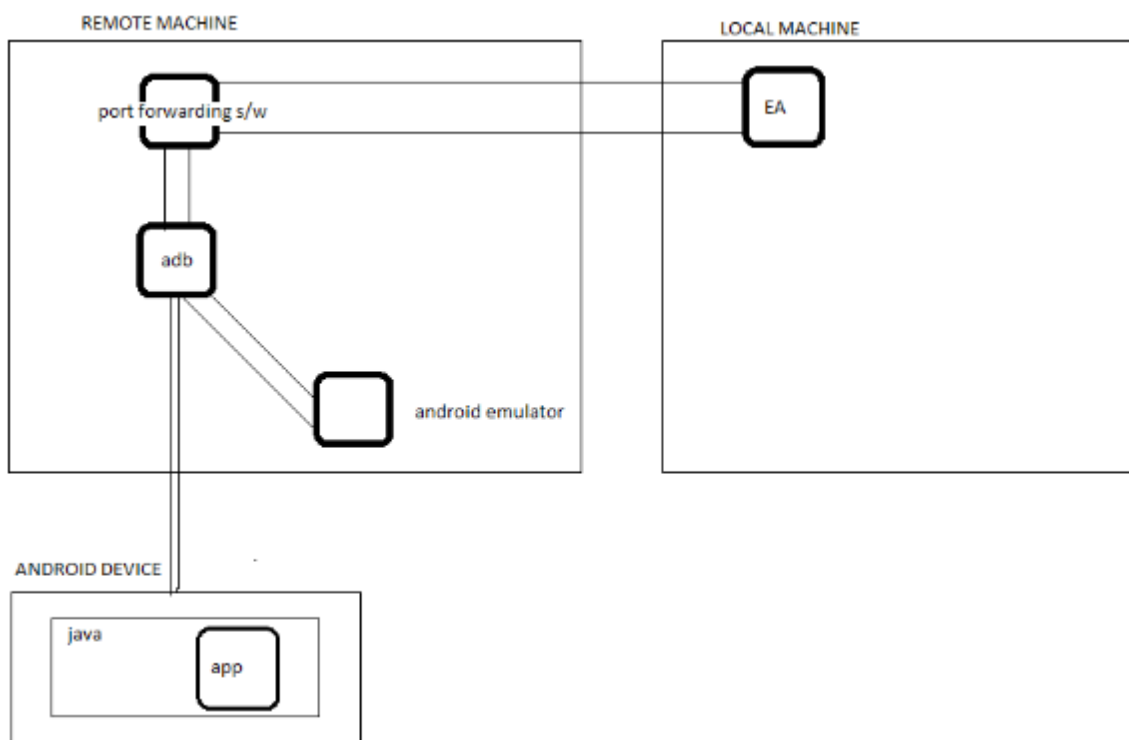
Task	Details			See also
	<b>port</b>	<b>&lt;nnnnn&gt;</b>	The port to connect to on the remote server.	
	<b>host</b>	<b>localhost</b>	The host name to connect to.	
	<b>fetch</b>	<b>T</b>	Set to retrieve the binary from the remote system.	
	<b>dumpgdb</b>	<b>&lt;path&gt;</b>	The filename to write the GDB output to.	
	<b>initpath</b>	<b>&lt;path&gt;</b>	The complete file path to the <i>gbin</i> file.	

#### Notes

- A requirement of the GDB is that your source code file path does not contain spaces; the debugger will not run correctly with spaces in the file path

#### 11.8.2.6.7 The Android Debugger

If you are developing **Java** applications running on **Android** devices or emulators, you can also debug them. The Local and Remote machines can be on either a 32-bit platform or a 64-bit platform.



**System Requirements**


On the Remote machine, this software is required:

- Android SDK, which includes the android debug ridge, ADB (you need to be familiar with the SDK and its tools)
- Java JDK (32 and 64 bit support)
- Port Forwarding software (3rd party)

On the Local machine, this software is required:

- Enterprise Architect Version 10

**Analyzer Script Settings**

Field/Button	Action	See also
Debugger	Click on the drop-down arrow and select <b>Java (JDWP)</b> .	
Run	Click on this radio button.	
Default Directory	Not applicable - leave blank.	
Application path		
Command Line Arguments		
Build first		
Show console		
Show diagnostic messages		
Connection		
Port	This is the application port forward-assigned using adb or other means, that connects to the Virtual Machine (VM) process on Android.	<a href="#">Port Forwarding</a>  2212

Field/Button	Action	See also
<b>Host</b>	<p>Host computer (defaults to localhost)</p> <p>If Android is running on an emulator on a device attached to a networked computer, enter the network name here.</p> <p>By default, debugging will attempt to connect to the port you specify on the local machine.</p>	
<b>Source</b>	<p>This is the source equivalent of the classpath setting in Java.</p> <p>The root to each source tree should be listed. If more than one is specified, they should be separated by a semi-colon; that is:</p> <pre>c:\myapp\src;c:\myserver\src</pre> <p>You must specify at least one root source path.</p> <p>When a breakpoint occurs the debugger searches for the java source in each of the source trees listed here.</p>	
<b>Logging</b>	<p>Enables logging additional information from debugger</p> <p>possible values: true,false,1,0,yes,no</p>	
<b>Output</b>	<p>Specifies the full name of the local log file to be written.</p> <p>The folder must exist or no log will be created.</p> <p>The log file typically contains a dump of bytes sent between debugger and VM.</p>	
<b>Platform</b>	<p>If you are debugging Java running under any android scenario, select <b>Android</b>.</p> <p>For all other scenarios, select <b>Java</b>.</p>	

### Configure Ports for Debugging - Port Forwarding (Local)

The debugger can only debug one VM at a time; it uses a single port for communication with the VM. The port for the application to be debugged can be assigned using ADB.

Before debugging, start the application once in the device. When the app starts, perform the following to discover its process id:

```
adb j dwp
```

The last number listed is the process identifier (pid) of the last application launched; note the pid and use it as follows to allow the debugger to connect to the VM:

- `adb forward tcp:port j dwp:pid`
  - `port` = port number listed in analyzer script
  - `pid` = process id of the application on the device

### Configure Ports for Debugging - Port Forwarding (Remote)

To debug remotely the same procedure should be followed as for the local machine, but the communication requires additional forwarding as the socket created using the `abd forward` command above will only listen on the local adapter. The socket is bound to the localhost and attempts to connect to this port will be met with connection refused messages.

In order to achieve remote debugging it is necessary to have a proxy running on the remote machine that listens to all incoming connections and forwards all traffic to the adb port.; there are numerous software products available to do this.

Remote debugging with Enterprise Architect will not work unless a proxy port forwarder has been configured by the user.

#### **11.8.2.6.8 Java JDWP Debugger**

If you are developing **Java** applications running on either 32-bit or 64-bit virtual machines, you can debug them either locally or remotely using the Java Debug Wire Protocol (JDWP).

### System Requirements

On the Remote machine, this software is required:

- Java JDK (32-bit and 64-bit support)

On the Local machine, this software is required:

- Enterprise Architect Version 10

### Analyzer Script Settings

Field/Button	Action	See also
<b>Debugger</b>	Click on the drop-down arrow and select <b>Java (JDWP)</b> .	
<b>Run</b>	Click on this radio button to run the debugger when the script is executed.	
<b>Default Directory</b>	Not applicable - leave blank.	
<b>Application path</b>		
<b>Command Line Arguments</b>		

Field/Button	Action	See also
Build first		
Show console		
Show diagnostic messages		
Connection		
Port	Set the application port forward-assigned to the VM process during start-up, in the Java command-line options.	<a href="#">Java Command Line Options</a>
Host	Set the host computer (defaults to <b>localhost</b> )  If VM is running on a networked computer, enter the network name or url here.  By default debugging will attempt to connect to the port you specify on the local machine.	
Source	This is the source equivalent of the <code>classpath</code> setting in Java.  List the root to each source tree; specify at least one root source path. If you specify more than one, separate them with a semi-colon; for example:  c: \ my app \ s r c ; c: \ m y s e r v e r \ s r c  When a breakpoint occurs the debugger searches for the Java source in each of the source trees listed here.	
Logging	Enable or disable logging of additional information from the debugger.  Possible values include: <ul style="list-style-type: none"><li>• <b>true</b></li><li>• <b>false</b></li><li>• <b>1</b></li><li>• <b>0</b></li><li>• <b>yes</b></li><li>• <b>no</b></li></ul>	
Output	Specify the full name of the local log file to be written. If the folder does not already exist, no log will be created.	

Field/Button	Action	See also
	The log file typically contains a dump of bytes sent between the debugger and VM.	
<b>Platform</b>	Select <b>Java</b> .	

### Configure Ports for Debugging

The debugger can only debug one VM at a time; it uses a single port for communication with the VM. The port for the application to be debugged is assigned when the VM is created.

### Local Debugging

Where both Enterprise Architect and the Java VM are running on the same machine, you can perform local debugging. It is necessary to launch the VM with the JDWP transport enabled - see the documentation on Java Platform Debugger Architecture (JPDA) at Oracle for the command line option specifications.

For example:

```
java -agentlib:jdwp=transport=dt_socket,address=localhost:9000,server=y,
suspend=n -cp "c:\samples\java\myapp;%classpath%" samples.MyApp "param1"
"param2"
```

In this example the values for the Analyzer script are **host = localhost** and **port = 9000**.

### Remote Debugging

Where Enterprise Architect is running on the local machine and the Java VM is running on a remote machine, you can perform remote debugging. It is necessary to launch the VM with the JDWP transport enabled - see the documentation on JPDA at Oracle for the command line option specifications.

For example (where the remote computer has the network name **test01**) :

```
java -agentlib:jdwp=transport=dt_socket,address=9000,server=y,suspend=n
-cp "c:\samples\java\myapp;%classpath%" samples.MyApp "param1" "param2"
```

No host name is defined in the **address**, so VM will listen for any connection on any adaptor. In this example the values for the Analyzer script are **host = test01** and **port = 9000**.

#### 11.8.2.6.9 Tracepoint Output

The Tracepoints page of the Analyzer Script enables you to direct where the output from any Trace statements goes during a debug session.

**Access** **Analyzer | Execution Analyzer (Shift+F12) : New** (Toolbar icon) > **Debug > Tracepoints**  
**Analyzer | Execution Analyzer (Shift+F12) : locate and double-click on required script > Debug > Tracepoints**

Field	Detail	See also
<b>Output</b>	You can select from two options: <ul style="list-style-type: none"> <li>• <b>Screen</b> (Default) - The output is directed to the Debug window</li> <li>• <b>File</b> - The output is directed to file</li> </ul>	
<b>Folder</b>	Enter the folder to use for Trace statement log files.	
<b>Filename</b>	Enter the name to use for the Trace statement log files.	
<b>Overwrite</b>	If selected, the specified file is overwritten each time a debug session is started.	
<b>Auto Number</b>	If selected, the Trace log file is composed of the filename you specify and a number.  Each time you start a debug session, the number is incremented.	
<b>Prefix trace output with function</b>	If selected, any Trace statements executed during the debug session run are prefixed with the current function call.	

#### 11.8.2.6.10 Workbench Setup

This topic describes the requirements for setting up the Object Workbench on Java and Microsoft .NET.

**Access** [Analyzer](#) | [Execution Analyzer](#) | [double-click on script](#) | [Debug](#) | [Workbench](#)

#### Topics

Topic	Detail	See also
<b>Platforms Supported</b>	The Workbench supports the following workbench platforms: <ul style="list-style-type: none"> <li>• Microsoft .NET (version 2.0 or later) excluding native C++, C and VB</li> <li>• Java (JDK 1.4 or later)</li> </ul> <p>The Execution Analysis script for the package must have a debugger configured.</p>	<a href="#">Set up a Debug script</a> <small>2185</small>
<b>Microsoft .NET Workbench</b>	The .NET workbench requires an assembly, which is used to create the workbench items.	<a href="#">General Setup for.</a>



Topic	Detail	See also
	<p>You specify the path to the assembly on the Workbench page of the Analyzer Script.</p> <p>There are two constraints in using the .NET workbench:</p> <ul style="list-style-type: none"> <li>Members defined as <i>struct</i> in managed code are not supported</li> <li>Classes defined as <i>internal</i> are not supported</li> </ul>	<a href="#">NET</a> <sup>[2200]</sup>
<b>Java Workbench</b>	The Java workbench uses the Virtual Machine settings configured in the Analyzer Script Debug page to create the JVM.	<a href="#">General Setup for Java</a> <sup>[2191]</sup>

Learn more

- [How it Works](#) <sup>[2567]</sup>
- [Create & Delete Workbench Instances](#) <sup>[2568]</sup>

**11.8.2.7 Add Run Command**

This topic describes how to create a **command for running** your executable code.

**Access** **Analyzer | Execution Analyzer (Shift+F12) : New** (Toolbar icon) > **Run Analyzer | Execution Analyzer (Shift+F12) : locate and double-click on required script > Run**

Topics

Topic	Detail	See also
<b>Usage</b>	This is the command that is executed when you select the <b>Analyzer   Run</b> menu option; at its simplest, the script would contain the location and name of the file to be run.	
<b>Example</b>	<p>The following two examples show scripts configured to run a .Net and a Java application in Enterprise Architect.</p> <p><b>.Net:</b></p> <pre>C: \ benchmark \ cpp \ example_net _1 \ release \ example.exe</pre> <p><b>Java:</b></p> <pre>customer</pre> <p>The command listed in this field is executed as if from the command prompt; as a result, if the executable path or any arguments contain spaces, they must be surrounded in quotes.</p>	

Notes

- Enterprise Architect provides the ability to start your application normally OR with debugging from the same script; the **Analyzer** menu has separate options for starting a normal run and a debug run

Learning Center topics

- (Alt+F1) | **Enterprise Architect** | **Build and Debug** | **Overview** | **Run**

**11.8.2.8 Add Deploy Command**

This topic explains how to how to create a **command for deploying** the current package.

**Access** **Analyzer** | **Execution Analyzer (Shift+F12)** : **New** (Toolbar icon) > **Deploy**  
**Analyzer** | **Execution Analyzer (Shift+F12)** : locate and double-click on required script > **Deploy**

Topics

Topic	Detail	See also
<b>Usage</b>	You create a script that is executed when you select the <b>Analyzer</b>   <b>Deploy</b> menu option or when you press <b>Ctrl+Shift+Alt+F12</b> .	
<b>Execute Command as:</b>	<p><b>Process</b></p> <p>If the deployment is handled externally, enter the path to the program or batch file to run followed by any parameters; the program is launched in a separate process.</p> <p>Example:</p> <pre>C:\apache-ant-1.7.1\bin\ant.cmd myproject deploy</pre> <p><b>Batch File</b></p> <p>When using this option, you can enter multiple commands that are then executed as a single script in a command console; you have access to any environment variables available in a standard command console.</p> <p>Example:</p> <pre>@echo on IF NOT EXIST "%1%" GOTO DEPLOY_NOWAR IF "%APACHE_HOME%" == "" GOTO DEPLOY_NOAPACHE xcopy /L "%1%" "%APACHE_HOME%\webapps" GOTO DEPLOY_END rem rem NO WAR FILE rem :DEPLOY_NOWAR echo "%1% WAR file not found" GOTO DEPLOY_END rem rem NO APACHE ENVIRONMENT VARIABLE rem :DEPLOY_NOAPACHE echo "APACHE_HOME environment variable not found"</pre>	

Topic	Detail	See also
	<pre>:DEPLOY_END pause</pre>	
<b>Parse Output</b>	<p>Selecting a Parser from the list causes output of the deploy script to be captured; the output is parsed according to the syntax selected from the list.</p> <p>The output window can be displayed by selecting <b>View   System Output</b>.</p>	

### 11.8.2.9 Configure Recording Detail

To avoid generating too complex a Sequence diagram, you must control the scope of what is being recorded by:

- Filtering - using the Analyzer Script you can exclude specific modules, Classes and functions from the recording

**Access** **Analyzer | Execution Analyzer (Shift+F12)** **Double-click on required Analyzer Script > Recording**

#### Topics

Topic	Detail	See also
<b>Filtering</b>	<p>If the <b>Enable Filter</b> checkbox is selected on the Recording page of the Execution Analyzer Script Editor, the debugger excludes calls to matching methods from the generated Sequence history and diagram; the comparison is case-sensitive.</p> <p>To add a value, click on the <b>New (Insert)</b> icon in the right corner of the Exclusion Filters box, and type in the comparison string; each filter string takes the form:</p> <pre>class_name_token::method_name_token</pre> <p>The <i>class_name_token</i> excludes calls to all methods of a Class or Classes that have a name matching the token; the string can contain the wildcard character * (asterisk).</p> <p>The <i>method_name_token</i> excludes calls to methods having a name that matches the token; again, the string can contain the wildcard character *.</p> <p>Both tokens are optional; if no Class token is present, the filter is applied only to global or public functions (that is, methods not belonging to any Class).</p>	<a href="#">Managing Analyzer Scripts</a> <sup>[2175]</sup> <a href="#">Analyzer Script Editor</a> <sup>[2176]</sup>
<b>Example</b>	<p>In the Java example below, the debugger would exclude:</p> <ul style="list-style-type: none"> <li>• Calls to the <i>OnDraw</i> method for the Class <i>Example.common.draw.DrawPane</i></li> <li>• Calls to any method of any Class having a name beginning with <i>Example.source.Collection</i></li> </ul>	

Topic	Detail	See also
	<ul style="list-style-type: none"> <li>• Calls to any constructor for any Class (such as <i>&lt;clint&gt;</i> and <i>&lt;init&gt;</i>)</li> </ul> <div> Filters <pre> Example.common.draw.DrawPane::OnDraw Example.source.Collection* *::init* </pre> </div> <p>In the Native Code example below, the debugger would exclude:</p> <ul style="list-style-type: none"> <li>• Calls made to Standard Template Library namespace</li> <li>• Calls to any Class beginning with <i>TOb</i></li> <li>• Calls to any method of Class <i>CLock</i></li> <li>• Calls to any Global or Public Function with a name beginning with <i>Get</i></li> <li>• Calls to the method <i>GetLocation</i> for Class <i>Ctrain</i></li> </ul> <div> Filters <pre> std* TOb* CLock CTrain::GetLocation ::Get* </pre> </div>	

### Reference

To Filter	Use Filter Entry
All public functions having a name beginning with <b>Get</b> from the recording session (for example, <i>GetClientRect</i> in Windows API).	<code>:: Get *</code>
All methods beginning with <b>Get</b> for every Class member method.	<code>* :: Get *</code>
All methods beginning with <b>Get</b> from the Class <i>CClass</i> .	<code>CCl ass :: Get *</code>
All methods for Class <i>CClass</i> .	<code>CCl ass :: *</code>
All methods for Classes belonging to Standard Template and Active Template Libraries.	<code>ATL * st d *</code>
The specific method <i>GetName</i> for Class <i>CClass</i> .	<code>CCl ass :: Get Name</code>



### Learn more

- [Control Stack Depth](#)<sup>[2536]</sup>

## 11.8.3 Build Application

This topic explains how to execute a Build script on your application, within Enterprise Architect.

**Access** **Analyzer | Execution Analyzer (Shift+F12) : select required script, then:**

- 
- **(Ctrl+Shift+F12)**
- **Analyzer | Build**
- Right-click and select either **Build** or **Rebuild** context menu option, **or**
- **Analyzer | Debugger:**  **> Build**

### Guide

When you select the **Build** option, it executes the Build command in the script selected in the Execution Analyzer window. The progress and outcome of the build operation are displayed in the Build tab of the Output window.

Locate any compilation errors and correct the code, and repeat the build-correct cycle until the compilation log on the Output window Build tab shows no compile errors.

### Learn more

- [Add Build Commands](#)<sup>[2180]</sup>
- [Analyzer Script Editor](#)<sup>[2179]</sup>
- [Locate Compiler Errors in Code](#)<sup>[2221]</sup>

### 11.8.3.1 Locate Compiler Errors in Code

When you build an application using a Build script, the compiler output is logged in the System Output window. The output includes any compilation error messages that were generated.

**Access** **View | System Output (Ctrl+Shift+8)**

If you double-click on a compiler error message, the system loads the appropriate source file and positions the cursor on the line in the code at which the error was reported, as illustrated below:

```

61
62 if(PeopleOFF > 0)
63 re turn PeopleOFF * 20;
64
65 return 0;

```

**Output**

```

Running Analyzer Script - CityLoop
Microsoft (R) Visual Studio Version 10.0.30319.1.
Copyright (C) Microsoft Corp. All rights reserved.
1>----- Build started: Project: CityLoop, Configuration: Debug Win32 -----
1> Train.cpp
1> c:\data\vea\microsoft native\cityloop\train.cpp(63): error C2065: 're' : undeclared identifier
1> c:\data\vea\microsoft native\cityloop\train.cpp(63): error C2146: syntax error : missing ';' before identifier 'turn'
1> c:\data\vea\microsoft native\cityloop\train.cpp(63): error C2065: 'turn' : undeclared identifier

```

#### Learn more

- [Build Application](#)<sup>[2221]</sup>
- [The Output Window](#)<sup>[169]</sup>

### 11.8.4 Debugging

This section describes how you define and execute the debugging actions:

- [Before Starting](#)<sup>[2222]</sup>
- [Set up a Debug script](#)<sup>[2185]</sup>
- [Breakpoint and Marker Management](#)<sup>[2224]</sup>
- [Debugger Facilities](#)<sup>[2231]</sup>

#### Notes

- You can register the Enterprise Architect debugger as the operating system just-in-time debugger, to be invoked when an application running outside Enterprise Architect on the system either encounters an exception or crashes

#### Learn more

- [Debug Menu](#)<sup>[114]</sup>

#### 11.8.4.1 Before Starting

The Model Driven Development Environment provides Debuggers for the following frameworks:

- Microsoft Native Code applications
- Microsoft .NET applications
- Java applications
- PHP applications
- Scripting languages: JavaScript (Mozilla), VBScript and JScript (Microsoft)

#### Checklist

Step	Action	See also
1	A model must be open.	
2	Ensure any source code for the areas of interest has been generated or imported into the model, synchronized with the model and, if necessary, compiled.	<a href="#">Generate Source Code</a> <sup>[2117]</sup> <a href="#">Importing Source Code</a> <sup>[2136]</sup> <a href="#">Update Package Contents</a> <sup>[2117]</sup>

### Option 1 - No Analyzer Script

If you have *not* configured an Analyzer Script, you can still debug a process that is running.

Step	Action	See also
1	Optionally set breakpoints before debugging.	<a href="#">Breakpoint and Marker Management</a> <sup>[2224]</sup>
2	Use the <b>Attach</b> button on the Debug toolbar <ul style="list-style-type: none"> <li>Select the debugging platform</li> <li>Select the process to debug</li> </ul>	

### Option 2 - Analyzer Script, with Debug Page Configured

Step	Action	See also
1	Ensure the application has been built.  You can do this internally using a Build Script, or you can build the application externally; the important thing is that the application is built on the latest versions of the source.	<a href="#">Build Application</a> <sup>[2227]</sup> <a href="#">Add Build Commands</a> <sup>[2180]</sup> (script)
2	Create an appropriate debug script on the Debug page of the Analyzer Script Editor.	<a href="#">Set up a Debug script</a> <sup>[2185]</sup>
3	Set breakpoints in the code.	<a href="#">Set Code Breakpoints</a> <sup>[2226]</sup>
4	Start the debugger.	<a href="#">Run the Debugger</a> <sup>[2237]</sup>



### 11.8.4.2 Breakpoint and Marker Management

Breakpoints work in Enterprise Architect much like in any other debugger. Adding a breakpoint notifies the debugger to trap code execution at the point you have specified. When a breakpoint is encountered by a thread of the application being debugged, the source code is displayed in the Source Code Editor, and the line of code where the breakpoint occurred is highlighted.




Selecting a different package in the project affects which breakpoints are displayed. An Enterprise Architect model maintains breakpoints for every package having a *Build Script - Debug* command.

Access    **Analyzer | Breakpoints & Events**





#### Breakpoint and Marker Options

Option	Detail	See also
<b>Delete a breakpoint or marker</b>	<p>To delete a specific breakpoint:</p> <ul style="list-style-type: none"> <li>• If the breakpoint is enabled, click on the red breakpoint circle in the left margin of the Source Code Editor, or</li> <li>• Right-click on the breakpoint or marker in the Source Code Editor, the <i>Breakpoints</i> folder or the Breakpoints &amp; Events window and select the <b>Delete</b> context menu option, or</li> <li>• Select the breakpoint in the Debug Breakpoints tab and press ( <b>Delete</b> )</li> </ul>	<a href="#">Setting Code Breakpoints</a> <small>[2226]</small>
<b>Delete all breakpoints</b>	Click on the <b>Delete all breakpoints</b> button on the Breakpoints & Events window toolbar (  ).	
<b>Convert breakpoint to Start Recording marker or End Recording marker</b>	<p>(For Visual Execution Analysis, when recording execution to generate a Sequence diagram).</p> <p>In the <i>Breakpoints</i> folder or the Breakpoints &amp; Events window, right-click on the breakpoint and select the context menu option to convert it to either a <b>Start Recording</b> marker or an <b>End Recording</b> marker.</p>	<a href="#">Recording Sequence Diagrams</a> <small>[2531]</small>
<b>Disable a breakpoint</b>	<p>Deselect the checkbox against the breakpoint or marker, on the Breakpoints &amp; Events window.</p> <p>The breakpoint is then shown as an empty grey circle.</p>	
<b>Enable a breakpoint or marker</b>	Select the checkbox against the breakpoint or marker, on the Breakpoints & Events window.	
<b>Disable all breakpoints</b>	Click on the <b>Disable all breakpoints</b> button in the Breakpoints & Events window toolbar (  ).	
<b>Enable all</b>	Click on the <b>Enable all breakpoints</b> button in the Breakpoints & Events	



Option	Detail	See also
<b>breakpoints</b>	window toolbar (  ).	
<b>Change the stack depth</b>	Set the number of stack levels to record to, relative to the level at which recording starts.	<a href="#">Control Stack Depth</a> <sup>[2536]</sup>
<b>Identify or change the marker set</b>	<p>Check the <b>Default</b>  field in the Breakpoints &amp; Events window toolbar.</p> <p>If necessary, click on the drop down arrow and select a different marker set.</p> <p>The Default set is normally used for debugging and is personal to your user ID; other marker sets are shared between all users within the model.</p>	<a href="#">Working with Marker Sets</a> <sup>[2541]</sup>
<b>Change how breakpoints and markers are grouped on the Breakpoints &amp; Events window</b>	<p>The breakpoints and markers can be grouped by Class or by code file.</p> <p>To group the items, click on the down arrow on the  icon in the toolbar, and click on the appropriate option.</p> <p>If you do not want to group the items, click on the selected option to deselect it; the breakpoints and markers are line listed by line number.</p>	

### Breakpoint States

DEBUGGER STATE			
	Running	Not running	See also
	Bound.	Enabled	
	Disabled.	Disabled	
	Not bound - this usually means that the DLL is not yet loaded or was not built with debug information.	N/a	
	Failed - this usually means a break could not be set at this time.	N/a	<a href="#">Failure to Bind Breakpoint</a> <sup>[2229]</sup>

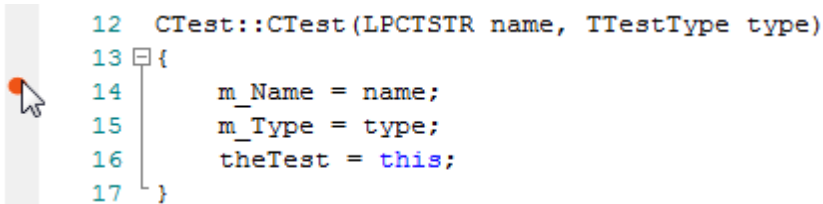
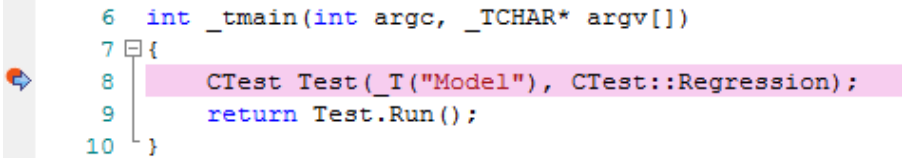
### Learn more

- [Marker Types](#) <sup>[2538]</sup>
- [Break When a Variable Changes Value](#) <sup>[2229]</sup>
- [Trace When Variable Changes Value](#) <sup>[2230]</sup>

### 11.8.4.2.1 Setting Code Breakpoints

Normal Breakpoints are typically set on a line of source code. When the Debugger hits the indicated line during normal execution, the Debugger halts execution and displays the local variables, call stack, threads and other run-time information.

#### Set a breakpoint on a line of code

Step	Action	See also
1	Open the source code to debug in the integrated source code editor.	
2	<p>Find the appropriate code line and click in the left margin column - a solid red circle in the margin indicates that a breakpoint has been set at that position.</p>  <p>If the code is currently halted at a breakpoint, that point is indicated by a blue arrow next to the marker.</p>  <p>Alternatively, you can set the Breakpoint marker (or other marker) by right-clicking on the left margin on the required line, to display the breakpoint/marker context menu; select the appropriate marker type.</p>	<a href="#">Set Record Markers</a> <sup>2537</sup>

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Build and Debug | Debug | Add Breakpoint**

### 11.8.4.2.2 Breakpoint Properties

Breakpoints have a number of additional properties that change the point at which the breakpoint is hit, and determine what happens at that point. These properties define:

- The action to be performed
- The line of code that the breakpoint applies to
- Constraints to restrict when the breakpoint is hit
- Trace information to be output when the breakpoint is hit

**Access**   **Breakpoint Context Menu | Properties**

Options

Field	Details	See also
<b>Action</b>	The behavior when the breakpoint is hit.	<a href="#">Marker Types</a> [2538]
<b>Line</b>	The line of source code that this breakpoint applies to.	
<b>Stack Height</b>	For Stack Capture markers, the number of caller frames to record. To record the entire Stack, set the value to <b>0</b> .	
<b>Constraints</b>	Defines a condition required for this breakpoint to be hit.  The program is allowed to continue normal execution if this condition is <b>not</b> met when the breakpoint is encountered.	<a href="#">Constraint Composition</a> [2581]
<b>Trace statement</b>	A message output to the Debug window when the breakpoint is hit. Variables currently in scope can be included in a trace statement output by prefixing the variable name with a <b>\$</b> token for string variables, or a <b>@</b> token for primitive types such as int or long.	<a href="#">Specifying a Trace Statement</a> [2228]  <a href="#">Run the Debugger</a> [2231]

Learning Center topics

- (Alt+F1) | **Enterprise Architect | Build and Debug | Debug | Breakpoint Properties**

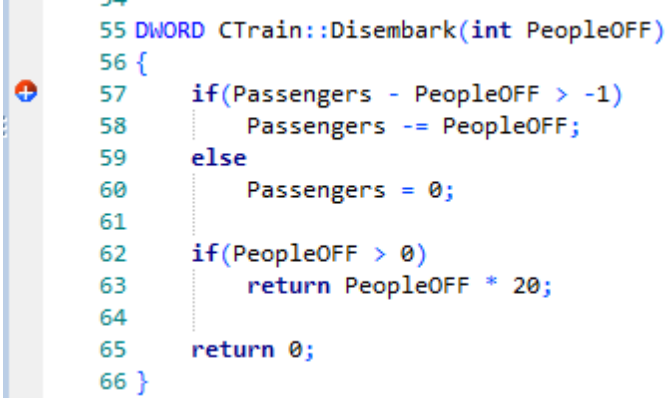
**11.8.4.2.3 Trace Statements**

A **Trace Statement** is a message that is output during execution of a debug session. Trace statements can be defined in Enterprise Architect without requiring any changes to your application source code.

**Tracepoint Markers** are set in the code editor. Like breakpoints, they are placed on a line of code. When that line of code executes, the Execution Analyzer evaluates and prints the statement either to the Debug window or to file, as configured in the Analyzer script.

Add a Tracepoint Marker

Step	Action	See also
<b>1</b>	Open the source code to debug in the source code editor.	<a href="#">Editing Source Code</a> [2146]
<b>2</b>	Find the appropriate code line, right-click in the left margin and select the <b>Add Tracepoint Marker</b> option. The <b>Breakpoint Properties</b> window is displayed.	<a href="#">Breakpoint Properties</a> [2226]

Step	Action	See also
3	Ensure the <b>Trace statement</b> checkbox is selected.	
4	In the text field below the <b>Trace statement</b> checkbox, type the required Trace statement.	<a href="#">Specifying a Trace Statement</a> <sup>[2228]</sup>
5	Click <b>OK</b> . A <b>Tracepoint Marker</b> is shown in the left margin of the code editor.   <pre> 55 DWORD CTrain::Disembark(int PeopleOFF) 56 { 57     if(Passengers - PeopleOFF &gt; -1) 58         Passengers -= PeopleOFF; 59     else 60         Passengers = 0; 61 62     if(PeopleOFF &gt; 0) 63         return PeopleOFF * 20; 64 65     return 0; 66 } </pre>	

### Specifying a Trace Statement

A trace statement can be any freeform text. The value of any variables currently in scope can also be included in a trace statement by prefixing the variable name with a special token. The available tokens are:

- **\$** - when the variable is to be interpreted or printed as a string
- **@** - when the variable is a primitive type (int, double, char)

Using the example in the image above, we could output the number of people getting off a train by using the following statement:

```
There were @Passengers before @PeopleOFF got off the train at $Arriving.
Name Station
```

### Notes

- Trace statements can be included on any type of breakpoint or marker.

### Learn more

- [Testpoint Management](#)<sup>[2578]</sup>

### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Build and Debug | Debug | Add Tracepoint**

#### 11.8.4.2.4 Failure to Bind Breakpoint

##### Topics

Topic	Detail	See also
<b>Breakpoint Failures</b>	<p>A breakpoint failure occurs if there is a problem in binding the breakpoint.</p> <p>A warning message displays in the <b>Details</b> column of the Breakpoints &amp; Events window, identifying the type of problem</p> <ul style="list-style-type: none"> <li>The source file for the breakpoint does not match the source file used to build the application image</li> <li>The time date stamp on the file is greater than that of the image</li> </ul> <p>A warning message is also output to the Debug window.</p>	<a href="#">Run the Debugger</a> <sup>[2231]</sup>

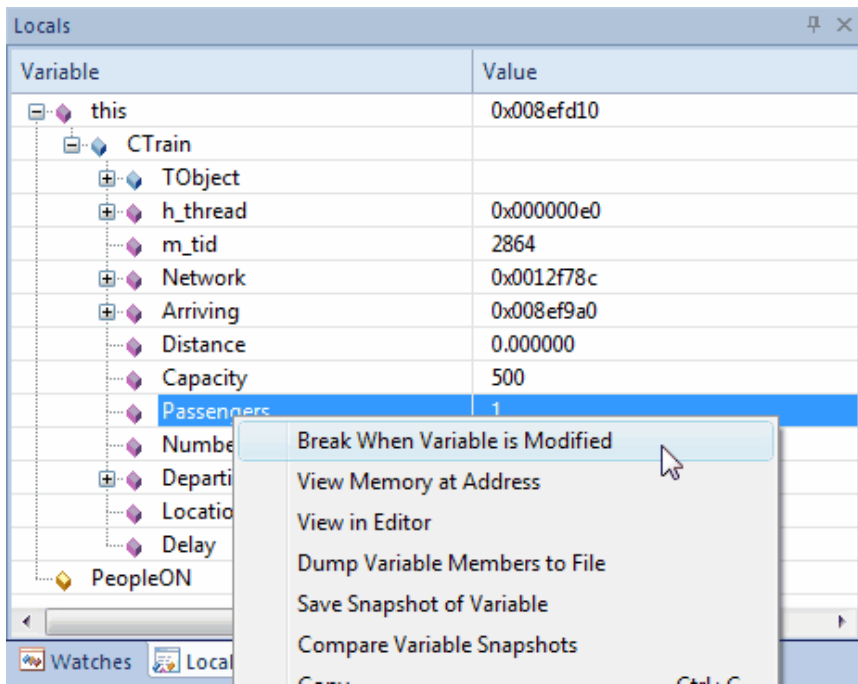
#### 11.8.4.2.5 Break When a Variable Changes Value

Data breakpoints can be set on a pre-determined memory variable to cause the debugger to halt execution at the line of code that has just caused the value of the variable to change. This can be useful when trying to track down the point at which a variable is modified during program execution, especially if it is not clear how program execution is affecting a particular object state.

**Access**    **Analyzer | Locals: right-click on variable | Break When Variable is Modified, or**  
**Analyzer | Watches: right-click on variable | Break When Variable is Modified**

##### Set a data breakpoint

Steps	Detail	See also
<b>1</b>	First set a <b>normal breakpoint</b> at a point in the code that presents the required scope of local variables to choose from.	<a href="#">Setting Code Breakpoints</a> <sup>[2226]</sup>
<b>2</b>	<p>Press <b>F6</b> to execute the debugger; once the program has halted execution, right-click on the variable of interest and select the <b>Break When Variable is Modified</b> context menu option.</p> <p>This example creates a notification on the <b>Passengers</b> integer member of the <i>CTrain</i> Class, in the Locals window.</p>	<a href="#">View the Local Variables</a> <sup>[2234]</sup>

Steps	Detail	See also
		
3	<p>There is no breakpoint indicator in the code, but in the Breakpoints &amp; Events window, the data breakpoint displays as a blue circle in the <b>Enabled</b> column.</p> <p>Continue to execute the debugger; when the variable changes and the debugger halts, the line of code is highlighted in the Source Code Editor and the log in the Debug window prints a statement reporting the effect.</p>	<a href="#">View Variables in Other Scopes</a> <sup>2237</sup>

### Notes

- This feature is not presently supported by the Microsoft .NET platform

### Learning Center topics

- (Alt+F1) | [Enterprise Architect](#) | [Build and Debug](#) | [Debug](#) | [Add Data Breakpoint](#)

#### 11.8.4.2.6 Trace When Variable Changes Value

When your code executes, it might change the value of a variable. It is possible to capture such changes and the variable's new value, on the Debug window. You can then double-click on the change record to display the line of code that caused the change, in the Code Editor.

**Access**    [Right-click on variable](#) | [Trace When Variable Modified](#)

### Set up Trace

The variable you are tracing has to be in scope, so to identify and select it set a normal breakpoint on the line of code where you know that the variable will exist. When the debugger reaches this breakpoint, check either

the Locals window or the Watches window (if examining public and static scopes) and locate the variable. Right-click on it and select the menu option.

Now disable the normal breakpoint and run the program. Each time the variable changes value, it will be logged to the output tab of the debugger. Check the change in value and double-click on the line to display the code in the Code Editor.

### Notes

- The debugger does not halt when the change event occurs, it only logs the change
- This facility is available on the Microsoft Native and Java platforms
- Microsoft .NET does not support breakpoints on values

### Learn more

- [Setting Code Breakpoints](#) <sup>[2226]</sup>
- [Run the Debugger](#) <sup>[2231]</sup>

## 11.8.4.3 Debugger Facilities

This section describes the facilities available while running a debug session. It covers:


- [Run the Debugger](#) <sup>[2231]</sup>
- [View the Local Variables](#) <sup>[2234]</sup>
- [View Variables in Other Scopes](#) <sup>[2237]</sup>
- [View the Call Stack](#) <sup>[2239]</sup> (and [Create Sequence Diagram of Call Stack](#)) <sup>[2241]</sup>
- [Inspect Process Memory](#) <sup>[2242]</sup>
- [Show Loaded Modules](#) <sup>[2243]</sup>
- [Debug Another Process](#) <sup>[2243]</sup>
- [Script Debugging](#) <sup>[2801]</sup>
- [Process First Chance Exceptions](#) <sup>[2244]</sup>







### 11.8.4.3.1 Run the Debugger

The debugger provides a number of ways to start and control a debug session. Many of these facilities are also available through the Debugger toolbar (**View | Toolbars | Debug**).





**Access** **Analyzer | Debugger** (Alt+8)  
**Analyzer Function keys**

### Using the Debug window

Action	Detail	See also
<b>Start the Debugger</b>	<p>If you have configured an Analyzer script for your target application and you have made it the default script, you can start debugging the application immediately in the following ways:</p> <ul style="list-style-type: none"> <li>• Click on the  button</li> </ul>	<p><a href="#">Before Starting</a> <sup>[2227]</sup></p> <p><a href="#">Analyzer Script Editor</a> <sup>[2179]</sup></p>

Action	Detail	See also
	<ul style="list-style-type: none"> <li>Press ( <b>F6</b> ), or</li> <li>Select the <b>Analyzer   Debug   Debug Run</b> menu option</li> </ul> <p>If you have not made your target application the default script it is possible to start any Execution Analysis script by right clicking on it in the Execution Analysis window and selecting the <b>Debug</b> option.</p> <p>If you have no Analyzer Script, it is still possible to debug a running application by attaching to that process directly:</p> <ul style="list-style-type: none"> <li>Click on the  (<b>Attach</b>) button and choose the debugging platform to use</li> </ul>	<a href="#">Debug Another Process</a> <sup>[2243]</sup>
<b>Pause/Resume Debugging</b>	To pause a debugging session, or to resume the session after pausing, click on the  button.	
<b>Stop the Debugger</b>	<p>To stop debugging, either:</p> <ul style="list-style-type: none"> <li>Click on the  (<b>Stop</b>) button</li> <li>Press ( <b>Ctrl+Alt+F6</b> ), or</li> <li>Select the <b>Analyzer   Debug   Debug Stop</b> menu option</li> </ul> <p>The debugger normally ends when the current debug process terminates, however some applications and services (e.g. Java Virtual Machine) may require the debugger to be manually <b>Stopped</b>.</p>	<a href="#">Breakpoint and Marker Management</a> <sup>[2224]</sup>
<b>Step Over Lines of Code</b>	<p>To step over the next line of code:</p> <ul style="list-style-type: none"> <li>Click on the  (<b>Step Over</b>) button</li> <li>Press ( <b>Alt+F6</b> ), or</li> <li>Select the <b>Analyzer   Debug   Step Over</b> menu option</li> </ul>	
<b>Step Into Function Calls</b>	<p>To step into a function call:</p> <ul style="list-style-type: none"> <li>Click on the  (<b>Step In</b>) button</li> <li>Press ( <b>Shift+F6</b> ), or</li> <li>Select the <b>Analyzer   Debug   Step In</b> menu option</li> </ul> <p>If no source is available for the target function then the debugger returns immediately to the caller.</p>	
<b>Step Out Of Functions</b>	<p>To step out of a function:</p> <ul style="list-style-type: none"> <li>Click on the  (<b>Step Out</b>) button</li> <li>Press ( <b>Ctrl+F6</b> ), or</li> </ul>	



Action	Detail	See also
	<ul style="list-style-type: none"> <li>Select the <b>Analyzer   Debug   Step Out</b> menu option</li> </ul> <p>If the debugger steps out into a function with no source code, it will continue to step out until a point is found that has source code.</p>	
<b>Show Execution Point</b>	<p>To return to the source file and line of code that the debugger is currently executing, click on the  (<b>Show Execution Point</b>) button.</p> <p>The appropriate line is highlighted, with a pink arrow in the left margin of the screen.</p>	
<b>Output</b>	<p>During a debug session, messages display in the Debug window detailing:</p> <ul style="list-style-type: none"> <li>Startup of session</li> <li>Termination of session</li> <li>Exceptions</li> <li>Errors</li> <li>Trace messages, such as those output using <i>Java System.out</i> or <i>.NET System.Diagnostics.Debug</i></li> </ul> <p>If you double-click on a debug message, either:</p> <ul style="list-style-type: none"> <li>A pop-up displays with more complete message text, or</li> <li>If there has been a memory leak, the file is displayed at the point at which the error occurred</li> </ul>	
<b>Save Output (and Clear Output)</b>	<p>You can save the entire contents of the Debug output to an external .txt file, or you can save selected lines from the output to the Enterprise Architect clipboard.</p> <p>To save all of the output to file, click on the  (<b>Save output to file</b>) button.</p> <p>To save selected lines to the clipboard, right-click on the selection and select the <b>Copy Selected to Clipboard</b> context menu option.</p> <p>When you have saved the output or otherwise do not want to display it any more, right-click on the current output and select the <b>Clear Results</b> context menu option.</p>	
<b>Switch to Profiler</b>	<p>If you are running a debug session on code, you can stop the debug session and immediately switch to a Profiling session.</p> <p>To switch from the debugger to the Profiler, click on the  <b>Switch to Profiler</b> option. The Profiler attaches to the currently-running process.</p> <p>This facility is available on Microsoft Native and Microsoft .NET</p>	<a href="#">Getting Started</a>  2558

Action	Detail	See also
	platforms.	

#### Learn more

- [Debug & Record Toolbar](#)<sup>[146]</sup>
- [View the Local Variables](#)<sup>[2234]</sup>
- [View the Call Stack](#)<sup>[2239]</sup>

#### 11.8.4.3.2 View the Local Variables

Whenever a thread encounters a breakpoint, the Locals window displays all the local variables for the thread at its current stack frame.

When a **Simulation** is active, the Local Variables displayed will be the current variables defined within the Simulation context using Javascript.

**Access** **Analyzer | Locals, or**

**Analyzer | Execution Analyzer:**  **| Locals**

#### Locals window facilities

Facility	Detail		See also
<b>Display Icons</b>	The value and type of any in-scope variables are displayed in a tree; each variable has a colored box icon that identifies the type of variable: <ul style="list-style-type: none"> <li>• Blue - Object with members</li> <li>• Green - Arrays</li> <li>• Pink - Elemental types</li> <li>• Yellow - Parameters</li> <li>• Red - Workbench instance</li> </ul>		
<b>Item Context Menu</b>	Option	Detail	
	<b>Break When Variable is Modified</b>	Set <b>data</b> breakpoints on the selected memory variable, to halt debugger execution at the line of code that has just caused the value of the variable to change.	<a href="#">Break When a Variable Changes Value</a> <sup>[2229]</sup>
	<b>View Memory at Address</b>	Display the raw values in memory at the selected address, in hex and ASCII.	<a href="#">Inspect Process Memory</a> <sup>[2242]</sup>
	<b>Show in String Viewer</b>	Display the variable string in the String Viewer dialog.	<a href="#">View Content Of Long Strings</a> <sup>[2235]</sup>

Facility	Detail		See also
	<b>Dump Variable Members to File</b>	Capture and store the selected variables to a separate location; a browser displays to select the appropriate .txt file name and file path.	
	<b>Save Snapshot of Variable</b>	Capture the value of a variable at a specific point in the life of that variable.	<a href="#">Variable Snapshots</a> <sup>[2236]</sup>
	<b>Compare Variable Snapshots</b>	Compare the values of a variable at different points in the life of that variable.	
	<b>Copy</b>	Copy the selected variable to the Enterprise Architect clipboard.	
	<b>Add Instance Run State to Diagram</b>	If you have opened a model diagram containing an Object of the Class for which the source code is being debugged, this option updates that Object with the Run State represented by the variable value.	
	<b>Set Conditional Breakpoint</b>	Add a breakpoint at the current execution position with a constraint for this variable matching its current value.	<a href="#">Breakpoint Properties</a> <sup>[2226]</sup>

**Learn more**

- [View Debug Variables in Code](#) <sup>[2236]</sup>
- [View Variables in Other Scopes](#) <sup>[2237]</sup>
- [Breakpoint and Marker Management](#) <sup>[2224]</sup>
- [View the Call Stack](#) <sup>[2239]</sup>

**Learning Center topics**

- (Alt+F1) | **Enterprise Architect | Build and Debug | Debug**

**11.8.4.3.2.1 View Content Of Long Strings**

For efficiency, the Locals window only shows partial strings. However, the size of any variable value displayed in the window can be up to 256 characters.

You can display the entire content of long strings on the String Viewer dialog.

**Access**    **Element | Show Source Viewer (Alt+7): right-click on variable | Show in String Viewer Analyzer | Locals: right-click on variable | Show in String Viewer, or**

**Analyzer | Execution Analyzer:**  | **Locals: right-click on variable | Show in String Viewer**

#### 11.8.4.3.2.2 View Debug Variables in Code Editors

During debugging, whenever a thread is suspended at a line of execution, you can inspect member variables in the Source Code Editor. To evaluate a member variable, use the mouse to move the cursor over the variable in the Editor, as shown in the following examples:

```
public void Print()
{
    int n = 0;
    while(names[n].Length > 0)
    {
        names = {[4] names[0]=book, names[0]=book, names[1]=novel, names[2]=film}, ...}
        Document d = new Document(names[n++]);
        d.Print();
    }
}
```

```
public void Print()
{
    int n = 0;
    while(32-bit signed integer n=0 0)
    {
        Document d = new Document(names[n++]);
        d.Print();
    }
}
```

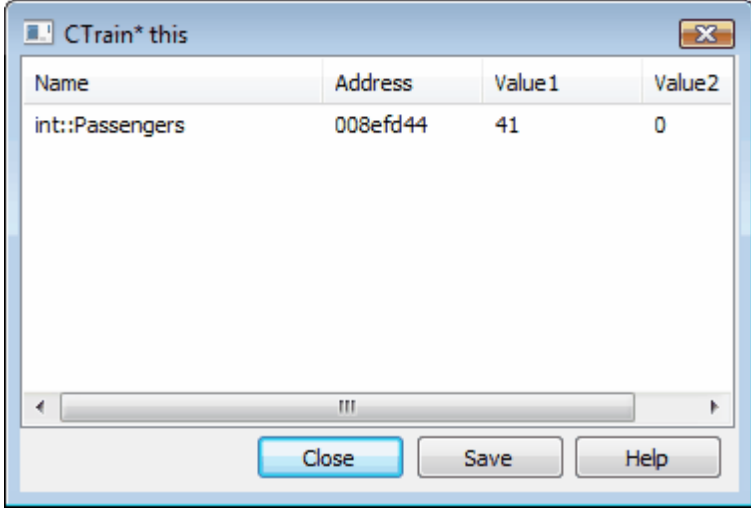
#### Learn more

- [View the Local Variables](#) <sup>2234</sup>
- [View Variables in Other Scopes](#) <sup>2237</sup>

#### 11.8.4.3.2.3 Variable Snapshots

It is possible to take a 'snapshot' of a variable when your program is at a breakpoint, and use this snapshot to see how the value of the variable changes at different points in its life. The debugger does not copy the value of the selected variable only; for complex variables it copies the values of the selected variable and of each of its hierarchy of members until it can find no debug information for a member or no more members can be found.

#### Capture variable snapshots

Step	Action	See also
1	In the Code Editor, set two breakpoints: <b>start</b> at the start of a function and <b>end</b> at the end of the function.	<a href="#">Setting Code Breakpoints</a> <small>[2226]</small>
2	At the <b>start</b> breakpoint, right-click on a variable in the Locals window and select the <b>Save Variable Snapshot</b> context menu option.	<a href="#">View the Local Variables</a> <small>[2234]</small>
3	Run the application.	
4	<p>When the <b>end</b> breakpoint is reached, right-click on the variable in the Locals window and select the <b>Compare Variable Snapshots</b> context menu option.</p> <p>A dialog displays that shows the original value from the first snapshot and the current value from the second snapshot, as illustrated below (this example is taken from the <i>EA.Example</i> model).</p> 	

**Notes**

- This facility compares the values of a variable by address; be aware that if there are different threads with different instances of the same variable name, these are not comparable

**11.8.4.3.3 View Variables in Other Scopes****Access** **Analyzer | Watches****Analyzer | Execution Analyzer:**  **| Watches****Topics**

Topic	Detail	See also
<b>Watches</b>	<p>The Watches window is most useful for native code (C, C++, VB) where it can be used to evaluate data items that are not available as Local Variables - data items with module or file scope and static Class member items.</p> <p>You can also use the window to evaluate static Class member items in Java and .NET</p> <p>To add a watch, type the name of the variable to in the toolbar, and press ( <b>Enter</b> ).</p> <p>To examine a static Class member variable in C++, Java or Microsoft .NET, enter its fully qualified name</p> <pre>CMyCl ass : : MySt at i cVar</pre> <p>To examine a C++ data symbol with module or file scope, just enter its name.</p> <p>Variables are evaluated by looking at the current scope; that is, the module of the current stack frame (you can change the scope at a breakpoint by double-clicking the frame in the Call Stack).</p> <p>If the global variable exists in a different module, you can examine the variable by prefixing the module name to the variable</p> <pre>modul ename! var i abl e_name</pre> <p>The history of items entered is saved. Previously entered names or expressions can be selected again using the up and down arrow keys.</p>	<p><a href="#">View Elements of Array</a> <sup>[2238]</sup></p> <p><a href="#">View the Local Variables</a> <sup>[2234]</sup></p> <p><a href="#">View Debug Variables in Code Editors</a> <sup>[2236]</sup></p> <p><a href="#">Inspect Process Memory</a> <sup>[2242]</sup></p>

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Build and Debug | Debug | View Global Variable**

#### **11.8.4.3.3.1 View Elements of Array**

##### Access **Analyzer | Watches**

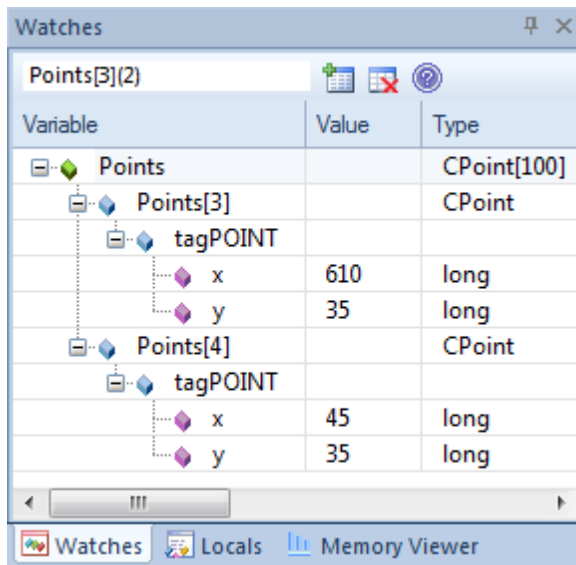
You can use the Watches window to inspect one or more specific elements of an array.

In the field to the left of the Watches window toolbar, type the name of the variable (the array) followed by the start element and, optionally, the number of elements to display. The start element is enclosed in square brackets and the count of elements is enclosed in parentheses; that is:

```
var i abl e[ st ar t _el ement ] ( co unt _of _el ement s )
```

For example:

Poi nt s[ 3] ( 2) displays the fourth and fifth elements of the Points array, as illustrated below.



If you entered `Points[3]` the Watches window would show the third array element only.

#### Learn more

- [View Variables in Other Scopes](#) <sup>[2237]</sup>
- [View the Local Variables](#) <sup>[2234]</sup>
- [View Debug Variables in Code Editors](#) <sup>[2236]</sup>

#### 11.8.4.3.4 View the Call Stack

The Call Stack window is used to display all currently running threads in a process. It can be used to identify which thread is operational immediately before program failure occurs.

When a **Simulation** is active, the Call Stack will show the current execution context for the running simulation. This will include a separate context stack for each concurrent simulation "thread".




A stack trace is displayed whenever a thread is suspended, through one of the step actions or through encountering a breakpoint. The Call Stack window can record a history of stack changes, and enables you to generate Sequence diagrams from this history.

#### Access **Analyzer | Call Stack**


#### Use to

- View stack history to understand the execution of a process
- View threads
- Save a call stack for later use
- Record call stack changes for Sequence diagram generation
- Generate a Sequence diagram from the call stack
- View the related code line in the Source Code Editor

#### Topics

Topic	Detail	See also
<b>Indicators</b>	<ul style="list-style-type: none"> <li>A pink arrow highlights the current stack frame</li> <li>A blue arrow indicates a thread that is running</li> <li>A red arrow indicates a thread for which a stack trace history is being recorded</li> </ul>	
<b>Save a Call Stack to a .TXT File</b>	<i>Not currently available.</i>	
<b>Record a Thread in a Debug Session</b>	<p>To record the execution of a <i>thread</i> and direct the recording to the Record &amp; Analyze window, right-click on the thread in the call stack and select the appropriate context menu option:</p> <ul style="list-style-type: none"> <li><b>Record</b> - to manually record the current thread during the debug session</li> </ul> <p>Used in conjunction with the 'step' buttons of the debugger; each function that is called due to a step command is logged to the Record &amp; Analyze window.</p> <ul style="list-style-type: none"> <li><b>Auto-Record</b> - to perform auto-recording during a debug session</li> </ul> <p>When you select this icon, the Analyzer begins recording and does not stop until either the program ends, you stop the debugger or you click on the <b>Stop</b> icon.</p>	<a href="#">The Recording History</a> <sup>[2533]</sup>
<b>Stop Recording</b>	<p>If you have started a manual or automatic recording of a thread you can stop it before completion; select the thread (indicated by a red arrow) and either:</p> <ul style="list-style-type: none"> <li>Click on the  (<b>Stop Recording</b>) button in the toolbar or</li> <li>Right-click and select the <b>Stop</b> context menu option</li> </ul>	
<b>Generate a Sequence Diagram from the Call Stack</b>	<p>To generate Sequence diagram from the call stack trace, either:</p> <ul style="list-style-type: none"> <li>Click on the  (<b>Generate Sequence Diagram of Stack</b>) button, or</li> <li>Right-click and select the <b>Generate Sequence Diagram</b> context menu option</li> </ul>	<a href="#">Create Sequence Diagram of Call Stack</a> <sup>[2241]</sup>
<b>Copy Stack to Recording History</b>	<p>To add the stack details immediately to the Record &amp; Analyze window (for later generation of Sequence diagrams) either:</p> <ul style="list-style-type: none"> <li>Click on the  (<b>Copy Stack to Recording History</b>) button, or</li> <li>Right-click and select the <b>Copy Stack to Record History</b> context menu option</li> </ul>	<a href="#">The Recording History</a> <sup>[2533]</sup>



Topic	Detail	See also
<b>Toggle Stack Depth</b>	To toggle between showing the full stack and showing only frames with source, click on the  ( <b>Toggle Stack Depth</b> ) button.	
<b>Display Related Code in Source Code Editor</b>	Double-click on a thread/frame to display the related line of code in the Source Code Editor; local variables are also refreshed for the selected frame.	<a href="#">Editing Source Code</a> <sup>[2146]</sup>

Learn more

- [Breakpoint and Marker Management](#) <sup>[2224]</sup>

**11.8.4.3.4.1 Create Sequence Diagram of Call Stack**


The Call Stack window records a history of stack changes from which you can generate Sequence diagrams.

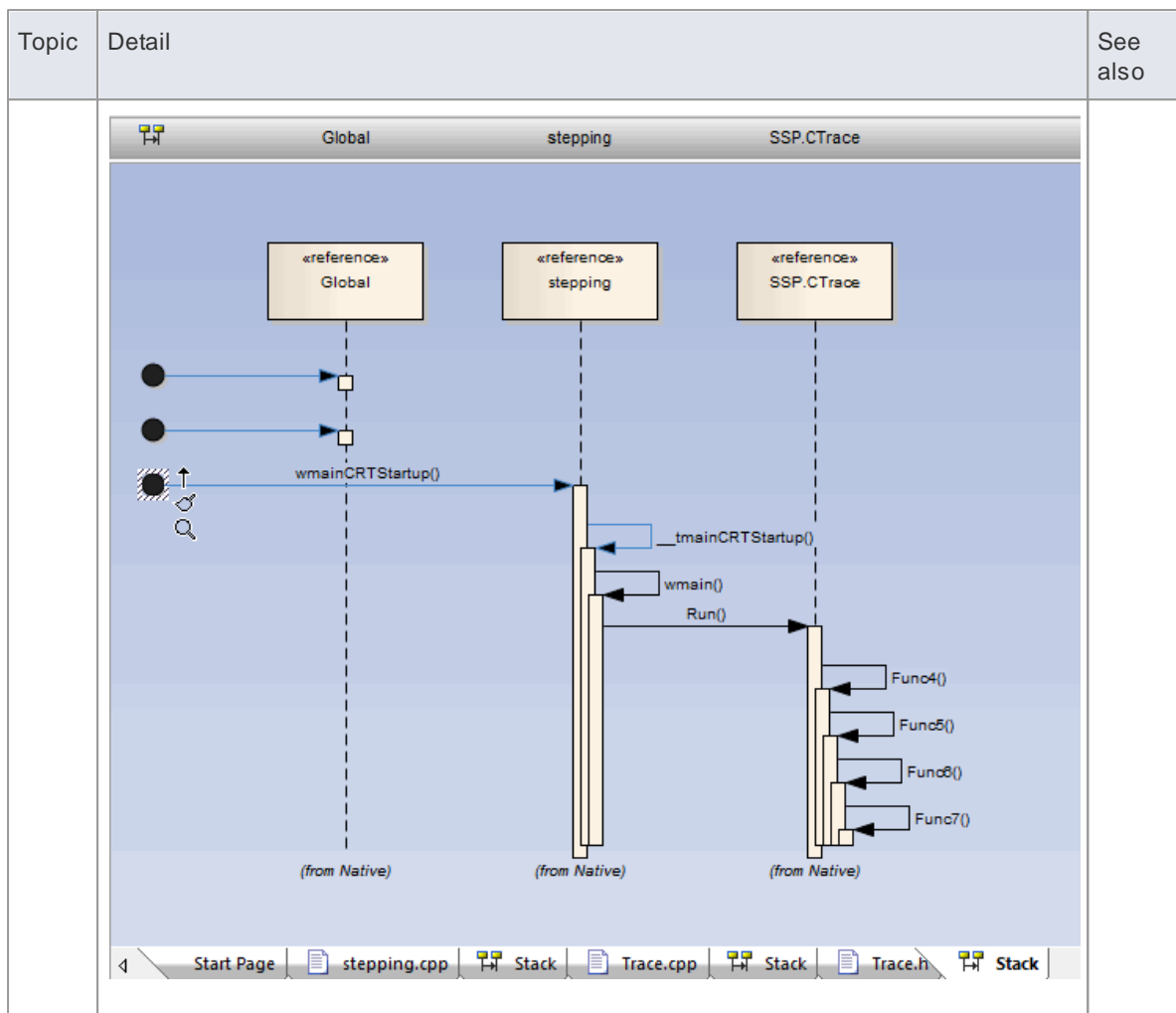
**Access**    **Analyzer | Call Stack**

Use to

- Record Call Stack changes for Sequence Diagram generation
- Generate a Sequence Diagram from the Call Stack

Topics

Topic	Detail	See also
<b>Usage</b>	To generate a Sequence diagram from the current Stack, click on the  ( <b>Generate Sequence Diagram of Stack</b> ) button on the Call Stack window toolbar.  This immediately generates a Sequence diagram in the Diagram View.	



### Learn more

- [View the Call Stack](#) <sup>2239</sup>

#### 11.8.4.3.5 Inspect Process Memory

Using the Memory Viewer, you can display the raw values of memory in hex and ASCII. You can manually define the memory address in the **Address** field (top right), or right-click on a variable in the Locals or Watches window and select the **View Memory at Address** context menu option.

**Access** **Analyzer | Memory Viewer**  
**Right-click on the item | View Memory at Address**

### Notes

- The Memory Viewer is available for debugging Microsoft Native Code Applications (C, C++, VB) running on Windows or within WINE on Linux

### Learn more

- [View the Local Variables](#) <sup>2234</sup>
- [View Variables in Other Scopes](#) <sup>2237</sup>

Learning Center topics

- (Alt+F1) | **Enterprise Architect** | **Build and Debug** | **Debug** | **View Memory**

**11.8.4.3.6 Show Loaded Modules**

For .NET and native Windows applications, you can list the **DLL's** loaded by the debugged process, using the **Modules window**. This list can also include associated **symbolic files** (**PDB** files) used by the debugger.

Access **Analyzer** | **Modules**

Modules Window display

Column	Description
<b>Path</b>	Shows the file path of the loaded module.
<b>Load Address</b>	Shows the base memory address of the loaded module.
<b>Modified Date</b>	Shows the local file date and the time the module was modified.
<b>Debug Symbols</b>	Shows: <ul style="list-style-type: none"><li>• The debug symbols type</li><li>• Whether debug information is present in the module, and</li><li>• Whether line information is present for the module (required for debugging)</li></ul>
<b>Symbol File Match</b>	Indicates the validity of the symbol file; if the value is false, the symbol file is out of date.
<b>Symbol Path</b>	Shows the file path of the symbol file, which must be present for debugging to work.
<b>Modified Date</b>	Shows the local file date and time the symbol file was created.

**11.8.4.3.7 Debug Another Process**



Rather than starting a process explicitly from within Enterprise Architect, you might want to debug an application (process) that is already running on your system.

In this case you can use the debugging capability to 'attach' to the process that is already running. Provided you have the appropriate debug information written into the running process, and/or associated debug files (such as .PDB files), the debugger binds to that process and initiates a debug session.

You can also 'detach' from the process after you have completed your inspection and leave the process to run as normal.

**Access** **Analyzer | Debugger > Debug window toolbar:** 

### Topics

Topic	Detail	See also
<b>Show Processes</b>	When you select to debug another process, the Attach To Process dialog displays.  You can limit the processes displayed using the radio buttons at the top of the dialog; to find a service such as Apache Tomcat or ASP.NET, select the <b>System</b> radio button.	
<b>Select Debugger</b>	When you select a process, you may need to choose the debugger from the <b>Debugger</b> dropdown list; however, if the selected package has already been configured in an Analyzer Script, then the debugger listed in the script is preset on the dialog.	
<b>Process Selection</b>	Once you double-click on a process containing debug information, and Enterprise Architect is attached to the process: <ul style="list-style-type: none"> <li>Any breakpoints encountered are detected by the debugger</li> <li>The process is halted when a breakpoint is encountered, and</li> <li>The information is available in the Debug window</li> </ul>	
<b>Detach From Process</b>	To detach from a process, click on the  ( <b>Debug Stop</b> ) button.	<a href="#">Debugger Facilities</a>  2231

#### 11.8.4.3.8 Process First Chance Exceptions

**Access** **Debug & Record Toolbar:**  | **Process First Chance Exceptions**

### Topics

Topic	Detail	See also
<b>Debug Process</b>	When an application is being debugged and the debugger is notified of an exception, the application is paused and the debugger responds in the way it is configured to do; it either: <ul style="list-style-type: none"> <li>Resumes the application and leaves the exception to the application to</li> </ul>	

Topic	Detail	See also
	<p>manage, or</p> <ul style="list-style-type: none"><li>• Keeps the application suspended and passes the exception to the appropriate routines for automatic resolution or manual intervention</li></ul>	
<b>Second Chance Exceptions</b>	<p>The Enterprise Architect debugger defaults to the first behavior, above.</p> <p>If the application can handle the exception, it continues to process; if it cannot handle the exception, the debugger is notified again and this time it must suspend the application and resolve the exception condition.</p> <p>In this behavior, because the debugger has encountered the exception twice, it is known as a <i>second-chance exception</i>; in this case, if the exception does not halt execution, it is ignored and you avoid spending time on conditions that do not impact the overall outcome of processing.</p> <p>You might work this way on large or complex systems that invariably involve exception conditions somewhere in the processing paths.</p>	
<b>First Chance Exceptions</b>	<p>However, if you want to examine every exception that occurs as soon as it occurs, you can set the debugger to adopt the second behavior.</p> <p>Because the debugger responds to the exception on first contact, it is known as a <i>first-chance exception</i>.</p> <p>You might work this way with individual functions or routines that must work cleanly or not at all.</p>	
<b>Selection</b>	<p>Select the <b>Process First Chance Exceptions</b> option to debug exceptions on first contact.</p> <p>Deselect the option to process exceptions only if the application fails when they occur.</p>	

## 11.9 Other Settings



You can set the default code options such as the editors for each of the programming languages available for Enterprise Architect and special options for how source code is generated or reverse engineered.

### Learn more

Topic	Details	Link
<b>General Options</b>	Describes the general options that apply to all languages when generating code from Enterprise Architect.	<a href="#">Source Code Engineering Options</a> <small>[2246]</small>
<b>Local Paths</b>	Describes how to define local paths for each Enterprise Architect user, using the Local Paths settings.	<a href="#">Local Paths</a> <small>[2255]</small>
<b>Local Path Dialog</b>	Enables you to set up local paths for a single user on a particular machine.	<a href="#">Local Path Dialog</a> <small>[2256]</small>
<b>Language Macros</b>	Describes how to define macros, which is helpful for reverse engineering.	<a href="#">Language Macros</a> <small>[2257]</small>
<b>Language Options</b>	Describes the options for generating and reverse-engineering code in each of the supported languages.	<a href="#">Language Options</a> <small>[2262]</small>
<b>Set Collection Classes</b>	Describes how to define Collection Classes.	<a href="#">Set Collection Classes</a> <small>[2259]</small>

### 11.9.1 Source Code Engineering Options

The following topics describe general options that apply to all languages when generating code from Enterprise Architect. These options are all available under the Source Code Engineering section of the Options dialog.

**Access** [Tools | Options | Source Code Engineering](#)

Topic	Detail	Link
<b>Source Code Options</b>	You can set certain defined options for a particular language while generating the code.	<a href="#">Source Code Options</a> <small>[2247]</small>

Topic	Detail	Link
<b>Options - Code Editors</b>	Describes how to access and configure the source code editor.	<a href="#">Options - Code Editors</a> [2250]
<b>Options - Object Lifetimes</b>	Describes configuring various options concerning Object Lifetimes.	<a href="#">Options - Object Lifetimes</a> [2253]
<b>Options - Attributes/Operations</b>	Describes configuring Attributes/Operations.	<a href="#">Options - Attribute/Operations</a> [2253]
<b>Code Page for Source Editing</b>	Provides an option to define the Unicode character set for code generation.	<a href="#">Code Page for Source Editing</a> [2255]

### 11.9.1.1 Source Code Options

When you generate code for a particular language, you can set certain options. These include:

- Create a default constructor
- Create a destructor
- Generate copy constructor
- Select default language
- Generate methods for implemented interfaces
- Set the Unicode options for code generation

**Access**   **Tools | Options | Source Code Engineering**

#### Configure code generation options

Field	Action	See also
<b>Always synchronize with existing file (recommended)</b>	Select the radio button to synchronize imported code with an existing file.	
<b>Replace (overwrite) existing source file</b>	Select the radio button to overwrite the existing source file with imported code .	
<b>Component Types</b>	Click on this button to open the Import component types dialog, to set up the importation of component types.	<a href="#">Import Component Types</a> [2249]
<b>Default Language for Code Generation</b>	Click on the drop-down arrow and select the default coding language for code generation.	

<b>Wrap long comment lines at</b>	Type in the number of characters to allow in a comment line before wrapping the text to the next line.	
<b>Auto Layout Diagram on Import</b>	Click on the drop-down arrow and select if and when a diagram is automatically generated on code import.	
<b>Output file use both CR &amp; LF</b>	Select the checkbox to include carriage returns and line feeds; set this option according to what operating system is currently in use, as code might not render correctly.	
<b>Prompt when synchronizing (reversing)</b>	Select the checkbox to display a prompt when synchronization occurs.	
<b>Remove hard breaks from comments on import</b>	Select the checkbox to remove hard breaks from commented sections on importation.	
<b>Auto generate role names when creating code</b>	Select the checkbox to generate role names when creating code.	
<b>Do not generate members where association direction is "Unspecified"</b>	Select the checkbox to prevent generation of members if the Association direction is <b>unspecified</b> .	
<b>Create dependencies for operation returns and parameter types</b>	Select the checkbox to generate dependencies for operation returns and parameter types.	
<b>Comments: Generate</b>	Select the checkbox to generate comments.	
<b>Comments: Reverse</b>	Select the checkbox to generate reverse comments.	
<b>Remove prefixes when generating Get/Set properties</b>	Type in the prefixes, separated by semi-colons, used in your variable naming conventions, to be removed in the variables' corresponding get/set functions.	
<b>Treat as suffixes</b>	Select the checkbox to use the prefixes defined in the <b>Remove prefixes when generating Get/Set properties</b> field as suffixes.	



<b>Capitalized Attribute Name for properties</b>	Select the checkbox to capitalize Attribute names for properties.	
<b>Use "Is" for Boolean property Get()</b>	Select the checkbox to use the /s keyword for the Boolean property <b>Get()</b> .	
<b>Code page for source editing</b>	Click on the drop-down arrow and select the character embedding format to apply.	

**Notes**

- It is worthwhile to configure these settings, as they serve as the defaults for all Classes in the model; you can override most of these on a per-Class basis using the custom settings (from the Code Generation dialog)

**Learn more**

- [Generate a Single Class](#) 

**11.9.1.1.1 Import Component Types**

The Import Component Types dialog enables you to configure what elements you would like to be created for files of any extension found while importing a source code directory.

**Access**   **Tools | Options | Source Code Engineering: Component Types**

**Use to**

- Configure what elements are to be created for files of any extension found while importing a source code directory

**Reference**

Field	Usage	See also
<b>Extension</b>	Indicates the extension name for a component type.	
<b>Type</b>	Indicates the type for a component.	
<b>Stereotype</b>	Indicates the stereotype name given to a component.	
<b>Component List</b>	The defined list of component types.	
<b>Save</b>	Saves a component definition and adds it to the component list.	

<b>New</b>	Allows a new component type to be defined.	
<b>Delete</b>	Deletes a highlighted component from the component list.	

#### Notes

- You can transport these import component types between models, using the Export Reference Data and Import Reference Data options

#### Learn more

- [Export Reference Data](#)<sup>[376]</sup>
- [Import Reference Data](#)<sup>[380]</sup>

### 11.9.1.2 Options - Code Editors

You access the source code editor options via the Code Editors page of the Options dialog. They enable you to configure options for Enterprise Architect's internal editor, as well as the default editor for DDL scripts. You can configure external editors for code languages on each language options page.

**Access**    **Tools | Options | Source Code Engineering | Code Editors**

#### Use to

- Configure options for Enterprise Architect's internal editor
- Configure the default editor for DDL Scripts
- Configure DDL Name Templates


#### Reference

Field	Usage	See also
<b>DDL Editor</b>	Defaults to blank, indicating that the Enterprise Architect code editor is the DDL editor in use.  You can select a different default editor if necessary: click on the (...) button to browse for and select the the required DDL editor. The editor name then displays in the <b>DDL Editor</b> field.	
<b>Default Database</b>	Indicates the default database used.	
<b>MySQL Storage Engine</b>	Indicates the storage engine used.	

Field	Usage	See also
<b>DDL Name Templates</b>	Specifies the Foreign Key, Primary Key and Unique Constraint Name templates; the (...) button enables you to specify template names.	<a href="#">Define Primary Key Name Template</a> <sup>[2357]</sup> <a href="#">Define Foreign Key Name Template</a> <sup>[2362]</sup>
<b>Use inbuilt editor if no external editor set</b>	Specifies the editor for code in a language if no external editor is defined for that language.	
<b>Show Line Numbers</b>	Shows line numbers in the editor.	
<b>Show Structure Tree</b>	Shows a tree with the results of parsing the open file (requires that the file is parsed successfully).	
<b>Don't parse files larger than</b>	Specifies an upper limit on file size for parsing. Use this to prevent performance decrease due to parsing very large files.	
<b>Syntax Highlighting Options</b>	Specifies both global and language-specific editor language properties.	<a href="#">Editor language properties</a> <sup>[2257]</sup>
<b>Automatically Reverse Engineer on File Save</b>	When this is checked, pressing <b>(Ctrl) + (S)</b> to save in the source code editor automatically reverse engineers the code in the same way as the <b>Save Source and Re-Synchronize Class</b> button does.	

#### 11.9.1.2.1 Editor Language Properties

Enterprise Architect enables you to specify syntax highlighting properties for all the programming languages that Enterprise Architect supports at installation.

**Access** **Tools | Options | Source Code Engineering | Code Editors: Syntax Highlighting Options (...)**  
**Project Browser element context menu | View Source Code:**  **| Syntax Highlighting Options**

#### Use to

- Configure Syntax highlighting for various coding languages

#### Reference

On the Editor Language properties dialog:

Topic	Detail	See also
<b>General Features</b>	<p>The panel on the left of the dialog lists the languages for which you can set properties.</p> <p>The <i>(Global)</i> item at the top of the list enables you to set properties that apply to all programming languages; however, you can reset a global property to a different value for a specific language, on the page for that language.</p> <p>Resetting a global property for one language does not affect that property's value for the other languages.</p> <p>Click on the required language in the list, to display the properties for that language:</p> <ul style="list-style-type: none"> <li>Properties shown in bold indicate that this is the highest level at which this property can be defined (for most language options other than <i>Global</i>, this is effectively the only point at which the property is defined)</li> <li>Properties shown in normal font are generally the global properties that you can reset just for the current language</li> </ul> <p>Scroll through the property categories and individual properties for the language; you can collapse and expand categories as necessary, using the expansion box next to the category name (☐).</p> <p>When you click on a property name, an explanation of that property displays in the panel at the bottom right of the dialog.</p>	
<b>Define Properties</b>	<p>To define a property, click on the value field following the property name; depending on the type of property, either the field is enabled for direct editing or a drop-down arrow or <b>Browse</b> button displays (as described for the Tagged Values window) to enable you to select the values to define the property.</p> <p>Select or type in the required values.</p> <p>The Toolbar icons enable you to:</p> <ul style="list-style-type: none"> <li>Save your changes to the properties</li> <li>Reset ALL properties fields to the default settings shipped with Enterprise Architect</li> <li>Reset the current style field to the default setting (not enabled for non-style fields)</li> </ul>	<a href="#">Tagged Values</a> <small>[1138]</small>
<b>Assign Keys to Macros</b>	<p>The Macros category enables you to assign <b>(Ctrl+Alt+&lt;n&gt;)</b> keystroke combinations to coding macros that you have created yourself in the Source Code Viewer.</p> <p>When you click on the <b>Browse</b> button in a selected <b>Macro</b> field, the Open Macro dialog displays; this dialog lists the existing macros and, if a key combination has been assigned to a macro, what that key combination is.</p> <p>Click on the name of the macro and on the <b>Open</b> button to assign the selected keys to the macro.</p>	<a href="#">Source Code Viewer</a> <small>[2149]</small>

### Notes

- You cannot currently set properties for any additional languages you include through an MDG Technology
- You can resize this dialog, if required

### Learn more

- [Function Details](#) 2157 (Syntax Highlighting)
- [MDG Technology](#) 1552

## 11.9.1.3 Options - Object Lifetimes

This set of options enables you to configure various options concerning Object Lifetimes.

**Access**   **Tools | Options | Source Code Engineering | Object Lifetimes**

### Use to

- Define constructor details when generating code
- Specify whether to create a copy constructor
- Destructor details

### Reference

Field	Usage	See also
<b>Constructor</b>	This set of options specify if a constructor is generated, if a constructor is in-line and the visibility of the default constructor.	
<b>Copy Constructor</b>	This set of options specify if a copy constructor is generated, if a copy constructor is in-line and the visibility of the default copy constructor.	
<b>Destructor</b>	This set of options specify if a destructor is generated, if a destructor is in-line, if a destructor is defined as <i>virtual</i> and the visibility of the default constructor.	

### Notes

- Some options are related to C++ only

## 11.9.1.4 Options - Attribute/Operations

Your use of attributes and operations can be configured in a number of ways. You can set options to:

- Set the default name generated from imported attributes
- Generate methods for implemented interfaces

- Delete model attributes not included in the code during reverse synchronization
- Delete model methods not included in the code during reverse synchronization
- Delete code from features contained in the model during forward synchronization
- Delete model associations and aggregations that correspond to attributes not included in the code during reverse synchronization
- Define whether or not the bodies of methods are included and saved in the model when reverse engineering
- Create attributes in quick succession, clearing the Attributes dialog when you click on **Save** so that you can enter another attribute name

You configure these options on the Attribute/Operations page of the Options dialog.

Access   **Tools | Options | Source Code Engineering | Attribute/Operations**

### Options

Field	Usage	See also
<b>Attribute Specifications: Default name for associated attrib</b>	Specify the default name generated from imported attributes.	
<b>Attribute Specifications: On reverse synch, delete model attributes not in code</b>	Set to indicate that on reverse synchronization, attributes in the model that are not included within code are automatically removed from the model.	<a href="#">Update Package Contents</a> <sup>[211]</sup>
<b>Attribute Specifications: On reverse synch, delete model associations not in code</b>	Set to indicate that on reverse synchronization, associations in the model that are not included within code are automatically removed from the model.	
<b>Operation Specifications: Generate methods for implemented interface</b>	Set to indicate that methods are generated for an implemented interface.	
<b>Operation Specifications: On reverse synch, delete model methods not in code</b>	Set to indicate that on reverse synchronization, methods in the model that are not included within code are automatically removed from the model.	<a href="#">Update Package Contents</a> <sup>[211]</sup>
<b>Operation Specifications: Include method bodies in model when reverse engineering</b>	Set to indicate that on reverse engineering code, method bodies in the code are included within your model.	
<b>Options: After Save, re-select edited item</b>	Set to indicate that after saving an attribute or operation, the Properties dialog continues to display the details of the selected feature.	<a href="#">General Properties of Attributes</a> <sup>[100]</sup>

Field	Usage	See also
	If deselected, indicates that the fields of the Properties dialog will clear so that you can enter another attribute or operation name and details immediately.	<a href="#">General Properties of Operations</a> <sup>[1015]</sup>
<b>On forward synch, prompt to delete code features not in model</b>	Set to indicate that, during forward synchronization, the Synchronize Element <package name>.<element name> dialog displays, so that you can either ignore, reassign or delete features in the code that are not in the model.	<a href="#">Synchronize Model and Code</a> <sup>[2118]</sup>

### 11.9.1.5 Code Page for Source Editing

Enterprise Architect enables you to define the Unicode character set for code generation.

**Access** [Tools | Options | Source Code Engineering](#)

#### Set the Unicode character set

Step	Action	See also
<b>1</b>	Select the <b>Source Code Engineering</b> option. The Source Code Engineering page of the Options dialog displays.	
<b>2</b>	In the <b>Code page for source editing</b> field, click on the drop-down arrow and select the appropriate Unicode character set.	
<b>3</b>	Click on the <b>Close</b> button.	

## 11.9.2 Local Paths

When a team of developers are working on the same Enterprise Architect model, each developer might store their version of the source code in their local file system, but not always at the same location as their fellow developers. To manage this scenario in Enterprise Architect, you can define local paths for each user, on the Local Paths dialog.

You can use local paths in generating code and reverse engineering, and in version control, developing XML schemas and generating document and web reports.

Local paths might take a little time to set up, but if you want to work collaboratively on source and model concurrently, the effort is well worth while.

For example, if:

- Developer A stores her .java files in a C:\Java\Source directory, while developer B stores his in D:

\Source, and

- Both developers want to generate and reverse engineer into the same Enterprise Architect model located on a shared (or replicated) network drive

Developer A might define a local path of:

```
JAVA_SOURCE = " C: \ Java \ Source "
```

All Classes generated and stored in the Enterprise Architect project are stored as:

```
%JAVA_SOURCE% <xxx. java>
```

Developer B defines a local path as:

```
JAVA_SOURCE = " D: \ Source "
```

Now, Enterprise Architect stores all java files in these directories as:

```
%JAVA_SOURCE% <filename>
```

On each developer's machine, the filename is expanded to the correct local version.

Access   **Settings | Local Directories and Paths**

Learn more

- [Local Paths Dialog](#) 

### 11.9.3 Local Paths Dialog

Using the Local Paths dialog, you can set up local paths for a single user on a particular machine. For a description of the use of Local Paths, see the *Local Paths* topic.

Access   **Settings | Local Directories and Paths**

Options

Field/Option/ Button	Action	See also
<b>Path</b>	Enter the path of the local directory in the file system (for example, <b>d:\java\source</b> ).	
<b>ID</b>	Enter the shared ID that is substituted for the Local Path (for example, <b>JAVA_SRC</b> ).	
<b>Type</b>	Enter the type of path to apply to (for example, <b>Java</b> ).	
<b>Apply Path</b>	Select a path and click on this button to update any existing full path names in the model to the shared relative path name. For example:	



Field/Option/ Button	Action	See also
	<code>d:\java\source\main.java</code> might become <code>%JAVA_SRC%\main.java</code>	
<b>Expand Path</b>	Click on this button to remove a relative path and substitute the full path name (the opposite effect of the <b>Apply Path</b> button).	

**Notes**

- You can also set up a hyperlink on a diagram to access the Local Paths dialog, to switch, update or expand your current Local Path
- If the act of Expanding or Applying a path for a linked file will create a duplicate record, the process will skip that record and display a message at the end of the process

**Learn more**

- [Local Paths](#)<sup>[2255]</sup>
- [Hyperlink](#)<sup>[2002]</sup>

**11.9.4 Language Macros**

When reverse engineering a language such as C++, you might find preprocessor directives scattered throughout the code. This can make code management easier, but can hamper parsing of the underlying C++ language.

To help remedy this, you can include any number of *macro* definitions, which are ignored during the parsing phase of the reverse engineering. It is still preferable, if you have the facility, to preprocess the code using the appropriate compiler first; this way, complex macro definitions and defines are expanded out and can be readily parsed. If you don't have this facility, then this option provides a convenient substitute.

**Access**   **Settings | Preprocessor Macros****Define a macro**

Step	Action	See also
<b>1</b>	Select the <b>Preprocessor Macros</b> menu option. The Language Macros dialog displays.	
<b>2</b>	Click on the <b>Add New</b> button.	
<b>3</b>	Enter details for your macro.	

Step	Action	See also
4	Click on the <b>OK</b> button.	

### Topics

Topic	Detail	See also
<b>Macros Embedded Within Declarations</b>	<p>Macros are sometimes used within the declaration of Classes and operations, as in the following examples:</p> <pre>class __declspec Foo {     int __declspec Bar (int p); };</pre> <p>If <i>declspec</i> is defined as a C++ macro, as outlined above, the imported Class and operation contain a Tagged Value called <i>DeclMacro1</i> with value <i>__declspec</i> (subsequent macros would be defined as <i>DeclMacro2</i>, <i>DeclMacro3</i> and so on).</p> <p>During forward engineering, these Tagged Values are used to regenerate the macros in code.</p>	
<b>Define Complex Macros</b>	<p>It is sometimes useful to define rules for complex macros that can span multiple lines; Enterprise Architect ignores the entire code section defined by the rule.</p> <p>Such macros can be defined in Enterprise Architect as in the following two examples; both types can be combined in one definition.</p> <p><b>Block Macros</b></p> <pre>BEGIN_INTERFACE_PART ^ END_INTERFACE_PART</pre> <p>The ^ symbol represents the body of the macro - this enables skipping from one macro to another; the spaces surrounding the ^ symbol are required.</p> <p><b>Function Macros</b></p> <pre>RTTI_EMULATION( )</pre> <p>Enterprise Architect skips over the token including everything inside the parentheses.</p>	

### Notes

- You can transport these language macro (or preprocessor macro) definitions between models, using the **Export Reference Data** and **Import Reference Data** options; the macros are exported as a Macro List

### Learn more







- [Export Reference Data](#) <sup>378</sup>

- [Import Reference Data](#) <sup>[380]</sup>




### 11.9.5 Set Collection Classes

Enterprise Architect enables you to define *Collection Classes* for generating code from Association connectors where the target role has a multiplicity setting greater than 1.

Topic	Detail	See also
<b>Defining Collection Classes</b>	<p>On the Source Code Engineering section of the Options dialog (select the <b>Tools   Options   Source Code Engineering</b> option), on each language page click on the <b>Collection Classes</b> button.</p> <p>The Collection Classes for Association Roles dialog displays.</p> <p>On this dialog, you can define:</p> <ul style="list-style-type: none"> <li>• The default Collection Class for 1..* roles</li> <li>• The ordered Collection Class to use for 1..* roles</li> <li>• The qualified Collection Class to use for 1..* roles</li> </ul>	<a href="#">Source Code Options</a> <sup>[224]</sup>
<b>Defining Collection Classes for a specific Class</b>	Class-specific Collection classes can be defined by clicking the <b>Collection Classes</b> button in the Class Properties dialog of the element.	<a href="#">Element Details</a> <sup>[959]</sup>
<b>Code Generation Precedence</b>	<p>When Enterprise Architect generates code for a connector that has a multiplicity role &gt;1:</p> <ol style="list-style-type: none"> <li>1. If the Qualifier is set use the qualified collection: <ul style="list-style-type: none"> <li>• for the Class if set</li> <li>• else use the code language qualified collection</li> </ul> </li> <li>2. If the Order option is set use the ordered collection: <ul style="list-style-type: none"> <li>• for the Class if set</li> <li>• else use the code language ordered collection</li> </ul> </li> <li>3. Else use the default collection: <ul style="list-style-type: none"> <li>• for the Class if set</li> <li>• else use the code language default collection</li> </ul> </li> </ol>	
<b>Using Markers</b>	You can include the marker <b>#TYPE#</b> in the collection name; Enterprise Architect replaces this with the name of the Class being collected at source generation time (for example, <i>Vector&lt;#TYPE#&gt;</i> would become <i>Vector&lt;foo&gt;</i> ).	

Topic	Detail	See also
	Conversely, when reverse engineering, an Association connector is also created if a matching entry (e.g. <i>foo</i> if <i>foo</i> is found in the model) is defined as a Collection Class.	
<b>Additional Collection Classes</b>	Additional Collection Classes can be defined in the language options page (C#, C++, Java).	<a href="#">Example Use of Collection Classes</a>  <a href="#">C# Options</a>  <a href="#">C++ Options</a>  <a href="#">Java Options</a> 
<b>Member Type</b>	<p>On both the Source Role and Target Role tabs of the Association Property dialog (accessible from the right-click context menu of any Association) there is a <b>Member Type</b> field.</p> <p>If you set this, the value you enter overrides all the above options.</p>	<a href="#">Source Role</a>  <a href="#">Target Role</a> 

#### Learn more

- [Generate Source Code](#) 
- [Importing Source Code](#) 
- [Source Code Options](#) 

### 11.9.5.1 Example Use of Collection Classes

Consider the following source code:

```

class Class1
{
public:
    Class1();
    virtual ~Class1();
    CMap<CString, LPCTSTR, Class3*, Class3*> att;
    Vector<Class2> *att1;
    TemplatedClass<class1, class2> *att2;
    CList<Class4> *att3;
};

class Class2
{
public:
    Class2();
    virtual ~Class2();
};

class Class3
{
public:

```

```

    Class3();
    virtual ~Class3();

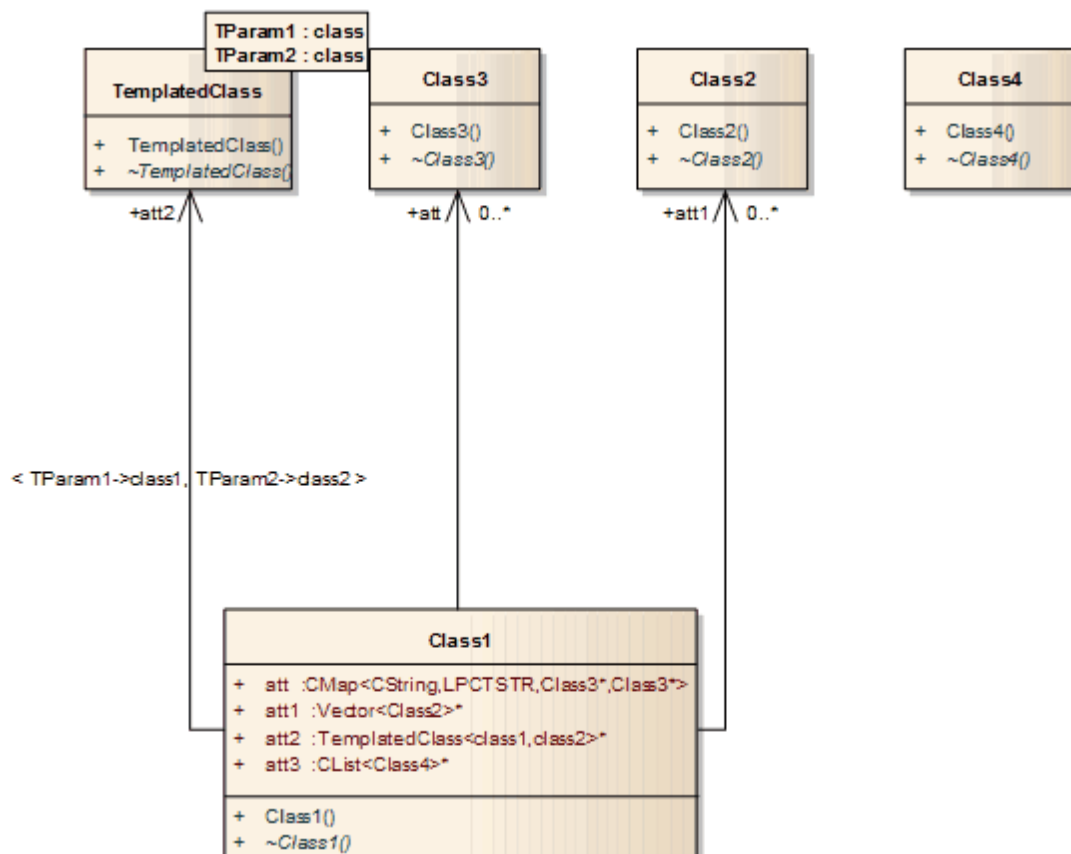
};

class Class4
{
public:
    Class4();
    virtual ~Class4();
};

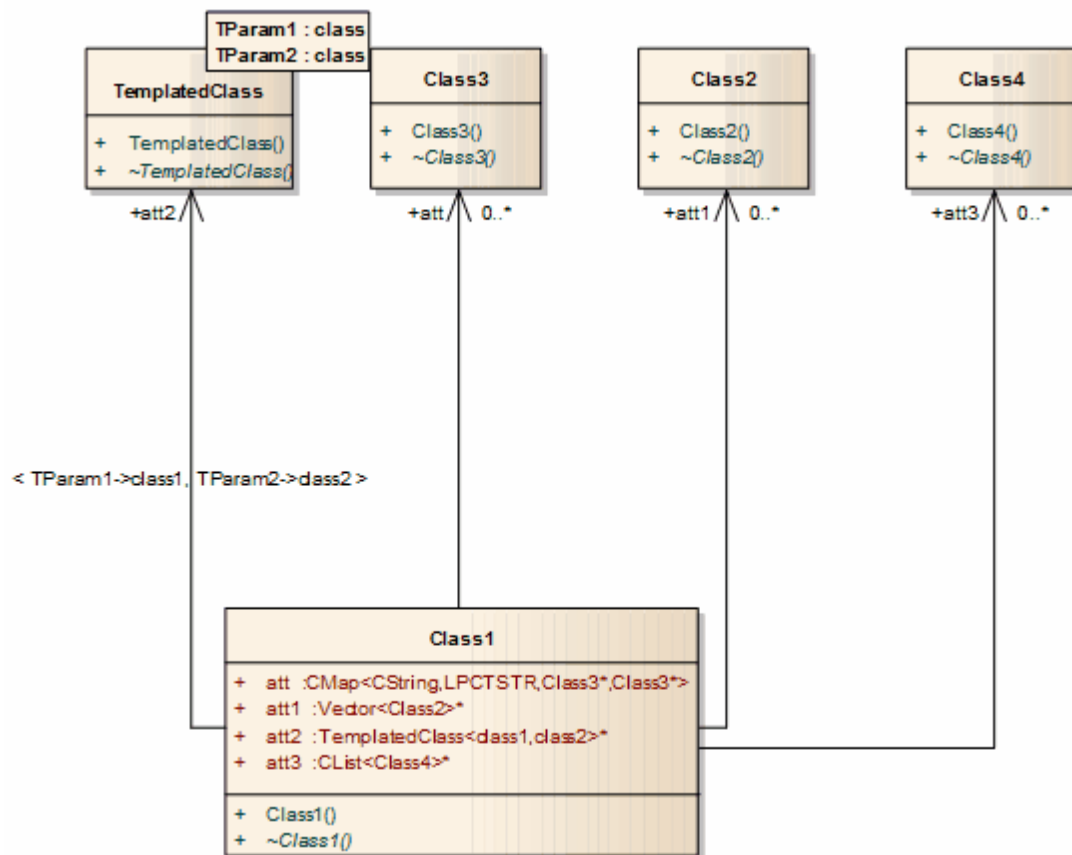
template<class TParam1, class TParam2>
class TemplatedClass
{
public:
    TemplatedClass() {
    }
    virtual ~TemplatedClass() {
    }
};

```

If this code is imported into the system with default import options, this diagram is generated:



If, however, you enter the value **CList<#Type#>** in the **Additional Collection Classes** field in the language options page (**C#**, **Java**, **C++**), an Association connector is also created to **Class 4**:



### Learn more

- [C# Options](#) <sup>2268</sup>
- [Java Options](#) <sup>2272</sup>
- [C++ Options](#) <sup>2269</sup>

## 11.9.6 Language Options

You can set up various options for how Enterprise Architect handles a particular language when generating and reverse-engineering code.

**Access** [Tools | Options | Source Code Engineering <language name>](#)

Languages Supported	See also
Action Script	<a href="#">ActionScript Options</a> <sup>2264</sup>

Languages Supported	See also
Ada 2005 (in the Systems Engineering and Ultimate editions of Enterprise Architect)	<a href="#">Ada 2005 Options</a>  2264
ArcGIS	<a href="#">ArcGIS Options</a>  2265
ANSI C	<a href="#">C Options</a>  2266
C#	<a href="#">C# Options</a>  2268
C++	<a href="#">C++ Options</a>  2269
Delphi	<a href="#">Delphi Options</a>  2271
Delphi Properties	<a href="#">Delphi Properties</a>  2272
Java	<a href="#">Java Options</a>  2272
PHP	<a href="#">PHP Options</a>  2274
Python	<a href="#">Python Options</a>  2273
SystemC	<a href="#">SystemC Options</a>  2275
Verilog (Systems Engineering and Ultimate editions)	<a href="#">Verilog Options</a>  2276
VHDL (Systems Engineering and Ultimate editions)	<a href="#">VHDL Options</a>  2277
Visual Basic	<a href="#">Visual Basic Options</a>  2277
Visual Basic .NET	<a href="#">VB.NET Options</a>  2275
MDG Technology Languages	<a href="#">MDG Technology Language Options</a>  2278
Reset Options	<a href="#">Reset Options</a>  2279

### 11.9.6.1 ActionScript Options

If you intend to generate **ActionScript** code from your model, you can configure the code generation options using the **ActionScript Specifications** page of the Options dialog to:

- Specify default ActionScript version to generate (AS2.0 or AS3.0)
- Specify default file extensions (header and source)
- Specify default source directory
- Specify the editor for ActionScript code

**Access**   **Tools | Options | Source Code Engineering | ActionScript**

#### Options

Field	Action	See also
<b>Disable Language</b>	Leave this checkbox unselected to support ActionScript code generation.  Select this check box to <b>disable</b> ActionScript code support.	
<b>Options for the current model</b>	Specifies the options used in the current model.  These options affect all users of the current model; however, they do not apply to other models.	
<b>Options for the current user</b>	Specifies the options used for the current user; these options apply to all models that are accessed by the user.	
<b>Collection Classes</b>	Click on this button to open the Collection Classes for Association Roles dialog, through which you specify the collection class definitions for Association connectors.	<a href="#">Collection Classes</a> [2259]

### 11.9.6.2 Ada 2005 Options


If you intend to generate **Ada 2005** code from your model, you can configure the code generation options using the **Ada** page of the Options dialog to:

- Inform the reverse engineering process whether the name of the Tagged Record is the same as the package name
- Advise the engine of the alternate Tagged Record name to locate
- Specify whether the engine should create a reference type for the Tagged Record (if one is not defined)
- Supply the name of the reference type to be created (default is Ref)
- To specify the reference parameter of a Reference / Access type
- Tell the engine to ignore the name of the reference parameter
- Indicate the name of the reference parameter to locate



**Access** [Tools](#) | [Options](#) | [Source Code Engineering](#) | [Ada](#)

### Options

Field	Action	See also
<b>Disable Language</b>	Leave this checkbox unselected to support Ada 2005 code generation. Select this check box to <b>disable</b> Ada 2005 code support.	
<b>Options for the current model</b>	Specifies the options used in the current model. These options affect all users of the current model; however, they do not apply to other models.	
<b>Options for the current user</b>	Specifies the options used for the current user; these options apply to all models that are accessed by the user.	
<b>Collection Classes</b>	Click on this button to open the Collection Classes for Association Roles dialog, through which you specify the collection class definitions for Association connectors.	<a href="#">Collection Classes</a> 

### Notes

- Ada 2005 support is available in the System Engineering and Ultimate editions of Enterprise Architect

### 11.9.6.3 *ArcGIS Options*


If you intend to generate **ArcGIS** code from your model, you can configure the code generation options using the **ArcGIS** page of the Options dialog to:

- Specify default file extensions
- Specify default source directory
- Specify the editor for ArcGIS code

**Access** [Tools](#) | [Options](#) | [Source Code Engineering](#) | [ArcGIS](#)

### Options

Field	Action	See also
<b>Disable Language</b>	Leave this checkbox unselected to support ArcGIS code generation. Select this check box to <b>disable</b> ArcGIS code support.	
<b>Options for the current model</b>	Specifies the options used in the current model.	

	These options affect all users of the current model; however, they do not apply to other models.	
<b>Options for the current user</b>	Specifies the options used for the current user; these options apply to all models that are accessed by the user.	
<b>Collection Classes</b>	Click on this button to open the Collection Classes for Association Roles dialog, through which you specify the collection class definitions for Association connectors.	<a href="#">Collection Classes</a> 

#### 11.9.6.4 C Options



If you intend to generate **C** code from your model, you can configure the code generation options using the **C Specifications** page of the Options dialog to:

- Support Object Oriented coding
- Indicate default file extensions (header and source)
- Indicate default source directory
- Indicate an editor for C code
- Specify a path that Enterprise Architect uses to search for the implementation file; the first path in the list is the default path when generating

Access   **Tools | Options | Source Code Engineering | C**

#### Options

Field	Action	See also
<b>Disable Language</b>	Leave this checkbox unselected to support C code generation. Select this option to <b>disable</b> C code support.	
<b>Options for the current model</b>	In the value fields, specify the options that affect all users of the <b>current</b> model: <ul style="list-style-type: none"> <li>• The default header and source file extensions for the code files</li> <li>• Support for Object Oriented programming; if this is <b>True</b>, then set: <ul style="list-style-type: none"> <li>• The Namespace delimiter character</li> <li>• Whether the first parameter of an operation is a Class reference</li> <li>• The parameter reference style in generated C code</li> <li>• The reference parameter name in generated code</li> <li>• The default Constructor name in generated code</li> <li>• The default Destructor name in generated code</li> </ul> </li> </ul>	

	These options do not apply to <b>other</b> models.	
<b>Options for the current user</b>	<p>In the value fields, specify the options that apply under your own user ID in <b>all</b> models that you access:</p> <ul style="list-style-type: none"> <li>• The default attribute type to create (fixed as <b>int</b>)</li> <li>• Whether a <b>#define</b> constant is imported as an attribute in imported C code (if Object Oriented programming is set to <b>True</b>)</li> <li>• Whether to generate comments for C methods to the declaration, and to reverse engineer comments from the declaration</li> <li>• Whether to generate comments for C methods to the implementation, and to reverse engineer comments from the implementation</li> <li>• Whether to update comments in regenerating code from the model</li> <li>• Whether to update the implementation file in re-generating code from the model</li> <li>• The default source code directory location (click on the <b>Browse</b> button - ( ... ))</li> <li>• The default file extensions to read when importing a directory of C code</li> <li>• The Code Editor to use (click on the <b>Browse</b> button - ( ... ))</li> <li>• The search path for the implementation file relative to the header file path</li> </ul>	
<b>Statemachine Engineering</b>	<p>In the value fields, use the drop-down arrows to set the options to <b>True</b> or <b>False</b>; these options apply to generating code from State Machine models in the <b>current</b> model only:</p> <ul style="list-style-type: none"> <li>• <b>Use the new Statemachine Template</b> - set to <b>True</b> to use the code generation templates from Enterprise Architect Release 11 and later, set to <b>False</b> to apply the EASL Legacy templates</li> <li>• <b>Generate Trace Code</b> - set to <b>True</b> to generate Trace code, <b>False</b> to omit it</li> </ul>	<a href="#">Legacy State Machine Templates</a>  <sup>[2126]</sup>
<b>Collection Classes</b>	Click on this button to open the Collection Classes for Association Roles dialog, through which you specify the collection class definitions for Association connectors.	<a href="#">Collection Classes</a>  <sup>[2259]</sup>

Learn more

- [C Conventions](#)  <sup>[2084]</sup>
- [Object Oriented Programming In C](#)  <sup>[2086]</sup>

### 11.9.6.5 C# Options

If you intend to generate **C#** code from your model, you can configure the code generation options using the **C# Specifications** page of the Options dialog to:

- Indicate the default File extension
- Specify a directory for opening and saving C# source code
- Indicate additional Collection Classes - to define custom collection classes, which can be simple substitutions (such as `CArray<#TYPE#>`) or a mix of other strings and substitutions (such as `Cmap<CString,LPCTSTR,#TYPE#*,#TYPE#*>`); the following collection classes are defined by default:
  - `List<#TYPE#>;Stack<#TYPE#>;Queue<#TYPE#>;`

**Access**   **Tools | Options | Source Code Engineering | C#**

#### Options

Field	Action	See also
<b>Disable Language</b>	Leave this checkbox unselected to support C# code generation. Select this check box to <b>disable</b> C# code support.	
<b>Options for the current model</b>	In the value fields, specify the options that affect all users of the <b>current</b> model: <ul style="list-style-type: none"> <li>• The default source file extensions for the code files</li> <li>• The additional Collection Classes</li> </ul> <p>These options do not apply to <b>other</b> models.</p>	
<b>Options for the current user</b>	In the value fields, specify the options that apply under your own user ID in <b>all</b> models that you access: <ul style="list-style-type: none"> <li>• The default attribute type to create</li> <li>• Whether Namespaces should be generated when generating C# Classes</li> <li>• Whether to remove new lines (hard carriage returns) from the summary tag when importing XML.NET style comments</li> <li>• Whether to generate a Finalizer method when generating code for a C# Class</li> <li>• Whether to generate a Dispose method when generating code for a C# Class</li> <li>• The default source code directory location (click on the <b>Browse</b> button - ( ... ))</li> <li>• The Code Editor to use (click on the <b>Browse</b> button - ( ... ))</li> </ul>	
<b>Statemachine Engineering</b>	In the value fields, use the drop-down arrows to set the options to <b>True</b> or <b>False</b> ; these options apply to generating code from State Machine models in the <b>current</b> model only.	

	<ul style="list-style-type: none"> <li>• <b>Use the new StateMachine Template</b> - set to <b>True</b> to use the code generation templates from Enterprise Architect Release 11 and later, set to <b>False</b> to apply the EASL Legacy templates</li> <li>• <b>Generate Trace Code</b> - set to <b>True</b> to generate Trace code, <b>False</b> to omit it</li> </ul>	<a href="#">Legacy State Machine Templates</a> <sup>[2126]</sup>
<b>Collection Classes</b>	Click on this button to open the Collection Classes for Association Roles dialog, through which you specify the collection class definitions for Association connectors.	<a href="#">Collection Classes</a> <sup>[2259]</sup>

Learn more

- [C# Conventions](#) <sup>[2088]</sup>

### 11.9.6.6 C++ Options

If you intend to generate **C++** code from your model, you can configure the code generation options using the **C++ Specifications** page of the Options dialog to:


- Indicate the version of C++ to generate; this controls the set of templates used and how properties are created
- Specify the default reference type used when a type is specified by reference
- Specify the default file extensions
- Specify default Get/Set prefixes
- Define additional Collection Classes - to define custom Collection Classes, which can be simple substitutions (such as `CArray<#TYPE#>`) or a mix of other strings and substitutions (such as `Cmap<CString,LPCTSTR,#TYPE#*,#TYPE#*>`); the following Collection Classes are defined by default:
  - `CArray<#TYPE#>;CMap<CString,LPCTSTR,#TYPE#*,#TYPE#*>;`
- Specify a default source directory
- Specify the path that the system uses to search for the implementation file; the first path in the list is the default path when generating new implementation files and parsing existing files - if you add further directories, the system also searches these when parsing existing files

Access   **Tools | Options | Source Code Engineering | C++**

Options

Field	Action	See also
<b>Disable Language</b>	Leave this checkbox unselected to support C++ code generation. Select this option to <b>disable</b> C++ code support.	
<b>Options for the current model</b>	In the value fields, specify the options that affect all users of the <b>current</b> model:	

	<ul style="list-style-type: none"> <li>• The version of C++ you are using (which determines which templates to use when generating code)</li> <li>• The default reference type to use when creating properties for C++ attributes by reference</li> <li>• The default header and source file extensions for the code files</li> <li>• The default 'Get' prefix</li> <li>• The default 'Set' prefix</li> <li>• The additional Collection Classes</li> </ul> <p>These options do not apply to <b>other</b> models.</p>	
<b>Options for the current user</b>	<p>In the value fields, specify the options that apply under your own user ID in <b>all</b> models that you access:</p> <ul style="list-style-type: none"> <li>• The default attribute type to create</li> <li>• Whether Namespaces should be generated when generating C++ Classes</li> <li>• What style to apply when generating and processing comments for C++</li> <li>• Whether to generate comments for C++ methods to the declaration, or reverse engineer comments from the declaration</li> <li>• Whether to generate comments for C++ methods to the implementation, or reverse engineer comments from the implementation</li> <li>• Whether to update comments in re-generating code from the model</li> <li>• Whether to update the implementation file in re-generating code from the model</li> <li>• The default source code directory location (click on the <b>Browse</b> button - ( ... ))</li> <li>• The default file extensions to read when importing a directory of C++ code</li> <li>• The Code Editor to use (click on the <b>Browse</b> button - ( ... ))</li> <li>• The search path for the implementation file relative to the header file path</li> </ul>	
<b>Statemachine Engineering</b>	<p>In the value fields, use the drop-down arrows to set the options to <b>True</b> or <b>False</b>; these options apply to generating code from State Machine models in the <b>current</b> model only:</p> <ul style="list-style-type: none"> <li>• <b>Use the new Statemachine Template</b> - set to <b>True</b> to use the code generation templates from Enterprise Architect Release 11 and later, set to <b>False</b> to apply the EASL Legacy templates; also set the <b>C++ Version</b> to <b>ANSI</b></li> <li>• <b>Generate Trace Code</b> - set to <b>True</b> to generate Trace code, <b>False</b> to omit it</li> </ul>	<a href="#">Legacy State Machine Templates</a> <sup>2126</sup>

<b>Collection Classes</b>	Click on this button to open the Collection Classes for Association Roles dialog, through which you specify the collection class definitions for Association connectors.	<a href="#">Collection Classes</a> 
---------------------------	--	--

Learn more

- [C++ Conventions](#) 


**11.9.6.7 Delphi Options**

If you intend to generate **Delphi** code from your model, you can configure the code generation options using the **Delphi Specifications** page of the Options dialog to:

- Indicate a default file extension
- Indicate a default source directory
- Set a default directory for opening and saving Delphi source code

Access   **Tools | Options | Source Code Engineering | Delphi**

Options

Field	Action	See also
<b>Disable Language</b>	Leave this checkbox unselected to support Delphi code generation. Select this option to <b>disable</b> Delphi code support.	
<b>Options for the current model</b>	Specifies the options used in the current model. These options affect all users of the current model; however, they do not apply to other models.	
<b>Options for the current user</b>	Specifies the options used for the current user; these options apply to all models that are accessed by the user.	
<b>Collection Classes</b>	Click on this button to open the Collection Classes for Association Roles dialog, through which you specify the collection class definitions for Association connectors.	<a href="#">Collection Classes</a> 

Learn more

- [Delphi Properties](#) 

### 11.9.6.7.1 Delphi Properties

Enterprise Architect has comprehensive support for Delphi properties. These are implemented as Tagged Values, with a specialized property editor to help create and modify Class properties. By using the **Feature Visibility** element context menu option, you can display the 'tags' compartment that contains the properties. Imported Delphi Classes with properties have this feature automatically made visible for your convenience.

#### Manually activate the property editor

- In the selected Class set the code generation language to **Delphi**
- Right-click on the Class and select the **Delphi Properties** context menu option to open the editor

Using the Delphi Properties editor, you can build properties quickly and simply; from here you can:

- Change the name and scope (only **Public** and **Published** are currently supported)
- Change the property type (the drop-down list includes all defined Classes in the project)
- Set the Read and Write information (the drop-down lists have all the attributes and operations from the current Class; you can also enter free text)
- Set **Stored** to **True** or **False**
- Set the Implements information
- Set the default value, if one exists

#### Notes

- When you use the Create Property dialog from the Attribute screen, the system generates a pair of Get and Set functions together with the required property definition as Tagged Values; you can manually edit these Tagged Values if required
- Public properties are displayed with a '+' symbol prefix and published with a '^'
- When creating a property in the Create Property Implementation dialog (accessed through the Attributes dialog), you can set the scope to **Published** if the property type is **Delphi**
- Only **Public** and **Published** are supported
- If you change the name of a property and forward engineer, a new property is added, but you must manually delete the old one from the source file

### 11.9.6.8 Java Options



If you intend to generate **Java** code from your model, you can configure the code generation options using the **Java Specifications** page of the Options dialog to:

- Set a default directory for opening and saving Java source code
- Indicate a default file extension
- Specify a default 'Get' prefix
- Specify a default 'Set' prefix
- Define additional Collection Classes - to define custom collection classes, which can be simple substitutions (such as `CArray<#TYPE#>`) or a mix of other strings and substitutions (such as `Cmap<CString,LPCTSTR,#TYPE#*,#TYPE#*>`); the following collection classes are defined by default:
  - `HashSet<#TYPE#>;Map<String,#TYPE#>;`

Access   **Tools | Options | Source Code Engineering | Java**



Options

Field	Action	See also
<b>Disable Language</b>	<p>Leave this checkbox unselected to support Java code generation.</p> <p>Select this checkbox to <b>disable</b> Java code support.</p>	
<b>Options for the current model</b>	<p>In the value fields, specify the options that affect all users of the <b>current</b> model; the:</p> <ul style="list-style-type: none"> <li>• Default file extension for the code files</li> <li>• The default Get and Set prefixes</li> <li>• The default and additional Collection Classes</li> </ul> <p>These options do not apply to <b>other</b> models.</p>	
<b>Options for the current user</b>	<p>In the value fields, specify the options that apply under your own user ID in <b>all</b> models that you access; the:</p> <ul style="list-style-type: none"> <li>• Default attribute type to create (select from the drop-down list)</li> <li>• Default source code directory location (click on the <b>Browse</b> button - ( ... ))</li> <li>• Code Editor to use (click on the <b>Browse</b> button - ( ... ))</li> </ul>	
<b>Statemachine Engineering</b>	<p>In the value fields, use the drop-down arrows to set the options to <b>True</b> or <b>False</b>; these options apply to generating code from State Machine models in the <b>current</b> model only:</p> <ul style="list-style-type: none"> <li>• <b>Use the new Statemachine Template</b> - set to <b>True</b> to use the code generation templates from Enterprise Architect Release 11 and later, set to <b>False</b> to apply the EASL Legacy templates</li> <li>• <b>Generate Trace Code</b> - set to <b>True</b> to generate Trace code, <b>False</b> to omit it</li> </ul>	<a href="#">Legacy State Machine Templates</a>  <sup>2126</sup>
<b>Collection Classes</b>	<p>Click on this button to open the Collection Classes for Association Roles dialog, through which you specify the collection class definitions for Association connectors.</p>	<a href="#">Collection Classes</a>  <sup>2259</sup>

Learn more

- [Java Conventions](#)  <sup>2096</sup>

**11.9.6.9 Python Options**

If you intend to generate **Python** code from your model, you can configure the code generation options using the **Python Specifications** page of the Options dialog to:

- Indicate the default file extension(s)
- Specify the default source directory to be used

- Specify the editor used to write code

**Access**   **Tools | Options | Source Code Engineering > Python**

#### Options

Field	Action	See also
<b>Disable Language</b>	Leave this checkbox unselected to support Python code generation. Select this option to <b>disable</b> Python code support.	
<b>Options for the current model</b>	Specifies the options used in the current model. These options affect all users of the current model; however, they do not apply to other models.	
<b>Options for the current user</b>	Specifies the options used for the current user; these options apply to all models that are accessed by the user.	

#### 11.9.6.10 *PHP Options*

If you intend to generate **PHP** code from your model, you can configure the code generation options using the **PHP Specifications** page of the Options dialog to:

- Specify the extension to be used when creating files for PHP source
  - Define a semi-colon separated list of extensions to look at when doing a directory code import for PHP
  - Set a default directory for opening and saving PHP source code
  - Specify the version of PHP to generate

**Access**   **Tools | Options | Source Code Engineering | PHP**

#### Options

Field	Action	See also
<b>Disable Language</b>	Leave this checkbox unselected to support PHP code generation. Select this option to <b>disable</b> PHP code support.	
<b>Options for the current model</b>	Specifies the options used in the current model. These options affect all users of the current model; however they do not apply to other models.	
<b>Options for the current user</b>	Specifies the options used for the current user; these options apply to all models that are accessed by the user.	

--	--	--

### Learn more

- [Import a Directory Structure](#) <sup>[2142]</sup>

## 11.9.6.11 SystemC Options

If you intend to generate **SystemC** code from your model, you can configure the code generation options using the **SystemC** page of the Options dialog to:

- Indicate default file extension(s)
- Specify a default source directory
- Specify an editor for changing code

**Access**   **Tools | Options | Source Code Engineering | SystemC**

### Options

Field	Action	See also
<b>Disable Language</b>	Leave this checkbox unselected to support SystemC code generation. Select this option to <b>disable</b> SystemC code support.	
<b>Options for the current model</b>	Specifies the options used in the current model. These options affect all users of the current model; however, they do not apply to other models.	
<b>Options for the current user</b>	Specifies the options used for the current user; these options apply to all models that are accessed by the user.	
<b>Collection Classes</b>	Click on this button to open the Collection Classes for Association Roles dialog, through which you specify the collection class definitions for Association connectors.	<a href="#">Collection Classes</a> <sup>[2258]</sup>

## 11.9.6.12 VB.NET Options

If you intend to generate **VB.NET** code from your model, you can configure the code generation options using the **VB.NET Specifications** page of the Options dialog to:

**Access**   **Tools | Options | Source Code Engineering | VB.Net**

### Use to

- Indicate default file extension(s)

- Specify a default source directory
- Specify an editor for changing code

### Options

Field	Action	See also
<b>Disable Language</b>	Leave this checkbox unselected to support VB.NET code generation. Select this option to <b>disable</b> VB.NET code support.	
<b>Options for the current model</b>	Specifies the options used in the current model. These options affect all users of the current model; however, they do not apply to other models.	
<b>Options for the current user</b>	Specifies the options used for the current user; these options apply to all models that are accessed by the user.	
<b>Collection Classes</b>	Click on this button to open the Collection Classes for Association Roles dialog, through which you specify the collection class definitions for Association connectors.	<a href="#">Collection Classes</a> <small>2259</small>

### 11.9.6.13 Verilog Options

If you intend to generate **Verilog** code from your model, you can configure the code generation options using the **Verilog** page of the Options dialog to:

- Indicate default file extension(s)
- Specify a default source directory
- Specify an editor for changing code

**Access**   [Tools](#) | [Options](#) | [Source Code Engineering](#) | [Verilog](#)

### Options

Field	Action	See also
<b>Disable Language</b>	Leave this checkbox unselected to support Verilog code generation. Select this option to <b>disable</b> Verilog code support.	
<b>Options for the current model</b>	Specifies the options used in the current model. These options affect all users of the current model; however, they do not apply to other models.	

<b>Options for the current user</b>	Specifies the options used for the current user; these options apply to all models that are accessed by the user.	
<b>Collection Classes</b>	Click on this button to open the Collection Classes for Association Roles dialog, through which you specify the collection class definitions for Association connectors.	<a href="#">Collection Classes</a> <small>[2259]</small>

#### 11.9.6.14 VHDL Options

If you intend to generate **VHDL** code from your model, you can configure the code generation options using the **VHDL** page of the Options dialog to:

- Indicate default file extension(s)
- Specify a default source directory
- Specify an editor for changing code

**Access** [Tools | Options | Source Code Engineering | VHDL](#)

#### Options

Field	Action	See also
<b>Disable Language</b>	Leave this checkbox unselected to support VHDL code generation. Select this option to <b>disable</b> VHDL code support.	
<b>Options for the current model</b>	Specifies the options used in the current model. These options affect all users of the current model; however, they do not apply to other models.	
<b>Options for the current user</b>	Specifies the options used for the current user; these options apply to all models that are accessed by the user.	
<b>Collection Classes</b>	Click on this button to open the Collection Classes for Association Roles dialog, through which you specify the collection class definitions for Association connectors.	<a href="#">Collection Classes</a> <small>[2259]</small>

#### 11.9.6.15 Visual Basic Options


If you intend to generate **Visual Basic** code from your model, you can configure the code generation options using the **VB Specifications** page of the Options dialog to:

- Indicate the default file extension when reading/writing
- Specify the default Visual Basic version
- Indicate the MTS transaction mode for MTS objects

- Specify if a Class uses Multi use (**true** or **false**)
- Specify if a Class uses the **Persistable** property
- Indicate data binding behaviours
- Set the global namespace
- Set the *Exposed* attribute
- Indicate if the *Creatable* attribute is **true** or **false**

**Access**   **Tools | Options | Source Code Engineering | Visual Basic**

#### Options

Field	Action	See also
<b>Disable Language</b>	Leave this checkbox unselected to support Visual Basic code generation.  Select this option to <b>disable</b> Visual Basic code support.	
<b>Options for the current model</b>	Specifies the options used in the current model.  These options affect all users of the current model; however, they do not apply to other models.	
<b>Options for the current user</b>	Specifies the options used for the current use; these options apply to all models that are accessed by the user.	
<b>Collection Classes</b>	Click on this button to open the Collection Classes for Association Roles dialog, through which you specify the collection class definitions for Association connectors.	<a href="#">Collection Classes</a> 

#### 11.9.6.16 MDG Technology Language Options

If you have loaded an MDG Technology that specifies a code module into your *Sparx Systems > EA > MDG Technologies* folder, the language is included in the Source Code Engineering list on the Options dialog. The language is only listed on the Options dialog if an MDG Technology file actually uses it in your model.

**Access**   **Tools | Options | Source Code Engineering | MDG**

#### Options

Field	Action	See also
<b>Default Extension</b>	Default extension for generated source files; shown if the option is in the technology.  This is saved per project.	

<b>Import File Extensions</b>	Default folder to import source files from; shown if the technology supports namespaces. This is saved once for all projects.	
<b>Generate Namespaces</b>	Indicates if namespaces are generated or not.	
<b>Default Source Directory</b>	The default directory to save generated source files. This is always shown.	
<b>Editor</b>	Indicates the editor that is used to edit source files.	
<b>Att Type</b>	Indicates the default attribute type.	

**Notes**

- These options are set in the technology inside the `<CodeOptions>` tag of a code module, as follows:

```
<CodeOption name="DefaultExtension">.rb</CodeOption>
```

**Learn more**

- [MDG Technology](#)<sup>[1527]</sup>
- [Adding Code Modules in MDG Technologies](#)<sup>[1552]</sup>

**11.9.6.17 Reset Options**

Enterprise Architect stores some of the options for a Class when it is first created. Some are global; for example, `$LinkClass` is stored when you first create the Class, so it won't automatically pick up the global change in the Options dialog in existing Classes. You must modify the options for the existing Class.

**Modify options for a single Class**

Step	Action	See also
1	Right-click on the Class to change, and select the <b>Generate Code</b> context menu option. The Generate Code dialog displays.	
2	Click on the <b>Advanced</b> button. The Object Options dialog displays.	
3	Click on the <b>Attributes/Operations</b> button.	

Step	Action	See also
4	Change the options, and click on the <b>Close</b> button to apply the changes.	

**Modify options for all Classes within a Package**

Step	Action	See also
1	Right-click on the package in the Project Browser. The context menu displays.	
2	Select the <b>Code Engineering   Reset Options for this Package</b> menu option. The Manage Code Generation dialog displays.	
3	Reset the required defaults for each existing Class.	
4	Click on the <b>OK</b> button to apply the changes.	



## 11.10 Code Template Framework Tool

The Code Template Framework (CTF) is used during forward engineering of UML models. The CTF enables you to:

- Generate source code from UML models
- Customize the way in which Enterprise Architect generates source code
- Forward engineer languages not specifically supported by Enterprise Architect

### Topics


Topic	Detail	See also
<b>Default Templates</b>	Default Code Templates are built into Enterprise Architect for forward engineering supported languages.	<a href="#">Code and Transform Templates</a> <sup>[1632]</sup> <a href="#">Base Templates</a> <sup>[1633]</sup>
<b>Code Template Editor</b>	A Code Template Editor is provided for creating and maintaining user-defined Code Templates.	<a href="#">Code Template Editor</a> <sup>[1641]</sup>
<b>Customizing Code Templates</b>	Descriptions of the template syntax and the macros and functions you can use to control the effects of the templates.	<a href="#">Code Template Framework</a> <sup>[1631]</sup>
<b>Synchronize Code</b>	A subset of the default Code Templates to synchronize code.	<a href="#">Synchronize Code</a> <sup>[1638]</sup>

## 11.11 ICONIX

### Topics

Topic	Detail	See also
<b>Abstract</b>	<p>The following text is derived from the <b>ICONIX</b> entry in the online Wikipedia:</p> <p><i>The ICONIX Process is a minimalist, streamlined approach to Use Case driven UML modeling that uses a core subset of UML diagrams and techniques to provide thorough coverage of object-oriented analysis and design. Its main activity is robustness analysis, a method for bridging the gap between analysis and design. Robustness analysis reduces the ambiguity in use case descriptions, by ensuring that they are written in the context of an accompanying domain model. This process makes the use cases much easier to design, test and estimate.</i></p> <p>The ICONIX Process was developed by Doug Rosenberg; for more information on ICONIX, refer to <i>ICONIX Software Engineering Inc.</i></p>	<a href="#">ICONIX</a> (Online Resource)
<b>ICONIX in Enterprise Architect</b>	<p>Enterprise Architect enables you to develop models under ICONIX quickly and simply, through use of an MDG Technology integrated with the Enterprise Architect installer.</p> <p>The ICONIX facilities are provided in the form of:</p> <ul style="list-style-type: none"> <li>• A set of ICONIX pages in the Toolbox</li> <li>• ICONIX element and relationship entries in the Toolbox Shortcut Menu and Quick Linker</li> </ul> <p>To further help you develop and manage a project under ICONIX, Enterprise Architect also provides a white paper on the ICONIX Roadmap.</p> <p>In addition, Enterprise Architect has an alternative visual layout specific to ICONIX.</p>	<a href="#">Toolbox Shortcut Menu</a> <sup>[799]</sup> <a href="#">Quick Linker</a> <sup>[896]</sup> <a href="#">ICONIX Roadmap</a> (Online Resource)
<b>ICONIX Toolbox Pages</b>	<p>Within the Toolbox, Enterprise Architect provides ICONIX versions of the pages for UML Analysis, Use Case, Class, Interaction (Sequence), Activity and Custom diagrams (which often form the basis for <i>Robustness diagrams</i>).</p> <p>Compared to the standard Toolbox pages, these have slightly different element and relationship sets; you can access them by either:</p> <ul style="list-style-type: none"> <li>• Selecting the <b>More tools   ICONIX   &lt;Diagram Type&gt;</b> menu option for a specific Toolbox page</li> <li>• Selecting the <b>ICONIX</b> option in the drop-down field of the Default Tools toolbar, which adds all six pages to the Toolbox; the first page and the Common page are expanded, and the others are closed up</li> </ul>	<a href="#">Analysis Diagram</a> <sup>[1807]</sup> <a href="#">Use Case Diagram</a> <sup>[1207]</sup> <a href="#">Class Diagram</a> <sup>[1184]</sup> <a href="#">Sequence Diagram</a> <sup>[1249]</sup> <a href="#">Activity Diagram</a> <sup>[1199]</sup> <a href="#">Custom Diagram</a> <sup>[1796]</sup>

Topic	Detail	See also
		
<b>ICONIX Layout</b>	<p>The ICONIX layout re-organizes the Enterprise Architect work area, opening the:</p> <ul style="list-style-type: none"> <li>• Toolbox on the right hand side of the screen (follow the instructions above to display the ICONIX pages)</li> <li>• The Learning Center window auto-hidden in the top right of the screen</li> <li>• Project Browser window in the top left of the screen, and</li> <li>• Notes, Properties and Tagged Values windows nested on the bottom left of the screen</li> </ul>	<a href="#">Workspace Layout Toolbar</a> <sup>[149]</sup>

Topic	Detail	See also
	To apply this layout, select the <b>Basic - Reversed</b> option in the drop-down list in the Workspace Layout toolbar.	
<b>Disable ICONIX</b>	<p>If you prefer not to use ICONIX in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog (<b>Settings   MDG Technologies</b>).</p> <p>This does not affect the ICONIX layout, which you can switch back to your own layout or the Enterprise Architect default layout using the <b>View   Workspaces and Commands</b> menu option.</p>	<a href="#">MDG Technologies</a> 

## 11.12 GoF Patterns

You can develop diagrams from GoF patterns quickly and simply, through use of an MDG Technology integrated with the Enterprise Architect installer.

*Gang of Four (GoF) Patterns are 23 classic software design patterns providing recurring solutions to common problems in software design. They were developed by Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, often referred to as the Gang of Four. The patterns are defined in the book Design Patterns: Elements of Reusable Object-Oriented Software (Gamma et al., ISBN 0-201-63361-2).*

The above text is derived from the *Design Patterns* entry in the online Wikipedia. For further information on the concepts of GoF Patterns, refer to the Wikipedia item and its linked sources.

### GoF Patterns in Enterprise Architect

Topic	Detail	See also
<b>GoF Pattern Facilities</b>	<p>The GoF Patterns are provided in the form of:</p> <ul style="list-style-type: none"><li>• GoF Behavioral Patterns, GoF Creational Patterns and GoF Structural Patterns pages in the Toolbox</li><li>• Gang of Four Pattern entries in the Toolbox Shortcut Menu</li></ul> <p><b>GoF Toolbox Pages</b></p> <p>You can access the GoF Pattern pages of the Toolbox through the <b>More tools   GoF Patterns</b> menu option; the following icons are available:</p>	<a href="#">Toolbox Shortcut Menu</a> <sup>[79]</sup>

Topic	Detail	See also
	 <p>When you drag one of the pattern elements onto a new diagram, the Add Pattern GoF &lt;pattern group&gt;&lt;pattern type&gt; dialog displays; if necessary, modify the action and/or default for the component elements, then click on the <b>OK</b> button to create a diagram based on the pattern.</p>	
<b>Disable GoF Patterns</b>	<p>If you prefer not to use the GoF Patterns technology in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog (<b>Settings   MDG Technologies</b>).</p>	<a href="#">MDG Technologies</a> <small>147</small>

#### Learn more

- [Design Patterns](#) (Online Resource)

**Part**

---

**XII**

## 12 Systems Engineering

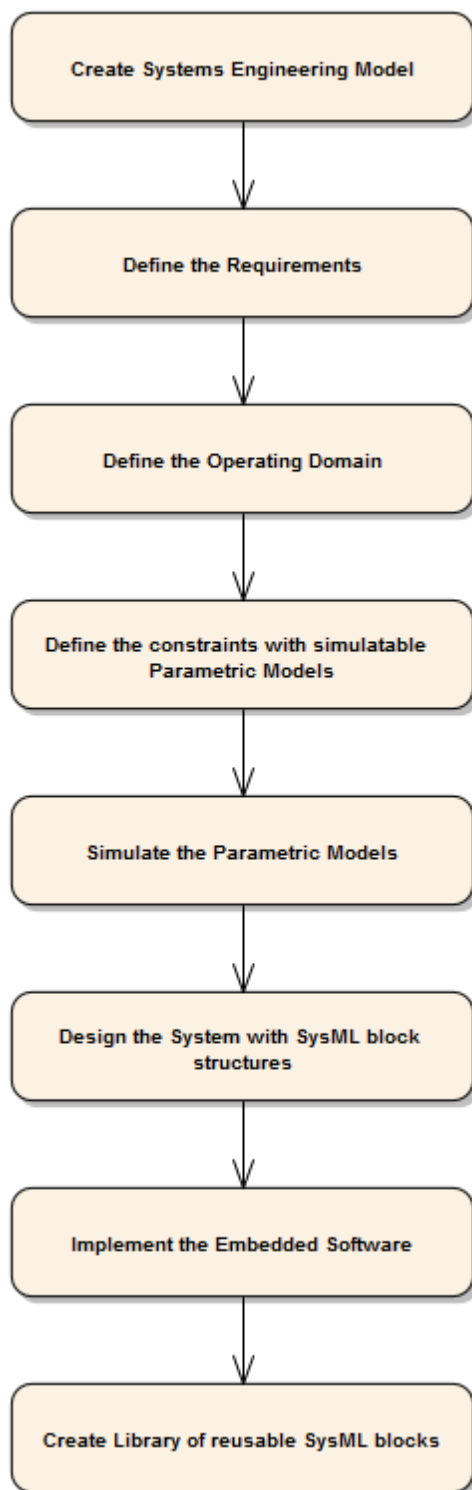
### Topics

Topic	Detail	See also
<b>Usage</b>	<p>To model Systems using <b>SysML</b> in Enterprise Architect, you work through the following steps:</p> <ul style="list-style-type: none"> <li>• <b>Create a Systems Engineering model</b> to develop your system</li> <li>• <b>Create a Requirements model</b> to define the systems requirements and expectations</li> <li>• Create an <b>Operational Domain model</b>, which describes the environment that the system operates within, and the entities it interacts with</li> <li>• <b>Create Constraint models</b> to describe the systems operating characteristics using parametric models</li> <li>• <b>Simulate the parametric models</b> to verify their correctness and obtain the desired characteristic</li> <li>• <b>Design the system's composition</b> using SysML Blocks and Parts</li> <li>• <b>Implement the embedded software</b> using UML Classes and behavioral models</li> <li>• <b>Create a Library of reusable SysML blocks</b>, representing subsystems that can be reused on other projects, and other common Type definitions</li> </ul>	<p><a href="#">SysML</a> <sup>[2294]</sup></p> <p><a href="#">Create a Systems Engineering Model</a> <sup>[2289]</sup></p> <p><a href="#">A SysML Requirements Model</a> <sup>[2325]</sup></p> <p><a href="#">A SysML Operational Domain Model</a> <sup>[2326]</sup></p> <p><a href="#">SysML Parametric Models</a> <sup>[2320]</sup></p> <p><a href="#">Simulating a SysML Model</a> <sup>[2322]</sup></p> <p><a href="#">Compose System Design</a> <sup>[2328]</sup></p> <p><a href="#">Generate from Behaviour Models</a> <sup>[2127]</sup></p> <p><a href="#">Create Re-usable Subsystems</a> <sup>[2330]</sup></p>

### Example

These steps are represented graphically in the following flow:



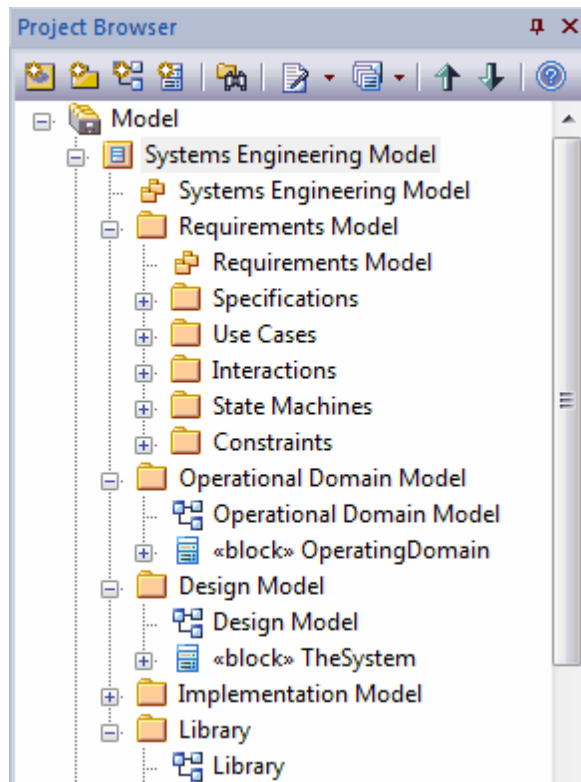


Create a Systems Engineering Model from a template

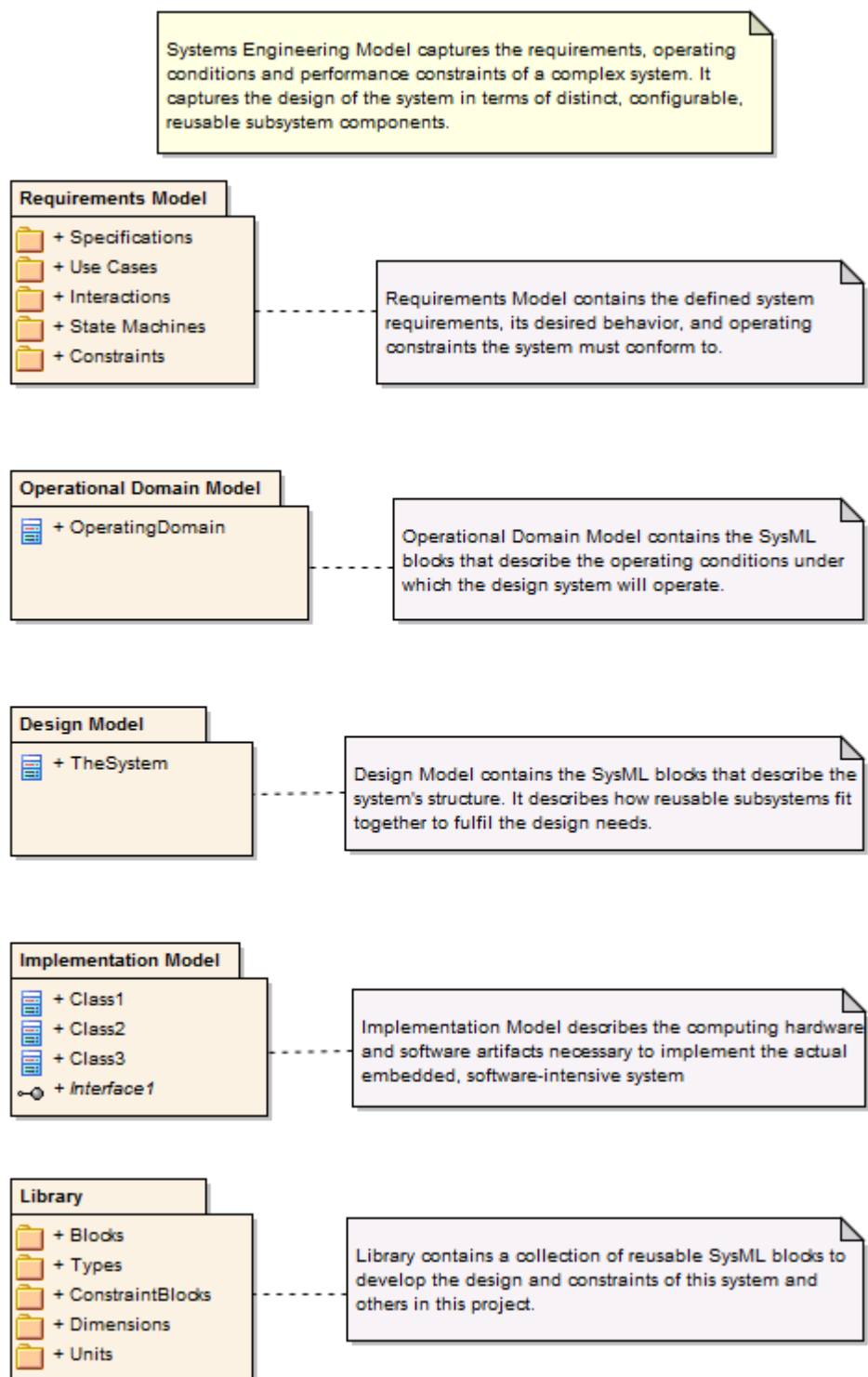
Step	Action	See also
1	<p>In the Project Browser, either:</p> <ul style="list-style-type: none"><li>Click on the <b>New Model From Pattern</b> icon in the toolbar</li><li>Right-click on a model root node and select the <b>Add a New Model using Wizard</b> context menu option, or</li><li>Right-click on a package and select the <b>Add   Add a New Model using Wizard</b> context menu option</li></ul> <p>The Select Model(s) dialog displays.</p>	
2	<p>In the <b>Select From</b> field, click on the drop-down arrow and select <b>Systems Engineering Model</b>.</p> <p>Alternatively, if it is listed in the Technology panel, select the <b>Systems Engineering Model</b> item.</p>	
3	<p>In the Name panel, select the checkbox next to the <b>Systems Engineering Model</b> icon.</p>	
4	<p>Click on the <b>OK</b> button.</p>	

### Example

The following model structure is created in the Project Browser:



The *Systems Engineering Model* diagram, shown below, encapsulates the key components of the Systems Engineering model.



### Notes

- Systems Modeling Language (SysML) is provided with the Systems Engineering edition and Ultimate edition of Enterprise Architect

Learning Center topics

- (Alt+F1) | **Enterprise Architect | Systems Engineering**

## 12.1 Systems Modeling Language (SysML)

To develop SysML models quickly and simply, you can use one of three versions of the **MDG Technology for SysML**, integrated with Enterprise Architect. The three available versions correspond to **SysML 1.1**, **SysML 1.2** and **SysML 1.3**; to avoid confusion, you only need to have **more than** one of these technologies enabled at a time if you are **migrating** your model from an earlier version to a later one.

The following text is derived from the official OMG SysML site of the Object Management Group.

'The OMG Systems Modeling Language (OMG SysML™) is a general-purpose graphical modeling language for specifying, analyzing, designing, and verifying complex systems that may include hardware, software, information, personnel, procedures, and facilities. The language provides graphical representations with a semantic foundation for modeling system requirements, behavior, structure, and parametrics, which is used to integrate with other engineering analysis models. SysML was developed in response to requirements developed jointly by the OMG and the International Council on Systems Engineering (INCOSE) by the diverse group of tool vendors, end users, academia, and government representatives.'

**Access**   [Diagram](#) | [Diagram Toolbox: More tools](#) | [SysML 1.1](#)  
[Diagram](#) | [Diagram Toolbox: More tools](#) | [SysML 1.2](#)  
[Diagram](#) | [Diagram Toolbox: More tools](#) | [SysML 1.3](#)

### SysML Integration

Facilities	Detail	See also
<b>SysML In Enterprise Architect</b>	<p>The MDG Technology for SysML provides:</p> <ul style="list-style-type: none"> <li>Each of the nine <b>SysML diagram types</b>, accessed through the New Diagram dialog</li> <li>A collection of SysML pages in the Diagram Toolbox that contain the SysML elements and relationships for each of the supported diagram types (see below)</li> <li>SysML element and relationship entries in the Toolbox Shortcut Menu and Quick Linker</li> <li><b>Simulation of SysML Parametric diagrams</b>, which supports engineering analysis of critical system parameters including the evaluation of key metrics such as performance, reliability and other physical characteristics</li> </ul>	<p><a href="#">Add New Diagram</a> [822]</p> <p><a href="#">Toolbox Shortcut Menu</a> [799]</p> <p><a href="#">Quick Linker</a> [896]</p> <p><a href="#">Simulate a SysML Model</a> [2322]</p>
<b>SysML Toolboxes</b>	<p>The MDG Technology for SysML provides Diagram Toolbox pages that you can access through the <b>More Tools</b> button. If you enable SysML as the active technology, you can also open the SysML Toolbox pages by default.</p> <p>The following sets of Toolbox pages are available:</p> <ul style="list-style-type: none"> <li><b>Activity</b> contains the constructs required to construct SysML Activity models</li> <li><b>Block Definition</b> contains the constructs required to design SysML blocks, constraint blocks, data and value types</li> <li><b>Interaction</b> contains the constructs required to construct SysML interactions and Sequence diagrams</li> <li><b>Internal Block</b> contains the constructs required to design SysML block compositions within Internal Block Diagrams</li> </ul>	<p><a href="#">SysML Activity Toolbox</a> [2295]</p> <p><a href="#">SysML Block Definition Toolbox</a> [2306]</p> <p><a href="#">SysML Interaction Toolbox</a> [2306]</p> <p><a href="#">SysML Internal Block Toolbox</a> [2308]</p>

Facilities	Detail	See also
	<ul style="list-style-type: none"> <li>• <b>Model Elements</b> contains the constructs required to build SysML models, package structures and views</li> <li>• <b>Parametrics</b> contains the constructs required to construct SysML Parametric Diagrams using constraint blocks</li> <li>• <b>Requirements</b> contains the constructs required to build SysML Requirements models</li> <li>• <b>State Machine</b> contains the constructs required to build SysML State Machines</li> <li>• <b>Use Case</b> contains the constructs required to build SysML Use Case models</li> </ul> <p>With the Model Elements pages there is a set of SysML Common elements and relationships; these are also provided with the other SysML Toolbox pages if the active technology is set to <b>SysML 1.1</b>, <b>SysML 1.2</b> or <b>SysML 1.3</b> on the Default Tools toolbar.</p>	<a href="#">SysML Model Elements Toolbox</a> <sup>[2310]</sup> <a href="#">SysML Parametrics Toolbox</a> <sup>[2312]</sup> <a href="#">SysML Requirements Toolbox</a> <sup>[2314]</sup> <a href="#">SysML State Machine Toolbox</a> <sup>[2316]</sup> <a href="#">SysML Use Case Toolbox</a> <sup>[2318]</sup>  <a href="#">Default Tools Toolbar</a> <sup>[135]</sup>
<b>Upgrade SysML Models</b>	You can migrate a SysML model (or part of a model) to a later SysML version, using the Automation Interface.	<a href="#">Migrate SysML Model to Later SysML Version</a> <sup>[2331]</sup>
<b>Disable SysML</b>	If you prefer not to use SysML in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog ( <b>Settings   MDG Technologies</b> ).	<a href="#">MDG Technologies</a> <sup>[1477]</sup>

### Notes

- Systems Modeling Language (SysML) is provided with the Systems Engineering and Ultimate editions of Enterprise Architect

### Learn more

- [OMG SysML](#) (Online Resource)

## 12.1.1 SysML Activity Toolbox

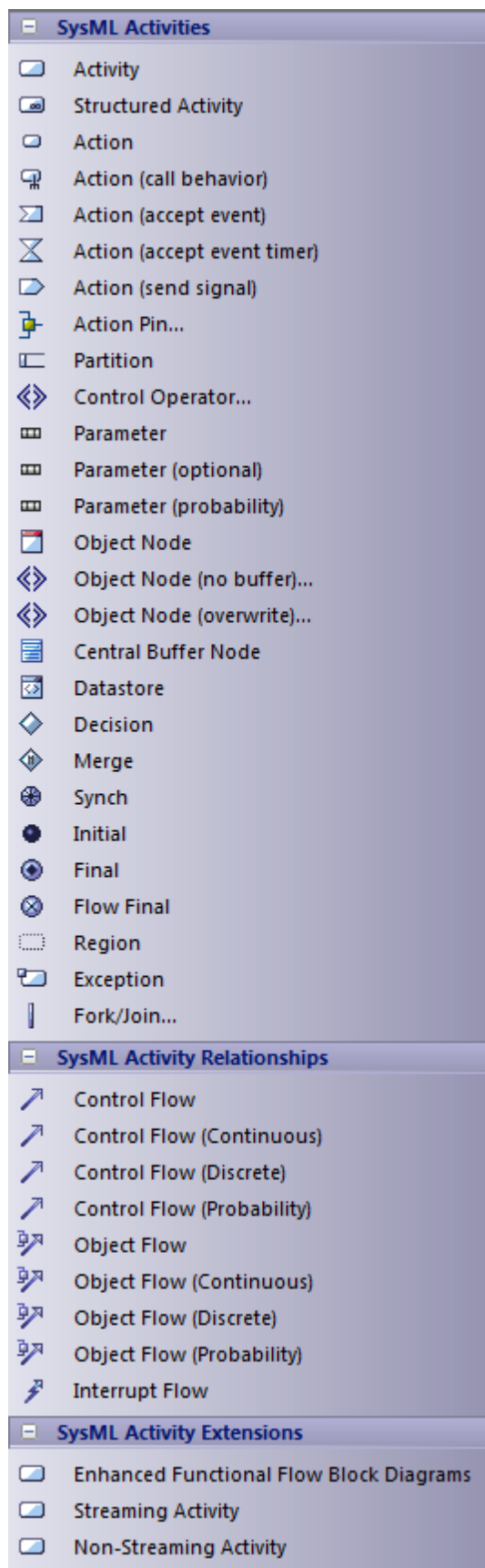
When you are constructing SysML models, you can populate the Activity diagrams using the icons on the SysML Activity pages of the Diagram Toolbox.

The Block that owns the Activity diagram is automatically represented by a diagram frame enclosing the Activity diagram elements. You can:


- Hide the frame by right-clicking on the diagram and selecting the **Hide Diagram Frame** context menu option
- Make the frame selectable to move or resize it, by right-clicking on it and selecting the **Selectable** context menu option
- Create Ports and Parts on the frame and create connectors between them

[Access](#) [Diagram](#) | [Diagram Toolbox: More Tools](#) | [SysML 1.3](#) | [SysML 1.3 Activity](#)





SysML Activity Objects

Page	Item	Action
<b>SysML Activities</b>	<b>Activity</b>	Defines a SysML Block of executable behavior as a UML Activity.
	<b>Structured Activity</b>	Defines a SysML Block of executable behavior as a UML Structured Activity.
	<b>Action</b>	Declares a unit of execution in an Activity as a UML Action.
	<b>Action (call behavior)</b>	Declares a unit of execution that calls another behavior.
	<b>Action (accept event)</b>	Declares a unit of execution that accepts an event raised by the system.
	<b>Action (accept event timer)</b>	Declares a unit of execution that accepts an event raised by a time epoch.
	<b>Action (send signal)</b>	Declares a unit of execution that sends a signal as an event.
	<b>Action Pin</b>	Defines the data values passed out of and into an Action. (See also: <a href="#">Action Pin</a>  )
	<b>Partition</b>	Creates an Activity Partition to group execution elements according to the node responsible for their execution.
	<b>Control Operator</b>	Controls the execution of an Activity.
	<b>Parameter</b>	Provides access to input and output objects within the Activity.
	<b>Parameter (optional)</b>	Defines a parameter whose contents are optional in the Activity's execution.
	<b>Parameter (probability)</b>	Tags a parameter with the probability of the parameter being in use in the Activity.
	<b>Object Node</b>	Declares a variable in the Activity, typed by a ValueType, DataType or Block.

Page	Item	Action
	<b>Object Node (no buffer)</b>	Declares an ObjectNode in an Activity that discards unconsumed tokens.
	<b>Object Node (overwrite)</b>	Declares an ObjectNode in an Activity that overwrites tokens.
	<b>Central Buffer Node</b>	Declares an ObjectNode that stores tokens for consumption throughout the Activity.
	<b>Datastore</b>	Defines permanently stored data. (See also: <a href="#">Datastore</a> <sup>[1293]</sup> .)
	<b>Decision</b>	Creates a branch of control in an Activity, based on a decision.
	<b>Merge</b>	Merges two or more Activity control branches.
	<b>Synch</b>	Establishes a rendezvous point for two or more Activity flows, in order to synchronize their execution in the Activity.
	<b>Initial</b>	Declares the start of an Activity's execution.
	<b>Final</b>	Declares the end of an Activity's execution, and the termination of the Activity.
	<b>Flow Final</b>	Declares the end of an Activity's execution path without terminating the Activity.
	<b>Region</b>	Groups a subset of an Activity into a common execution context.
	<b>Exception</b>	Declares a node of execution that happens outside the normal flow of execution of an Activity.
	<b>Fork/Join</b>	Simultaneously branches / joins a set of Control or Object Flows.
<b>SysML Activity Relationships</b>	<b>Control Flow</b>	Establishes a flow of logic between two Activity nodes.
	<b>Control Flow (Continuous)</b>	Declares a continuous control flow.
	<b>Control Flow</b>	Declares a discrete control flow.

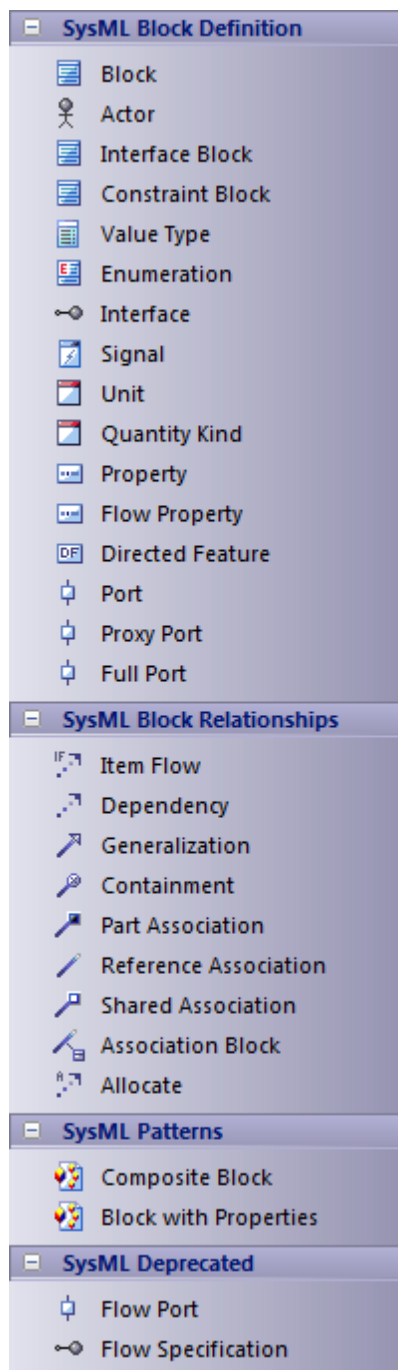
Page	Item	Action
	<b>(Discrete)</b>	
	<b>Control Flow (Probability)</b>	Tags a control flow with a probability of the likelihood of the flow's traversal.
	<b>Object Flow</b>	Establishes a flow of objects (data) between two Activity nodes.
	<b>Object Flow (Continuous)</b>	Declares a continuous object flow.
	<b>Object Flow (Discrete)</b>	Declares a discrete object flow.
	<b>Object Flow (Probability)</b>	Tags an object flow with the probability of the flow's traversal.
	<b>Interrupt Flow</b>	Declares a control flow that interrupts flow within a Region.
<b>SysML Activity Extensions</b>	<b>Enhanced Functional Flow Block Diagrams</b>	Declares an Activity used to contain an Enhanced Functional Flow Block Diagram (EFFBD).
	<b>Streaming Activity</b>	Declares an Activity where the flow of tokens passes through its parameters continuously throughout the Activity's execution.
	<b>Non-Streaming Activity</b>	Declares an Activity where the flow of tokens passes through its parameters at the start of the Activity's execution.

### 12.1.2 SysML Block Definition Toolbox

When you are constructing SysML models, you can populate the Block Definition diagrams using the icons on the SysML Block Definition pages of the Diagram Toolbox.

You can also generate **Property** elements on the Block, based on the Association relationships created for the Block element. These Properties (or Parts) are initially created in the **Project Browser**, but you can quickly render them on the **Internal Block diagram** for the Block.

[Access](#) [Diagram](#) | [Diagram Toolbox: More Tools](#) | [SysML 1.3](#) | [SysML 1.3 Block Definition](#)



### SysML Block Definition Objects

Page	Item	Action
<b>SysML Block Definition</b>	<b>Block</b>	Defines a composite system entity in SysML.
	<b>Actor</b>	Represents a user that interacts with one or more SysML systems.

Page	Item	Action
	<b>Interface Block</b>	A specialized kind of block that has no behaviors or internal parts, which is used to type Proxy Ports (below).
	<b>Constraint Block</b>	Defines a composite constraint as a system of parametric equations.
	<b>ValueType</b>	Defines a SysML quantity, expressed as a measurable dimension with specific units.
	<b>Enumeration</b>	Defines a data type as a set of symbols or values.
	<b>Interface</b>	Defines an element that describes a specification of an interaction point with properties and methods.
	<b>Signal</b>	Defines a SysML message, containing attributes, exchanged between system blocks in an interaction.
	<b>Unit</b>	Represents a standard unit of measure in SysML.
	<b>QuantityKind</b>	Identifies a measurable quantity in SysML.
	<b>Property</b>	Describes the decomposition of a SysML Block in the context of its whole, using instances of reusable SysML Blocks.
	<b>Flow Property</b>	Creates a single kind of Flow element that can flow to or from a block.
	<b>Directed Feature</b>	<p>Generates an <b>operation</b> if dropped on a Block, or a <b>DirectedFeature</b> (Property) element if dropped on the diagram, both of which indicate that the associated Block either owns (<b>provided</b>) the feature, uses (<b>required</b>) the feature owned by another Block, or both (<b>providedrequired</b>).</p> <p>The direction value is defined in the SysML1.3 page of the element Properties dialog, or the Tagged Values page of the Operation Properties dialog.</p>
	<b>Port</b>	Describes a structural interaction point of a SysML Block, which in turn connects interacting parts of a block.
	<b>Proxy Port</b>	Exposes features of the owning Block or its internal parts.

Page	Item	Action
	<b>Full Port</b>	Specifies an element of the system separate from the owning Block or its internal parts.
<b>SysML Block Relationships</b>	<b>Item Flow</b>	Specifies the items that flow across a connector in an interaction point. Used in the same way as UML Information Flows. See <a href="#">Using Information Flows</a> <sup>[141]</sup>
	<b>Dependency</b>	Establishes a traceable relationship describing how one element is dependant upon another.
	<b>Generalization</b>	Describes an element as a specialized descendant of another element, containing additional properties and behavior.
	<b>Containment</b>	Graphically displays ownership of one element within a parent element.
	<b>Part Association</b>	Describes the characteristics of a connection between a SysML Block and its internal parts, such as the multiplicity and type.
	<b>Reference Association</b>	Describes the characteristics of a connection between separate SysML Blocks, such as the multiplicity and type.
	<b>Shared Association</b>	Describes the characteristics of a common connection between SysML Blocks, such as the multiplicity and type.
	<b>Allocate</b>	Connects model elements to formalize a refinement of behavior, structure, constraints or design expectations.
<b>SysML Patterns</b>	<b>Composite Block</b>	A pattern that creates a Composite Block, made up of Blocks related by Aggregation relationships.
	<b>Block with Parts</b>	A pattern that creates a Composite Block made up of nested parts.
<b>SysML Deprecated</b>	<b>Flow Port</b>	Describes what flows in and out of interacting SysML Blocks. See <a href="#">Show Direction on SysML 1.3 Ports</a> <sup>[2306]</sup>
	<b>Flow Specification</b>	Defines a set of flow properties that correspond to individual pieces of a common interaction point.

Page	Item	Action
		See <a href="#">Show Direction on SysML 1.3 Ports</a> <sup>[2306]</sup> .

### Learn more

- [SysML Internal Block Toolbox](#) <sup>[2308]</sup>
- [Generate Properties from Block Associations](#) <sup>[2304]</sup>
- [Block Element Compartments](#) <sup>[2305]</sup>

#### 12.1.2.1 Generate Properties From Block Associations

On a SysML Block diagram, an **Association** between two Block elements has ends that are represented by **properties**. If an Association end is navigable, the property that represents it is owned by the Block element at the other end of the Association.

In Enterprise Architect you can automatically create **Part** elements to represent these owned properties, in one of two ways:

- Create a **directed** Association (**Tools | Options | Links > select Association default = source --> target** checkbox), which generates an anonymous Part on the source Block element
- Open the source Block's internal block diagram, right-click on the diagram and select **Synchronize Structural Elements**

In both cases, the **Part** property is considered to be **bound** to the **Association End** property – they represent the same property, so changing one updates the other, either automatically or at the next synchronization; that is: if you change the Association Source Role name, multiplicity or Aggregation setting, the Part name, multiplicity and isReference setting are updated; if you change the Part details, the Association properties are updated.

If you want to **change** the binding of a Property to an Association, or bind existing Associations and Properties that are not yet bound to each other, right click on the Property in:

- The Project Browser, and select **Add | Bind to Connector Role**, or
- The Internal Block Diagram and select **Advanced | Bind to Connector Role**

In each case the Choose Connector Role to Bind dialog displays, listing the Associations issuing from the parent Block element. Select the Association to bind the Property to, and click on the **OK** button.

If you subsequently **delete** an Association that is bound to a property, when you save the diagram you are prompted to confirm whether to also **Delete** the property or **Keep** it, unbound to a connector. If the property element is locked, it cannot be deleted.

### Notes

- You can create an Association connector on a SysML Block diagram by dragging the **Reference Association** icon from the Toolbox or selecting the **Association** option from the Quick Linker arrow drop-down list
- Associations have direction when it is explicitly selected on the connector Properties, when the connector has assigned Source and Target roles, and/or when the **Association default = source --> target** checkbox is selected on the Options dialog



### 12.1.2.2 Block Element Compartments

On a **SysML Block diagram**, SysML 1.3 **Block** and **Constraint Block** elements can display any of a number of element **compartments** that list their child elements that are not already shown on the diagram. You can select to display or hide these compartments using the Feature Visibility dialog (press **Ctrl+Shift+Y** on the **element**); however, if there are no child elements corresponding to a type of compartment, that compartment is not shown.

#### SysML 1.3 Block element compartments

Compartment Name	Child Element Types Listed
<b>values</b>	Parts that are typed by a « <i>valueType</i> » element.
<b>participants</b>	Parts with the « <i>participantProperty</i> » stereotype.
<b>flow properties</b>	Parts with the « <i>flowProperty</i> » stereotype.
<b>references</b>	Parts for which the <b>isReference</b> Tagged Value is set to <b>true</b> .
<b>constraints</b>	Parts with the « <i>constraintProperty</i> » stereotype.
<b>properties</b>	Parts without a stereotype.
<b>«stereotype»</b>	Parts with stereotypes other than those identified above.
<b>flow ports</b>	Ports with the « <i>flowPort</i> » stereotype.
<b>full ports</b>	Ports with the « <i>fullPort</i> » stereotype.
<b>proxy ports</b>	Ports with the « <i>proxyPort</i> » stereotype.
<b>ports</b>	Any other Ports not listed in the previous compartments.

#### SysML 1.3 Constraint Block element compartments

Compartment Name	Child Element Types Listed
<b>constraint properties</b>	Parts with the « <i>constraintProperty</i> » stereotype.

Compartment Name	Child Element Types Listed
<b>parameters</b>	Any other types of Part.

#### Learn more

- [Part](#)<sup>[1383]</sup>
- [Port](#)<sup>[1384]</sup>
- [Feature Visibility](#)<sup>[845]</sup>
- [Compartments](#)<sup>[955]</sup>

### 12.1.2.3 Show Direction on SysML 1.3 Ports

In SysML 1.3 you can show Ports, Full Ports and Proxy Ports with arrows indicating the flow direction.

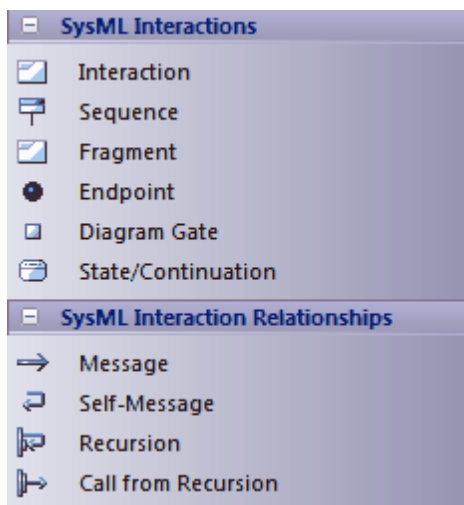
#### Create a new port with direction arrows

Step	Action	See Also
1	Create a <b>Port</b> , <b>Full Port</b> or <b>Proxy Port</b> from the <b>Block Definition</b> toolbox for SysML 1.3.	<a href="#">SysML Block Definition Toolbox</a> <sup>[2300]</sup>
2	Right-click on the Port, select <b>Advanced   Set Property Type</b> . The <b>Select &lt;Item&gt; Dialog</b> opens.	<a href="#">Select &lt;Item&gt; Dialog</a> <sup>[994]</sup>
3	Select a <b>Block</b> or <b>Interface Block</b> that owns one or more <b>Flow Property</b> elements. Click <b>OK</b> .  The Port will automatically display arrows based upon the aggregated 'direction' Tagged Values of all the Flow Property elements owned by the Block or Interface Block that is set as the Property Type.  For example, if the Block contains an 'in' Flow Property and an 'out' Flow Property, the Port typed with this Block will display an 'inout' arrow.	

### 12.1.3 SysML Interaction Toolbox

When you are constructing SysML models, you can populate the Interaction and Sequence diagrams using the icons on the SysML Interaction pages of the Diagram Toolbox.

Access   **Diagram | Diagram Toolbox: More Tools | SysML 1.3 | SysML1.3 Interaction**



### SysML Interaction Objects

Page	Item	Action
<b>SysML Interactions</b>	<b>Interaction</b>	Defines a SysML Block of executable behavior as a UML Interaction.
	<b>Sequence</b>	References an instance of a SysML Block as a Lifeline in the Interaction.
	<b>Fragment</b>	Declares a portion of an interaction as a group with specific behavior semantics.
	<b>Endpoint</b>	Creates an exit point for the Interaction.
	<b>Diagram Gate</b>	Creates an endpoint for the interaction, which bridges between nested interactions.
	<b>State/Continuation</b>	Constrains the Interaction with assertions of the state that the lifeline is expected to be in.
<b>SysML Interactions Relationships</b>	<b>Message</b>	Describes a message exchange between two lifelines in an Interaction.
	<b>Self-Message</b>	Describes a message exchange between a lifeline and itself in an Interaction.
	<b>Recursion</b>	Describes a recursive message exchange between a lifeline and

Page	Item	Action
		itself in an Interaction.
	<b>Call from Recursion</b>	Describes a message exchange between two lifelines within a recursive exchange.

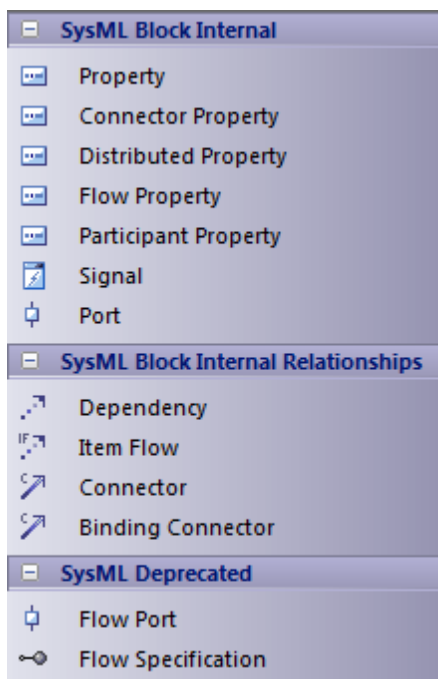
### 12.1.4 SysML Internal Block Toolbox

When you are constructing SysML models, you can populate the Internal Block diagrams using the icons on the SysML Block Internal pages of the Diagram Toolbox.

The Block that owns the Internal Block diagram is automatically represented by a diagram frame enclosing the Internal Block diagram elements. You can:

- Hide the frame by right-clicking on the diagram and selecting the **Hide Diagram Frame** context menu option
- Make the frame selectable to move or resize it, by right-clicking on it and selecting the **Selectable** context menu option
- Create Ports and Parts on the frame and create connectors between them
- Generate **Property elements** on the diagram inside the Block frame, based on the **Associations** that the **Block element** has on the **Block diagram**

**Access** [Diagram](#) | [Diagram Toolbox: More Tools](#) | [SysML 1.3](#) | [SysML 1.3 Internal Block](#)



#### SysML Internal Block Objects

Page	Item	Action
<b>SysML Block Internal</b>	<b>Property</b>	Describes the decomposition of a SysML Block in the context of its whole using instances of reusable SysML Blocks.
	<b>Connector Property</b>	Creates an instance of an Association Block.
	<b>Distributed Property</b>	Creates a stereotype of Property, to apply a probability distribution to the values of the property.
	<b>Flow Property</b>	Creates a single kind of Flow element that can flow to or from a block.
	<b>Participant Property</b>	Creates the end of a connector owned by an Association Block.
	<b>Signal</b>	Defines a SysML message, containing attributes, exchanged between system Blocks in an interaction.
	<b>Port</b>	Describes a structural interaction point of a SysML Block which, in turn, connects between interacting parts of a block.
<b>SysML Block Internal Relationships</b>	<b>Dependency</b>	Establishes a traceable relationship describing how one element is dependant upon another.
	<b>Item Flow</b>	Specifies the items that flow across a connector in an interaction point. Used in the same way as UML Information Flows. See <a href="#">Using Information Flows</a> <sup>[141]</sup>
	<b>Connector</b>	Establishes Communication links between parts. (See also: <a href="#">Connector</a> <sup>[1402]</sup> .)
	<b>Binding Connector</b>	Establishes a connection between two parts in a system decomposition.
<b>SysML Deprecated</b>	<b>Flow Port</b>	Describes what flows in and out of interacting SysML Blocks.
	<b>Flow Specification</b>	Defines a set of flow properties that correspond to individual pieces of a common interaction point.

[Learn more](#)

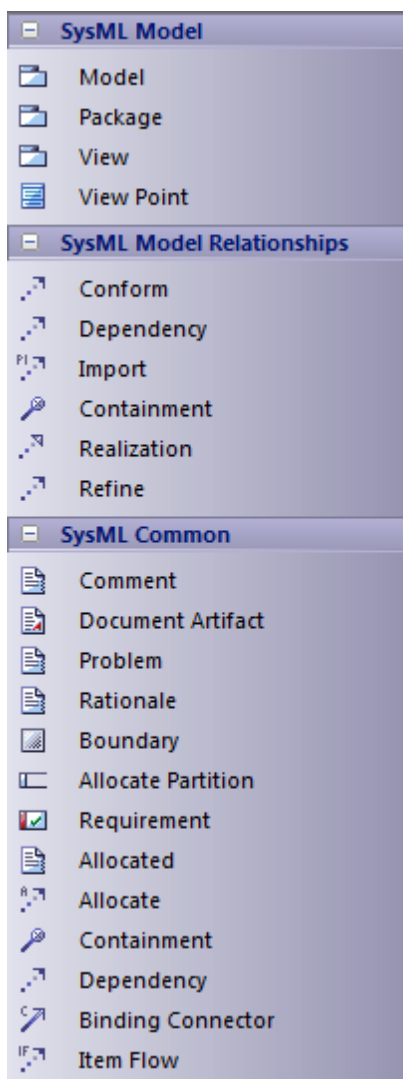
- [Generate Properties From Block Associations](#) <sup>[2304]</sup>

### 12.1.5 SysML Model Elements Toolbox

When you are constructing SysML models, you can populate the diagrams with Model, Package and View elements using the icons on the **SysML Model** pages of the Diagram Toolbox.

The SysML Model toolbox includes a page of **SysML Common** element and relationship icons. You can add this page to all Diagram Toolboxes so that it is always available regardless of what type of diagram you are using; to do this, set the SysML 1.3 Technology to Active (**Settings | MDG Technologies: SysML 1.3: Set Active**).

**Access**   **Diagram | Diagram Toolbox: More Tools | SysML 1.3 | SysML 1.3 Model Elements**



#### SysML Model Element Objects

Page	Item	Action
<b>SysML Model</b>	<b>Model</b>	Creates a Package containing a SysML Model.
	<b>Package</b>	Groups model constructs in a single unit of containment.
	<b>View</b>	Creates a stereotyped Package that defines a SysML View of a system, from the perspective of a SysML View Point.
	<b>View Point</b>	Creates a stereotyped Class that defines a SysML View Point, which specifies the rules and conventions for the construction and use of Views.
<b>SysML Model Relationships</b>	<b>Conform</b>	Establishes a conformance dependency of a View to the defining View Point.
	<b>Dependency</b>	Establishes a traceable relationship describing how one element is dependant upon another.
	<b>Import</b>	Represents a reuse of elements from one model package in another.
	<b>Containment</b>	Graphically displays ownership of one element within a parent one.
	<b>Realization</b>	Identifies a design fulfillment of a specification between elements.
	<b>Refine</b>	Represents a refinement of one element by another.
<b>SysML Common</b>	<b>Comment</b>	Creates a textual annotation that can be attached to a set of elements of any other type.  The attachment is created separately, using a Notelink connector.
	<b>Document Artifact</b>	Attaches a linked document to the diagram by associating this element with the document.
	<b>Problem</b>	A stereotyped Comment that documents the failure of model elements to satisfy a requirement.
	<b>Rationale</b>	A stereotyped Comment that documents the justification for decisions.

Page	Item	Action
	<b>Boundary</b>	Defines a conceptual boundary, to visually group logically related elements.
	<b>Allocate Partition</b>	A stereotyped Activity Partition that contains elements deemed to be allocated to the classifier of the partition.
	<b>Requirement</b>	Specifies the capabilities of the system, or the conditions that it should satisfy.
	<b>Allocated</b>	A stereotyped Comment that defines the source element being allocated to the target element in an Allocate relationship.
	<b>Allocate</b>	<p>A stereotyped Abstraction that relates model elements to formalize a refinement of behavior, structure, constraints or design expectations.</p> <p>The Allocate relationship points from the element being allocated to the element that is the target of the allocation.</p> <p>The system provides an Allocations search that lists all Allocate abstractions in tabular format, showing the 'To' and 'From' elements. Select <b>Edit   Find in Project   SysML 1.3   Allocations</b>.</p>
	<b>Containment</b>	Graphically displays ownership of one element within a parent one.
	<b>Dependency</b>	Establishes a traceable relationship describing how one element is dependant upon another.
	<b>Binding Connector</b>	A stereotyped Connector that establishes a connection between two parts in a system decomposition.
	<b>Item Flow</b>	<p>A stereotyped Information Flow that specifies the items that flow across a connector in an interaction point.</p> <p>See <a href="#">Using Information Flows</a> <sup>[141]</sup></p>

### 12.1.6 SysML Parametrics Toolbox

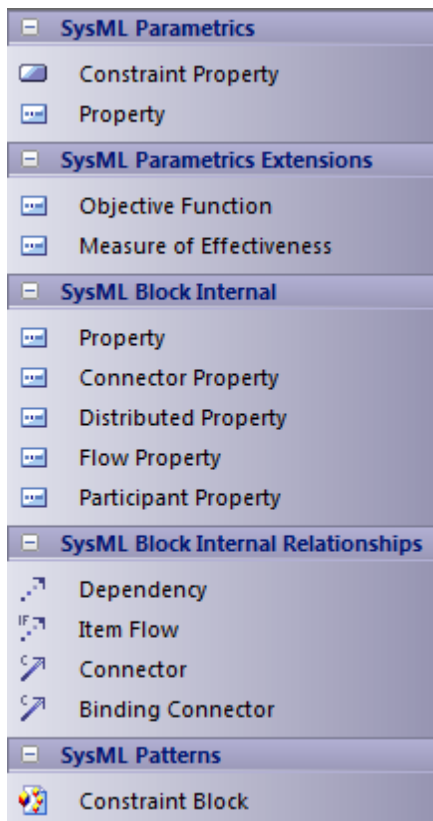
When you are constructing SysML models, you can populate the SysML Parametric diagrams with constraint blocks, using the icons on the SysML Parametrics pages of the Diagram Toolbox.

The Block that owns the Parametric diagram is automatically represented by a diagram frame enclosing the Parametric diagram elements. You can:



- Hide the frame by right-clicking on the diagram and selecting the **Hide Diagram Frame** context menu option
- Make the frame selectable to move or resize it, by right-clicking on it and selecting the **Selectable** context menu option
- Create Ports and Parts on the frame and create connectors between them

**Access**   **Diagram** | **Diagram Toolbox: More Tools** | **SysML 1.3** | **SysML 1.3 Parametrics**



### SysML Parametrics Objects

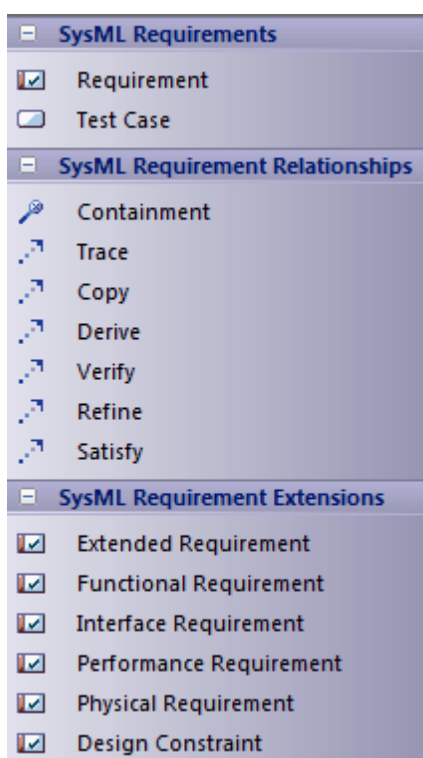
Page	Item	Action
<b>SysML Parametrics</b>	<b>Constraint Property</b>	Instantiates a Constraint Block for use in a Parametric diagram.
	<b>Property</b>	Defines a SysML property typed by a DataType, ValueType or Block.
<b>SysML Parametrics Extensions</b>	<b>Objective Function</b>	Defines a SysML Constraint Block for use as an objective function to evaluate Measures of Effectiveness (MOEs).
	<b>Measure of Effectiveness</b>	Defines a SysML property for use as a Measure of Effectiveness (MOE).

Page	Item	Action
<b>SysML Block Internal</b>	<b>Property</b>	Describes the decomposition of a SysML Block in the context of its whole using instances of reusable SysML Blocks.
	<b>Connector Property</b>	Creates an instance of an Association Block.
	<b>Distributed Property</b>	Creates a stereotype of Property, to apply a probability distribution to the values of the property.
	<b>Flow Property</b>	Creates a single kind of Flow element that can flow to or from a block.
	<b>Participant Property</b>	Creates the end of a connector owned by an Association Block.
<b>SysML Block Internal Relationships</b>	<b>Dependency</b>	Establishes a traceable relationship describing how one element is dependant upon another.
	<b>Item Flow</b>	Specifies the items that flow across a connector in an interaction point. Used in the same way as UML Information Flows. See <a href="#">Using Information Flows</a> <sup>[1411]</sup>
	<b>Connector</b>	Establishes Communication links between parts. (See also: <a href="#">Connector</a> <sup>[1402]</sup> .)
	<b>Binding Connector</b>	Establishes a connection between two Parts in a system decomposition.
<b>SysML Patterns</b>	<b>Constraint Block</b>	A pattern that creates a typical Constraint property that owns two Parts.

### 12.1.7 SysML Requirements Toolbox

When you are constructing SysML models, you can populate the Requirements diagrams using the icons on the SysML Requirements pages of the Diagram Toolbox.

[Access](#) [Diagram](#) | [Diagram Toolbox: More Tools](#) | [SysML 1.3](#) | [SysML 1.3 Requirements](#)



### SysML Requirement Objects

Page	Item	Action
<b>SysML Requirement s</b>	<b>Requirement</b>	Specifies the capabilities of the system, or the conditions that it should satisfy.
	<b>Test Case</b>	Describes the verification of a Requirement through methods of inspection, analysis, demonstration or testing.
<b>SysML Requirement Relationships</b>	<b>Containment</b>	Graphically displays ownership of one element within a parent element.
	<b>Trace</b>	Declares a trace relationship between a SysML Requirement and another SysML element.
	<b>Copy</b>	Declares a copy of one SysML Requirement by another.
	<b>Derive</b>	Derives a SysML Requirement from another.
	<b>Verify</b>	Declares a verification of a SysML Requirement by another SysML element.

Page	Item	Action
	<b>Refine</b>	Declares a refinement of a SysML Requirement by another SysML element.
	<b>Satisfy</b>	Declares that the SysML Requirement is satisfied by another SysML element.
<b>SysML Requirement Extensions</b>	<b>Extended Requirement</b>	Extends a SysML Requirement with additional Tag properties.
	<b>Functional Requirement</b>	Declares a SysML Requirement that describes the operation, or behavior, that the system must perform.
	<b>Interface Requirement</b>	Declares a SysML Requirement that describes how the system connects, or interfaces with, other systems.
	<b>Performance Requirement</b>	Declares a SysML Requirement that describes how the system performs against defined capabilities or conditions.
	<b>Physical Requirement</b>	Declares a SysML Requirement that describes the physical characteristics, or physical constraints, of the system.
	<b>Design Requirement</b>	Declares a SysML Requirement that specifies a constraint on the implementation of the system.

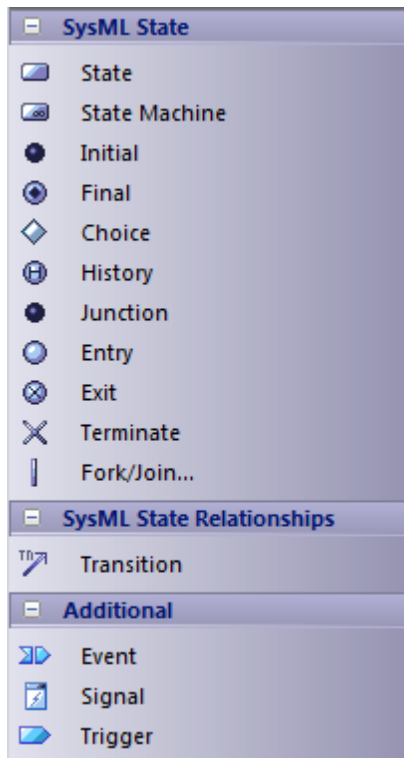
### 12.1.8 SysML State Machine Toolbox

When you are constructing SysML models, you can populate the State Machine diagrams using the icons on the SysML State Machine pages of the Diagram Toolbox.

The Block that owns the State Machine diagram is automatically represented by a diagram frame enclosing the State Machine diagram elements. You can:

- Hide the frame by right-clicking on the diagram and selecting the **Hide Diagram Frame** context menu option
- Make the frame selectable to move or resize it, by right-clicking on it and selecting the **Selectable** context menu option
- Create Ports and Parts on the frame and create connectors between them

**Access** [Diagram](#) | [Diagram Toolbox: More Tools](#) | [SysML 1.3](#) | [SysML 1.3 State Machine](#)



### SysML State Machine Objects

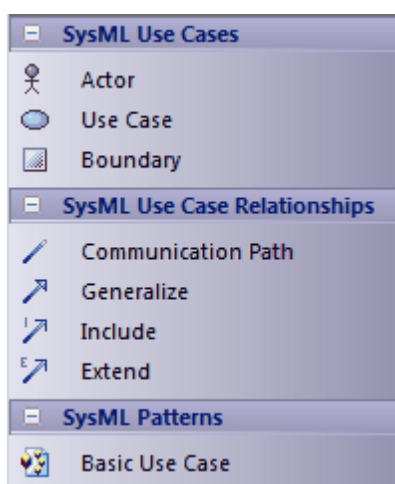
Page	Item	Action
<b>SysML State</b>	<b>State</b>	Declares a significant condition in the life of a SysML Block within its State Machine.
	<b>State Machine</b>	Describes the life-cycle behavior of a SysML Block in terms of its states and transitions.
	<b>Initial</b>	Declares the starting state of the State Machine.
	<b>Final</b>	Declares the ending state of the State Machine, and its completion.
	<b>Choice</b>	Declares a Junction with a mandatory 'else' transition.
	<b>History</b>	Represents the last active State of the State Machine prior to its interruption.
	<b>Junction</b>	Declares a decision point at which a Transition branches out into multiple guarded, alternative paths.

Page	Item	Action
	<b>Entry</b>	Declares an Entry point between State Machines, Substate Machines and Regions.
	<b>Exit</b>	Declares an Exit point between State Machines, Substate Machines and Regions.
	<b>Terminate</b>	Declares a termination State in which the State Machine no longer operates.
	<b>Fork/Join</b>	Simultaneously branches and joins a set of Transitions.
<b>SysML State Relationships</b>	<b>Transition</b>	Establishes a life-cycle path between one State and another, based on its operational conditions.
<b>Additional</b>	<b>Event</b>	Depicts the action of sending a signal.
	<b>Signal</b>	A specification of Send request instances communicated between objects.
	<b>Trigger</b>	Indicates an event that initiates an action (and might arise from completion of a previous action).

### 12.1.9 SysML Use Case Toolbox

When you are constructing SysML models, you can populate the Use Case diagrams using the icons on the SysML Use Cases pages of the Diagram Toolbox.

[Access](#) [Diagram](#) | [Diagram Toolbox: More Tools](#) | [SysML 1.3](#) | [SysML 1.3 Use Cases](#)



### SysML Use Case Objects

Page	Item	Action
<b>SysML Use Cases</b>	<b>Actor</b>	Represents a user that interacts with one or more SysML systems.
	<b>Use Case</b>	Describes the expected functionality of a system as a UML Use Case.
	<b>Boundary</b>	Graphically binds elements in a diagram within a border.
<b>SysML Use Case Relationships</b>	<b>Communication Path</b>	Declares which Actors perform in the Use Case.
	<b>Generalize</b>	Describes an element as a specialized descendant of another element, containing additional properties and behavior.
	<b>Include</b>	Describes one Use Case as a subset of another.
	<b>Extend</b>	Describes one Use Case as an extension of another.
<b>SysML Patterns</b>	<b>Basic Use Case</b>	A pattern that creates a typical simple Use Case diagram of Actor, Use Case and System Boundary elements.

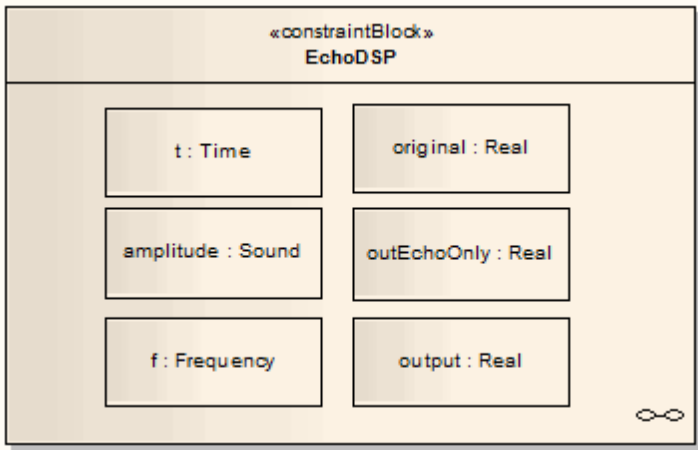
## 12.2 SysML Parametric Models

SysML Parametric models support the engineering analysis of critical system parameters, including the evaluation of key metrics such as performance, reliability and other physical characteristics.

### Topics

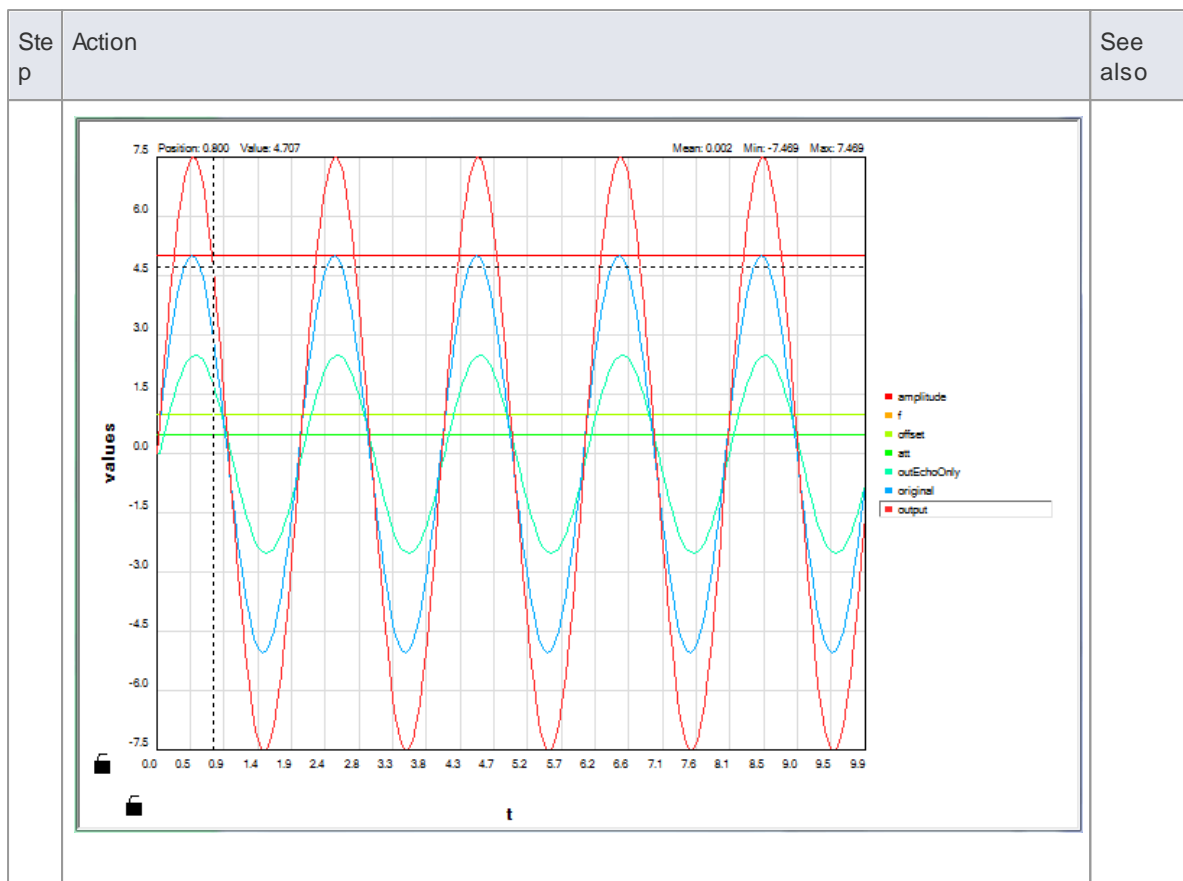
Topic	Detail	See also
<b>SysML Parametric Models - Concept</b>	<p>SysML Parametric Models unite requirements models with system design models by capturing executable constraints based on complex mathematical relationships.</p> <p>The following text is derived from the SysML entry in the online Wikipedia.</p> <p><i>The advantages of SysML over UML for systems engineering become obvious if you consider a concrete example, such as modeling an automotive system. With SysML you can use Parametric diagrams to precisely define performance and mechanical constraints such as maximum acceleration, curb weight, air conditioning capacity, and interior cabin noise management.</i></p> <p>For further information on the concepts of SysML Parametric models, refer to the official OMG SysML website and its linked sources.</p> <p>Enterprise Architect enables you to develop SysML Parametric models quickly and simply; these models can also be simulated.</p>	<p><a href="#">SysML</a> (Online Resource)</p> <p><a href="#">OMG SysML</a> (Online Resource)</p>

### Create a Parametric model

Step	Action	See also
1	<p>Create a collection of <i>SysML Constraint Blocks</i> that formally describe the function of a constraint in a simulation model.</p> <p>Each Constraint Block contains properties that describe its input and output parameters, as well as <i>Element Script</i> that describes the constraint's executable component.</p>  <p>The diagram shows a SysML Constraint Block named «constraintBlock» EchoDSP. It contains six properties arranged in two columns: t : Time, original : Real, amplitude : Sound, outEchoOnly : Real, f : Frequency, and output : Real. An infinity symbol is located in the bottom right corner of the block.</p>	



Step	Action	See also
2	<p>Right-click on each of the <i>constraintBlocks</i> and select the <b>SysML   Add Element Script</b> context menu option to add script to the constraint block.</p> <p>This is where you express the relationship / behavior of the constraint block as an executable script.</p>	
3	<p>Create a SysML Constraint Block to contain the Parametric model to simulate; the Parametric model contains properties and occurrences of constraint blocks as Constraint Property elements, connected in a Parametric Diagram.</p>	
4	<p>Right-click within a Parametric Diagram and select the <b>SysML   Simulate Diagram...</b> context menu option.</p>	
5	<p>Depending on your <b>configuration selections</b>, the simulation's results are either written to a comma-separated CSV file or graphed in a 2-dimensional plot.</p>	



### Notes

- Systems Modeling Language (SysML) Parametric Model Simulation is available in the Systems Engineering and Ultimate editions of Enterprise Architect

### Learn more

- [SysML](#)<sup>[2294]</sup>
- [Simulate a SysML Model](#)<sup>[2322]</sup>

## 12.2.1 Simulate a SysML Model

You simulate a SysML model from a Parametric diagram, using the Simulation Configuration dialog.

**Access** [Parametric Diagram Context Menu | SysML | Simulate Diagram..](#)

### Simulate a SysML model

Step	Action	See also
1	The Parameters panel lists all of the parameters that can be assigned input. Select each of the required parameters and click on the right <b>Arrow</b> button to assign them as input.	

Step	Action	See also
	<p>Parameters designated as input parameters are listed in the Inputs panel on the right.</p> <p>There must be at least one input parameter assigned for the simulation to execute.</p>	
2	<p>Assign a set of values for each of the designated input parameters.</p> <p>For each input parameter, in the Input Values panel select one of the two possible value kinds:</p> <ul style="list-style-type: none"> <li>• <b>Discrete</b> - To enter a constant or a comma-separated range of discrete values</li> <li>• <b>Range</b> - To enter a range of values beginning at the <b>From</b> value and ending at the <b>To</b> value; the input values are incremented by the <b>Step</b> value</li> </ul>	
3	<p>Specify the classes of output value:</p> <ul style="list-style-type: none"> <li>• <b>Parameters</b> - To output the parameters' data, select the checkbox</li> <li>• <b>Variables</b> - To output the data generated within each internal variable, select the checkbox; internal variables are automatically generated by the simulator</li> </ul>	
4	<p>Specify how the simulation results are to be reported.</p> <p>The Output Format panel enables you to choose how the simulation outputs the simulation data:</p> <ul style="list-style-type: none"> <li>• <b>Plot To Graph</b>: To plot the results on a 2-dimensional graph, select the checkbox; if you select this option, you must specify an input parameter for the plot's <b>X Axis</b></li> <li>• <b>Title</b> - To enter a title for the graph, type in the title text</li> <li>• <b>Output to File</b> - To output the results to a CSV text file, select the checkbox and type or browse ( ... ) for the file name</li> </ul>	
5	Click on the <b>OK</b> button to execute the simulation.	

### Notes

- Systems Modeling Language (SysML) Parametric Model Simulation is available in the Systems Engineering and Ultimate editions of Enterprise Architect

### Learn more

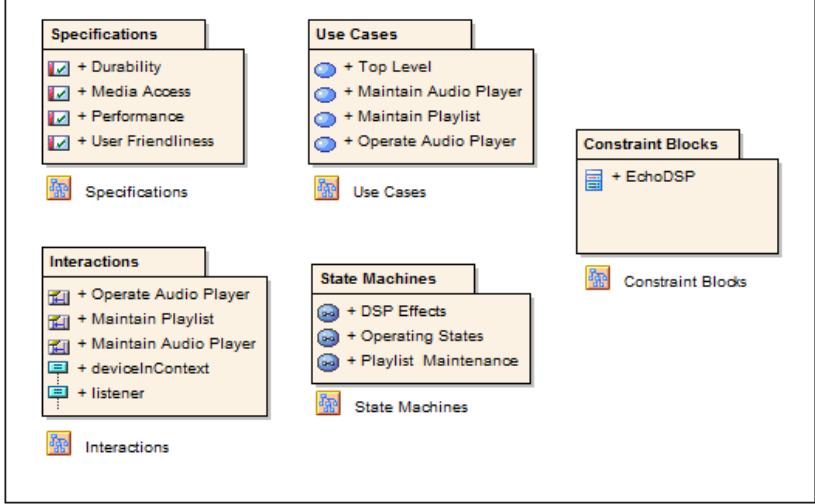
- In the **EAExample Model**, see:
  - Project Models > Systems Engineering > Requirements Model > Constraint Blocks > «constraintBlock» EchoDSP > EchoDSP

Learning Center topics

- (Alt+F1) | [Enterprise Architect](#) | [System Engineering](#) | [Parametric Models](#) | [Getting Started](#)
- (Alt+F1) | [Enterprise Architect](#) | [System Engineering](#) | [Parametric Models](#) | [Parametric Example Models](#)

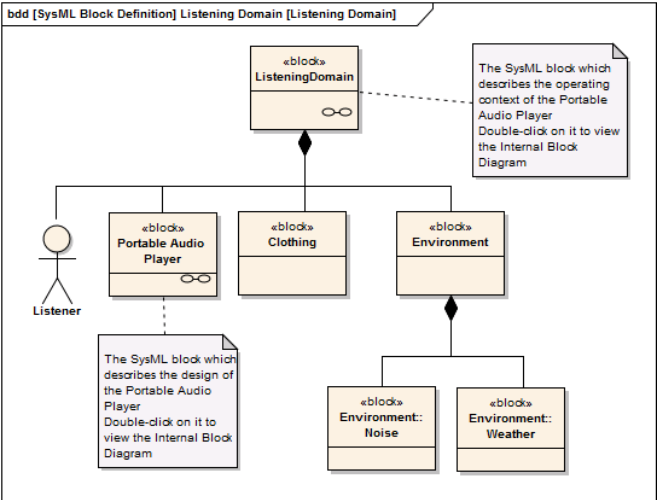
## 12.3 A SysML Requirements Model

### Topics

Topic	Detail	See also
<b>Example</b>	<p>The SysML <i>Requirements Model</i> provides the system requirements, the expected abstract behavior and the operating constraints that the designed system must conform to.</p> <p>The following diagram shows an example requirements model for a <i>Portable Audio Player</i>.</p>  <p>In the example Requirements Model, each of the child packages contains child models that capture the following aspects of the system's requirements:</p> <ul style="list-style-type: none"> <li>• The <i>Specifications</i> package contains SysML Requirements describing the overall expectations of the designed system</li> <li>• The <i>Use Cases</i> package contains SysML Use Cases that describe the general interaction between the system and its users</li> <li>• The <i>Interactions</i> package contains SysML Interactions that describe a detailed sequence of interactions between the system and its users</li> <li>• The <i>State Machines</i> package contains SysML State Machines that describe each of the operational states the designed system has</li> <li>• The <i>Constraint Blocks</i> package contains SysML ConstraintBlocks that describe the expected performance and operating boundaries of the system</li> </ul>	

## 12.4 A SysML Operational Domain Model

### Topics

Topic	Detail	See also
<b>Abstract</b>	<p>The SysML <i>Operational Domain Model</i> defines the system's operating environment, which describes the operating conditions that the system is intended to operate within.</p> <p>The following diagram shows an example Operational Domain model for a <i>Portable Audio Player</i>; the SysML Block Definition Diagram describes the Operational Domain (in this example - the <i>ListeningDomain</i>) as a system composition.</p>  <p>The diagram is a SysML Block Definition Diagram (bdd) titled "ListeningDomain [ListeningDomain]". It shows a composition of blocks. At the top is the "«block» ListeningDomain" block, which contains three sub-blocks: "«block» Portable Audio Player", "«block» Clothing", and "«block» Environment". The "«block» Portable Audio Player" block is connected to a "Listener" actor. The "«block» Environment" block contains two sub-blocks: "«block» Environment:: Noise" and "«block» Environment:: Weather". Two callout boxes provide additional information: one for the "«block» Portable Audio Player" stating "The SysML block which describes the design of the Portable Audio Player. Double-click on it to view the Internal Block Diagram", and another for the "«block» ListeningDomain" stating "The SysML block which describes the operating context of the Portable Audio Player. Double-click on it to view the Internal Block Diagram".</p> <p>In this example, the <i>ListeningDomain</i> is defined as a system containing other subsystems within it; the domain contains subsystems that define the Listener (i.e. User), the Portable Audio Player, Clothing (which the user wears), and the External Environment.</p> <p>Details of the <i>ListeningDomain</i> system are further detailed in the <i>ListeningDomain</i>'s Internal Block Diagram:</p>	

Topic	Detail	See also
	<div><div><div>ibd [SysML Internal Block] ListeningDomain [ListeningDomain]</div><div><p>The diagram shows an internal block definition for 'ListeningDomain'. It contains four components: a stick figure labeled ':Listener', a portable audio player labeled 'portableAudioPlayer', a black t-shirt labeled 'listenerClothing', and a landscape with a sun and clouds labeled 'weather'. The 'portableAudioPlayer' is connected to the ':Listener' and 'listenerClothing'. The 'weather' component is connected to the 'portableAudioPlayer' and 'listenerClothing'. The 'externalNoise' component is also connected to the 'portableAudioPlayer'.</p></div></div><p>In this example, the <i>ListeningDomain</i>'s system's detailed composition shows how the Portable Audio Player and other sub-systems fit together to form the Listening Domain; it also describes the binding relationships between the parts, which describe how the parts are functionally bound to one another.</p></div>	

## 12.5 Compose System Design

### Topics

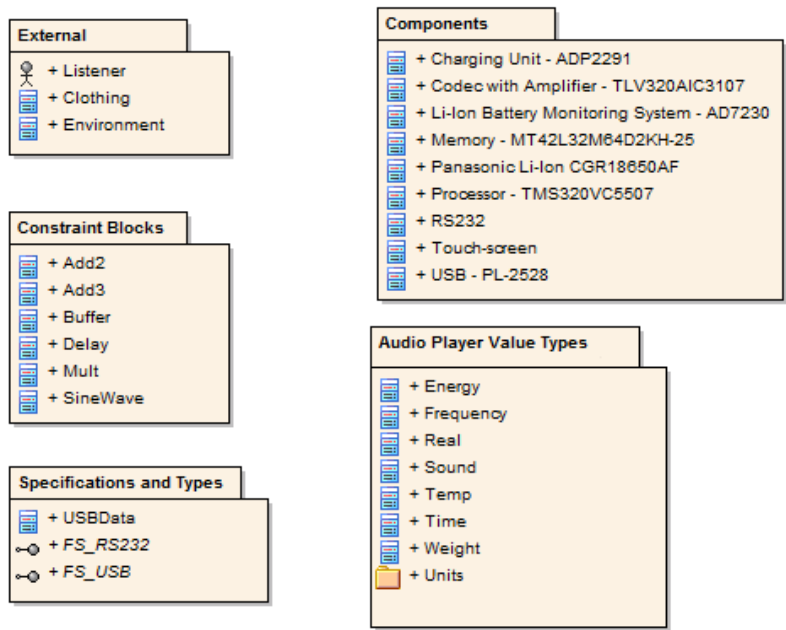
Topic	Detail	See also
<b>SysML Design Model</b>	<p>The <i>SysML Design Model</i> contains the blocks that define the system's composition; it describes the manner in which reusable subsystems fit together to fulfill the design requirements.</p> <p>The following diagram shows an example Design Model for a <i>Portable Audio Player</i>; the <i>SysML Block Definition Diagram</i> describes the <i>Portable Audio Player</i> as a composition of various reusable off-the-shelf subsystems and in-house designed ones.</p> <pre> graph TD     subgraph "bdd [SysML Block Definition] Design [Design]"         direction TB         PAA["«block» Portable Audio Player"]         PPS["«block» Portable Audio Player:: PowerSubsystem"]         PPRS["«block» Portable Audio Player:: ProcessingSubsystem"]         PUI["«block» Portable Audio Player::User Interface"]         PTS["«block» Portable Audio Player::Transport Subsystem"]                  PAA --&gt; PPS         PAA --&gt; PPRS         PAA --&gt; PUI         PAA --&gt; PTS                  PPS --&gt; BAT["«block» Panasonic Li-Ion CGR18650AF"]         PPS --&gt; CHG["«block» Charging Unit - ADP2291"]         PPS --&gt; PMON["«block» Li-Ion Battery Monitoring System - AD7230"]                  PPRS --&gt; CPU["«block» Processor - TMS320VC5507"]         PPRS --&gt; COD["«block» Codec with Amplifier - TLV320AIC3107"]         PPRS --&gt; MEM["«block» Memory - MT42L32M64D2KH-25"]                  PUI --&gt; TSC["«block» Touch-screen"]         PUI --&gt; BTN["«block» Portable Audio Player::User Interface::Buttons"]                  PTS --&gt; RST["«block» RS232"]         PTS --&gt; USB["«block» USB - PL-2528"]     end </pre> <p>In the example above, the <i>Portable Audio Player</i> is defined as a SysML system containing subsystems that perform specific tasks; the design contains subsystems for supplying power, performing playback and audio processing, interfacing with other devices and the user interface.</p> <p>Details of the <i>Portable Audio Player</i>'s composition are further described in detail within the <i>Portable Audio Player's Internal Block Diagram</i>:</p>	



© 1998-2014 Sparx Systems Pty Ltd

## 12.6 Create Reusable Subsystems

### Topics

Topic	Detail	See also
<b>SysML Design Model</b>	<p>Model Based Systems Engineering provides the flexibility and expressiveness to define complex systems quickly and effectively, by reusing common entities across design projects.</p> <p>A <i>Library</i> is a package containing many reusable subsystems, parametric constraints, common data types and common value types, dimensions and units.</p> <p>The following diagram shows an example library model:</p>  <p>The diagram illustrates a library model with five packages:</p> <ul style="list-style-type: none"> <li><b>External</b>: Contains + Listener, + Clothing, and + Environment.</li> <li><b>Components</b>: Contains + Charging Unit - ADP2291, + Codec with Amplifier - TLV320AIC3107, + Li-Ion Battery Monitoring System - AD7230, + Memory - MT42L32M64D2KH-25, + Panasonic Li-Ion CGR18650AF, + Processor - TMS320VC5507, + RS232, + Touch-screen, and + USB - PL-2528.</li> <li><b>Constraint Blocks</b>: Contains + Add2, + Add3, + Buffer, + Delay, + Mult, and + SineWave.</li> <li><b>Specifications and Types</b>: Contains + USBData, + FS_RS232, and + FS_USB.</li> <li><b>Audio Player Value Types</b>: Contains + Energy, + Frequency, + Real, + Sound, + Temp, + Time, + Weight, and + Units.</li> </ul> <p>In the example Library, each of the child packages contains child models that capture the following reusable entities:</p> <ul style="list-style-type: none"> <li><i>Blocks</i> defining systems such as those listed in the <i>Components</i> package, or those defined in the <i>External</i> package</li> <li><i>ConstraintBlocks</i> defining parametric constraints for use in parametric models.</li> <li><i>Value Types</i> describing quantities, expressed as measurable dimensions in specific units</li> <li><i>Data Types</i> and <i>Flow Specifications</i> describing data structures and <i>Flows</i></li> </ul>	

## 12.7 Migrate SysML Model to Later SysML Version

If you have created a model (or part of a model) under an older release of SysML, you can migrate it to the next release (that is, 1.1 to 1.2, or 1.2 to 1.3). using the Automation Interface function *Migrate()*. This function updates the Tagged Values and, if required, stereotypes to the later release for all elements, attributes, connectors and diagrams under the selected package or element.

### Example Script for Migrating SysML 1.1 to SysML 1.2

The following VB script calls the *Migrate()* function to migrate the SysML 1.1 package or element to SysML 1.2:

```
Sub MigrateElement (sGUID, lngPackageID)
    Dim proj as EA.Project
    set proj = Repository.GetProjectInterface
    proj.Migrate sGUID, "SysML1.1", "SysML1.2"

    'refresh the model
    If lngPackageID <> 0 Then
        Repository.RefreshModelView (lngPackageID)
    End If
End Sub

Sub MigrateSelectedItem
    Dim selType
    Dim selElement as EA.Element
    Dim selPackage as EA.Package
    selType = GetTreeSelectedItemType
    If selType = 4 Then 'means Element
        set selElement = GetTreeSelectedObject
        MigrateElement selElement.ElementGUID, selElement.PackageID
        MsgBox "Element Migration Completed", 0, "SysML Migration"
    ElseIf selType = 5 Then 'means Package
        set selPackage = GetTreeSelectedObject
        MigrateElement selPackage.PackageGUID, selPackage.PackageID
        MsgBox "Package Migration Completed", 0, "SysML Migration"
    Else
        MsgBox "Select a Package or Element in the Project Browser to
        initiate migration", 0, "SysML Migration"
    End If
End Sub

Sub Main
    MigrateSelectedItem
End Sub

Main
```

### Migrate SysML 1.1 to SysML 1.3

Migrate from SysML 1.1 to SysML 1.2, as above, then migrate from SysML 1.2 to SysML 1.3, as below.

### Migrate SysML 1.2 to SysML 1.3

The MDG Technology for SysML 1.3 has a built-in script for migrating 1.2 models to 1.3. You need to enable both the MDG Technology for SysML 1.2 and the MDG Technology for SysML 1.3. Firstly, you select the SysML 1.2 package in the Project Browser, Then you open the Scripting window and open the *SysML 1.3* script group, and execute the *Migrate* script.

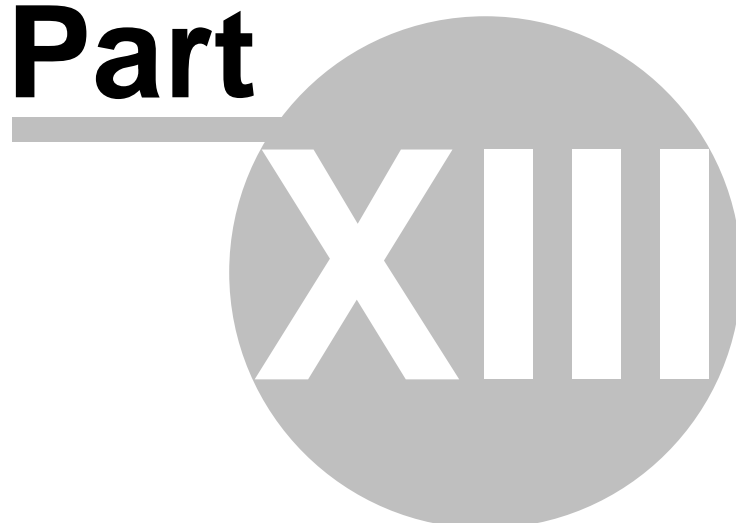
**Notes**

- When migrating from SysML1.1 to SysML1.2:
  - The stereotype *dimension* is changed to *quantitykind*
  - The stereotype *dataType* is removed from SysML1.2
  - The Tagged Value *dimension* is migrated to *quantitykind*; this applies to stereotypes *unit* and *valueType*
  - The Tagged Value *isConjugated* of stereotype *flowport* is migrated to custom properties

**Learn more**

- [Migrate\(\)](#)<sup>[2978]</sup>
- [Scripting](#)<sup>[2791]</sup>
- [Manage MDG Technologies](#)<sup>[1477]</sup>

**Part**



## 13 Database Engineering



Enterprise Architect supports Data Modeling on different levels of abstraction to develop the standard Conceptual, Logical and Physical models. It supports:

- Interconnecting models across the three levels
- Using MDA transforms to generate the Physical model from the Logical model
- The ability to generate schema scripts from the Physical model for loading to a DBMS
- Features for remodeling legacy systems, including Reverse Engineering an existing DBMS structure

This section describes database engineering, explaining:

- How to create a Physical Data Model
- How to import database schema, and
- How to generate DDL for the model

### Learn more

- [Data Models](#)<sup>[1937]</sup> (Conceptual, Logical and Physical Models)
- [Import Database Schema](#)<sup>[2376]</sup>
- [Generate DDL](#)<sup>[2380]</sup>
- [Physical Data Model](#)<sup>[2335]</sup>
- [Supported Databases](#)<sup>[2375]</sup>

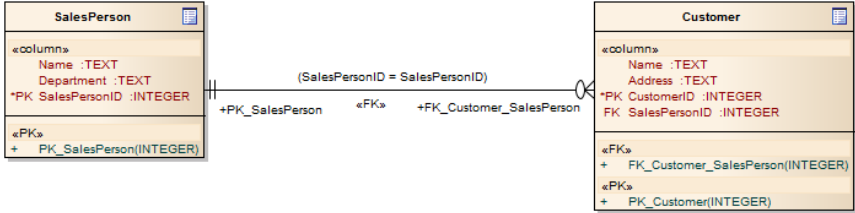
### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Database Engineering**

## 13.1 Physical Data Model

The Physical data model visually represents the structure of the data as implemented by a relational database schema.

### Topics

Topic	Detail	See also
<b>Benefit</b>	<p>In addition to providing a visual abstraction of the database structure, an important benefit of defining a Physical data model is that you can automatically derive the database schema from the model.</p> <p>This is possible due to the richness of meta-data captured by a Physical data model and its close mapping to aspects of the database schema, such as tables, columns, primary and foreign keys.</p>	<a href="#">Data Models</a> [1937]
<b>Example</b>	<p>The following is an example of a Physical data model that could be used to automatically generate a database schema.</p> <p>Each table is represented by a UML Class; columns, including Primary and Foreign Keys, are modeled using UML attributes and operations.</p>  <p>A Physical data model defined using a UML Profile and the Information Engineering notation</p>	
<b>Notation</b>	<p>The above model is defined using Enterprise Architect's <i>UML Profile for Data Modeling</i>; the relationship between the tables uses the Information Engineering notation.</p> <p>Information Engineering is one of three notations that Enterprise Architect supports to help Data Modelers distinguish cardinality in relationships.</p>	<a href="#">Data Modeling Notations</a> [2384] <a href="#">Connectors Tab</a> [831]
<b>Set Default DBMS</b>	<p>Prior to creating a Physical data model it is advisable to set the default DBMS. This pre-sets the default database when you create any new Table elements.</p> <p>If the DBMS is not set, Tables are created without a DBMS type and field typing cannot be allocated when creating new fields in the table.</p> <p>You can set the default DBMS type using either:</p> <ul style="list-style-type: none"> <li>The <b>Settings   Database Datatypes</b> menu option or</li> <li>The Code Generation Toolbar</li> </ul>	<a href="#">Add New Datatypes</a> [2350] <a href="#">Set the Database Type</a> [2347] <a href="#">Code Generation Toolbar</a> [144]

Learn more

- [Create a Data Model Diagram](#) <sup>[2336]</sup>
- [Tables and Columns](#) <sup>[2338]</sup>
- [Data Types](#) <sup>[2348]</sup>
- [Primary and Foreign Keys](#) <sup>[2354]</sup>
- [Stored Procedures](#) <sup>[2364]</sup>
- [Indexes](#) <sup>[2365]</sup>
- [Triggers](#) <sup>[2368]</sup>
- [Check Constraints](#) <sup>[2370]</sup>
- [Views](#) <sup>[2371]</sup>
- [Supported Databases](#) <sup>[2375]</sup>
- [Default Constraints](#) <sup>[2371]</sup>
- [Oracle Packages](#) <sup>[2374]</sup>
- [Forward and Reverse Engineer Databases](#) <sup>[2334]</sup>
- [Transform Abstract Models to Physical Data Models Using MDA](#) <sup>[2025]</sup>
- [Connect to Element Feature](#) <sup>[1110]</sup>

Learning Center topics

- (Alt+F1) | **Enterprise Architect | Database Engineering**

**13.1.1 Create a Data Model Diagram**

Data Modeling diagrams are extended Class diagrams, and are used for modeling the structure of a relational database. These diagrams can include Tables, Views and Procedures, and can be set to use the connector notations *UML DDL Profile*, *IDEF1X* or *Information Engineering*.

**Access** **Project Browser package context menu | Add Diagram**

Create a Data Modeling diagram

Field/Button	Action	See also
<b>Name</b>	This field defaults to the package name. If required, overwrite the default name with your preferred name.	
<b>Select From</b>	Click on <b>Extended</b> .	<a href="#">Add New Diagrams</a> <sup>[822]</sup>
<b>Diagram Types</b>	Click on <b>Data Modeling</b> .	
<b>OK</b>	Click on this button. The Diagram View displays the blank diagram, and the Data Modeling pages display in the Diagram Toolbox.	<a href="#">Data Modeling Toolbox</a> <sup>[818]</sup>



Field/Button	Action	See also
	Drag elements and connectors from the Toolbox onto your diagram, to create your data model.	

### Notes

- The new diagram connector notation defaults to **Information Engineering**; If you want to change this, display the Connectors tab of the diagram Properties dialog (**Diagram | Properties > Connectors**) and select the required option from the **Connector Notation** drop-down list

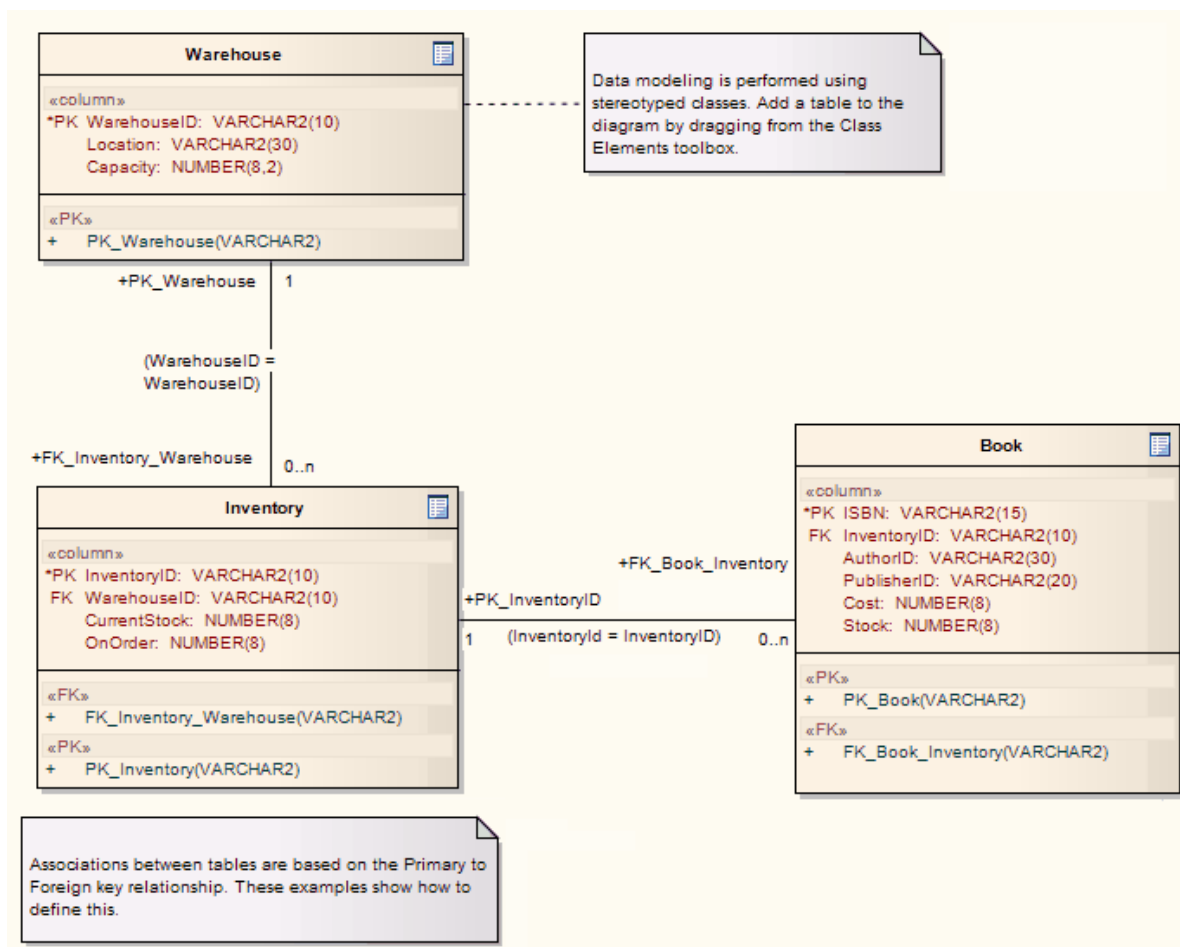
### Learn more

- [Data Modeling Notations](#) <sup>[2384]</sup>
- [Connectors Tab](#) <sup>[831]</sup>
- [Example Data Model Diagram](#) <sup>[2337]</sup>

#### 13.1.1.1 Example Data Model Diagram

An example of a *Data Model* diagram is provided below, showing three tables that are linked on *primary to foreign* key pairs with associated multiplicity.

### Example



### 13.1.2 Tables and Columns

Tables and columns form the basis of Physical data models.

#### Topics

Topic	Detail	See also
<b>Rows and Columns</b>	<p>The basic structural element of a relational database is the <i>table</i>, which represents a set of records, or rows, that have the same structure.</p> <p>Each row contains one or more <i>columns</i>.</p> <p>Every individual item of data entered into a relational database is represented by a value in a column of a row in a table.</p> <p>Enterprise Architect's <i>UML Profile for Data Modeling</i> represents:</p> <ul style="list-style-type: none"> <li>Database tables as Classes with a stereotype of &lt;&lt;table&gt;&gt;</li> <li>Columns by Attributes with a stereotype of &lt;&lt;column&gt;&gt;</li> </ul>	<a href="#">Create a Table</a> <sup>2339</sup> <a href="#">Create Columns</a> <sup>2345</sup>
<b>Example</b>	A simple example of a Physical data model in Enterprise Architect is shown below. The example consists of two tables represented by UML Classes, named <i>SalesPerson</i> and <i>Customer</i> . The table	<a href="#">Add New Datatypes</a> <sup>2350</sup>

Topic	Detail	See also
	<p>stereotype is denoted by the icon in the top-right corner of each Class.</p> <p>The tables each define three columns using UML Attributes typed appropriately for the target DBMS, which happens to be MySQL.</p> <div data-bbox="432 517 1114 696"> </div> <p>A simple Data model consisting of two tables, represented by UML Classes</p>	
<b>Default DBMS</b>	<p>Prior to creating a tables in a Physical data model it is advisable to set the default DBMS. This pre-sets the default database when you create any new Table elements.</p> <p>If the DBMS is not set, Tables are created without a DBMS type and field typing cannot be allocated when creating new fields in the table.</p> <p>You can set the default DBMS type using either:</p> <ul style="list-style-type: none"> <li>The <b>Settings   Database Datatypes</b> menu option or</li> <li>The Code Generation Toolbar</li> </ul>	<a href="#">Add New Datatypes</a> <sup>[2350]</sup>
<b>Attribute/Field Mapping</b>	To help you map Class Attributes to Table fields, you can create connectors between specific attributes (features) in the Class element and the column Attributes in the Table element.	<a href="#">Connect to Element Feature</a> <sup>[1110]</sup>
<b>Model Driven Architecture</b>	Enterprise Architect can generate simple DDL scripts to create the tables in your model. You can also perform Model Driven Architecture (MDA) Transformations to DDL - Enterprise Architect provides a template specifically for DDL transformations.	<a href="#">Model Transformation</a> <sup>[2013]</sup> <a href="#">DDL Transformation</a> <sup>[2025]</sup>
<b>Primary and Foreign Keys</b>	Primary Key and Foreign Key information modeling is considered in later topics.	<a href="#">Primary and Foreign Keys</a> <sup>[2354]</sup>

### 13.1.2.1 Create a Table

Fundamental to data modeling is the creation of Tables within the model.

#### Add a Table to your data model

Step	Action	See also
1	Create and/or open a Data Modeling diagram.	<a href="#">Create a Data Model Diagram</a> <sup>[2336]</sup>
2	Click on the <i>Table</i> element in the Diagram Toolbox, then click on the diagram. The Table element is displayed on the diagram.	<a href="#">Data Modeling Toolbox</a> <sup>[818]</sup>
3	Use the Table Properties dialog to set the table name and other properties as required.	<a href="#">Working with Table Properties</a> <sup>[2340]</sup>
4	Use the Quicklinker to add related Table elements.	<a href="#">The Quick Linker</a> <sup>[896]</sup>

#### Learn more

- [Working with Table Properties](#) <sup>[2340]</sup>

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Database Engineering | Physical Data Model | Create Table**

#### **13.1.2.1.1 Working with Table Properties**

Once you have created a table, you must set its properties to define database-specific information that can be used to generate the schema in DDL. Table properties are set using the Properties dialog.

It is recommended that you set the DBMS you are using as a *default*, to automatically apply that DBMS definition to all tables you create. However, if you do not set a default, or want to change the value for specific tables for any reason, you can also identify the DBMS on the Table Properties dialog for each Table.

For MySQL and Oracle, some additional properties are defined using Tagged Values, such as the Table Owner and DBMS-specific options. For other DBMSs, additional table properties are not yet supported.

#### Topics

Topic	Link
<p><b>Set the Database type for a table</b> - other than the table name, the most important property to set for a table is the database type, which defines:</p> <ul style="list-style-type: none"> <li>• The list of data types that are available for defining columns, and</li> <li>• Which dialect of DDL is generated</li> </ul> <p>You can either set the DBMS as a default for all tables, or within the Table Properties for each individual Table.</p>	<p><a href="#">Add New Datatypes</a> <sup>[2350]</sup></p> <p><a href="#">Set the Database Type</a> <sup>[2341]</sup></p>

Topic	Link
<b>Set the Table Owner</b> - Some relational databases enable you to assign a table to one of several owners within the database; Enterprise Architect enables you to model this using the <i>Owner</i> Tagged Value, so that in the generated DDL script the table name is prefixed by the owner name.	<a href="#">Set Table Owner/Schema</a> <sup>[2342]</sup>
<b>Example MySQL options</b> - To make use of foreign keys in MySQL, you must declare the table type as InnoDB.	<a href="#">Set MySQL Options</a> <sup>[2342]</sup>
<b>Example Oracle options</b> - To set additional Oracle table properties, use the table's Tagged Values.	<a href="#">Set Oracle Table Properties</a> <sup>[2343]</sup>

Once the table properties are defined, you are ready to add columns.

#### Learn more

- [Create Columns](#) <sup>[2345]</sup>

#### 13.1.2.1.1.1 *Set the Database Type*

The most important property to set for a table (after its name) is the database type. This defines the list of data types that are available for defining columns, and also declares which dialect of DDL is generated.

If this is not set, tables are created without a DBMS type and field typing cannot be allocated when creating new fields in the table.

You can set the database type to default to the required DBMS. However, if you have not set a default or want to change the value for specific tables, you can set the DBMS for each table individually.

**Access** **Table element | Element | Properties | General**

#### Reference

Field/Button	Action	See also
<b>Database</b>	Click on the drop-down button and select the required database type from the list.	
<b>Apply</b>	Save your changes.	
<b>OK</b>		

#### Learn more

- [Add New Datatypes](#) <sup>[2350]</sup> (to set *default* DBMS)

- [Code Generation Toolbar](#)<sup>[144]</sup>

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Database Engineering | Physical Data Model | DBMS Datatypes**

#### **13.1.2.1.1.2 Set Table Owner/Schema**

Some relational databases enable you to assign a table to one of several schemas (or *owners*) within the same database. Enterprise Architect enables you to model this information using the *Owner* Tagged Value. By setting this optional value, you can tailor the generated DDL script, causing the table name to be prefixed by its assigned owner name.

#### How to

To set the table owner Tagged Value

Step	Action	See also
1	Add a Tagged Value named <i>Owner</i> , to the Table element.	<a href="#">How to Add a Tagged Value</a> <sup>[1137]</sup>
2	Type the owner name for this table in the <b>Value</b> field of the Tagged Value.  Changes are automatically saved.	

#### Notes

- For a PostgreSQL database, you should instead create a Tagged Value named OWNER TO with the corresponding value

#### **13.1.2.1.1.3 Set MySQL Options**

To make use of foreign keys in MySQL, you must declare the table type as InnoDB.

#### How to

To declare the table type as InnoDB

Step	Action	See also
1	Add a Tagged Value named <b>Type</b> to the table.	<a href="#">How to Add a Tagged Value</a> <sup>[1137]</sup>
2	Set the <b>Value</b> field to <b>InnoDB</b> .	

When you generate DDL for this table, the table type is included in the SQL script.

To allow for later versions of MySQL, additional table options that can be added in the same manner include:

Tag	Value (Example)
ENGINE	InnoDB
CHARACTER SET	latin1
CHARSET	latin1
COLLATE	latin1_german2_ci

#### 13.1.2.1.1.4 Set Oracle Table Properties

To set additional Oracle table properties, you use the table's Tagged Values

##### How to

Step	Action	See also
1	Add one or more Tagged Values to the table, using the names provided in the <b>Property/Tag</b> column of the reference table below.	<a href="#">How to Add a Tagged Value</a> <sup>[1137]</sup>
2	Specify the appropriate value for each tag. Examples are provided in the <b>Value</b> column of the reference table below.	

The same properties can be added to indexes and constraints, by highlighting the index or constraint Operation and adding the appropriate Tagged Values.

##### Reference

Property/Tag	Value
<b>BUFFER_POOL</b>	DEFAULT
<b>CACHE</b>	NOCACHE
<b>DBVERSION</b>	9.0.111
<b>FREELISTS</b>	1

Property/Tag	Value
<b>GRANT OWNER1</b>	SELECT
<b>GRANT OWNER2</b>	DELETE, INSERT, SELECT, UPDATE
<b>INITIAL</b>	65536
<b>INITRANS</b>	1
<b>LOGGING</b>	LOGGING
<b>MAXEXTENTS</b>	2147483645
<b>MAXTRANS</b>	255
<b>MINEXTENTS</b>	1
<b>MONITORING</b>	MONITORING
<b>OWNER</b>	OWNER1
<b>PARALLEL</b>	NOPARALLEL
<b>PCTFREE</b>	10
<b>PCTINCREASE</b>	0
<b>PCTUSED</b>	0
<b>SYNONYMS</b>	PUBLIC:TABLE_PUB;OWNER2:TABLE_OWNER2
<b>TABLESPACE</b>	MY_TABLESPACE
<b>TEMPORARY</b>	YES



### 13.1.2.2 Create Columns


Columns are represented in the UML Data Modeling Profile as a stereotyped attribute; that is, an attribute with the *Column* stereotype.

#### Create columns in a table

Step	Action	See also															
1	Right-click on the required Table in a diagram, and select the <b>Attributes</b> context menu option.  The Attributes dialog displays, showing the columns of the selected Table.																
2	In the <b>Name</b> field, type the column name.																
3	In the <b>Data Type</b> field, click on the drop-down arrow and select the data type (ensure the Table's target database is set first; this populates the <b>Data Type</b> drop-down list).  Click on the <b>Save</b> button.	<a href="#">Set the Database Type</a> <sup>[2341]</sup>  <a href="#">Oracle Data Types</a> <sup>[2352]</sup>															
4	Complete the required optional fields, listed in the reference table below.																
5	Click on the <b>Column Properties</b> button.  The Database Columns Properties dialog displays.  Supported column properties are identified below; it is assumed that you understand which data types these properties apply to:																
<table> <tr> <th>Property</th><th>DBMS</th><th>Notes</th></tr> <tr> <td><b>Autonum</b></td><td>Oracle MySQL SQL Server DB2 PostgreSQL Sybase ASA Sybase ASE</td><td>If you require a sequence, select the Autonum value to <b>True</b> and, if necessary, define the start increment</td></tr> <tr> <td><b>Zerofill</b></td><td>MySQL</td><td></td></tr> <tr> <td><b>Unsigned</b></td><td>MySQL</td><td></td></tr> <tr> <td><b>LengthType</b></td><td>Oracle</td><td>Character semantics</td></tr> </table>			Property	DBMS	Notes	<b>Autonum</b>	Oracle MySQL SQL Server DB2 PostgreSQL Sybase ASA Sybase ASE	If you require a sequence, select the Autonum value to <b>True</b> and, if necessary, define the start increment	<b>Zerofill</b>	MySQL		<b>Unsigned</b>	MySQL		<b>LengthType</b>	Oracle	Character semantics
Property	DBMS	Notes															
<b>Autonum</b>	Oracle MySQL SQL Server DB2 PostgreSQL Sybase ASA Sybase ASE	If you require a sequence, select the Autonum value to <b>True</b> and, if necessary, define the start increment															
<b>Zerofill</b>	MySQL																
<b>Unsigned</b>	MySQL																
<b>LengthType</b>	Oracle	Character semantics															

Step	Action	See also
6	Click on the <b>OK</b> button to return to the Attributes dialog, and then click on the <b>Save</b> button.	
7	Click on the: <ul style="list-style-type: none"> <li>• <b>New</b> button to define another column or</li> <li>• <b>Close</b> button to exit from the dialog</li> </ul>	

### Options

Field/Option/ Button	Description	See also
<b>Primary Key</b>	Indicates whether the column is the primary key for this table.	<a href="#">Primary Key</a> 
<b>Not Null</b>	Indicates whether empty values are forbidden for this column.	
<b>Participates in Unique</b>	Indicates whether the column participates in any Unique constraints.  A Unique constraint can be applied to either a single column or multiple columns. When a Unique constraint is applied to: <ul style="list-style-type: none"> <li>• A single column, each row in the table must have a unique value</li> <li>• Multiple columns, the combination of all columns involved must be unique in each record</li> </ul>	
<b>Initial</b>	A value that can be used as a default value for this column, if required.	
<b>Access</b>	Determines the column's scope ( <b>Private</b> , <b>Protected</b> or <b>Public</b> ). The default value is <b>Public</b> .	
<b>Alias</b>	An alternative name for the field for display and documentation purposes.	
<b>Notes</b>	Additional information necessary to document the column. You can format the text using the Notes toolbar at the top of the field.	

### Notes

- For MySQL, before creating columns you must first add ENUM and SET datatypes

#### Learn more

- [Reorder Columns](#) <sup>[2347]</sup>
- [Data Types](#) <sup>[2348]</sup>
- [MySQL Data Types](#) <sup>[2352]</sup>
- [Options - Code Editors](#) <sup>[2250]</sup>



#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Database Engineering | Physical Data Model | Add Column**

### 13.1.2.2.1 Reorder Columns

You can change the order in which columns are listed in a table, to define the order in which columns appear in the generated DDL that defines your database schema.

#### Change the column order

Step	Action	See also
1	In the Columns dialog, highlight a column name in the Columns panel.	
2	Click on the: <ul style="list-style-type: none"> <li> button to move the column up one position</li> <li> button to move the column down one position</li> </ul>	

### 13.1.3 Manage DBMS Settings

Using the Manage DBMS Options dialog, you can change the owner of a selected table or all of the tables in a package. With the same dialog, you can add an index on the columns participating in a Foreign Key, provided that those columns are not already indexed.

Access **Project Browser Package context menu | Code Engineering | Reset DBMS Options**

#### Change Table Owner

Field/Button	Action	See also
<b>Change Table Owner</b>	Select this checkbox.	
<b>Current Owner</b>	Click on the drop-down arrow and select either:	

Field/Button	Action	See also
	<ul style="list-style-type: none"> <li>The current owner to change, or</li> <li><b>&lt;All&gt;</b> to change the ownership of all tables in the package to the name you type in the <b>New Owner</b> field</li> </ul>	
<b>New Owner</b>	Type the name of the new table owner.	
<b>Process Child Packages</b>	If there are child packages that also require changing, select the checkbox.	
<b>OK</b>	Click on this button to change the ownership of the selected Table, or all Tables in the selected packages that have the specified current owner.	

#### Create Indexes for Foreign Keys

Field/Button	Action	See also
<b>Create Indexes on Foreign Keys</b>	Select this checkbox to create an index on all Foreign Keys (if one does not already exist).	
<b>Process Child Packages</b>	If there are child packages that also require changing, select the checkbox.	
<b>OK</b>	Click on this button to create the Foreign Key indexes.	

#### Notes

- Existing data models may not contain indexes on Foreign Key columns

#### Learn more

- [Define a Foreign Key Name Template](#)  <sup>[2362]</sup>
- [Define a Foreign Key Index Template](#)  <sup>[2363]</sup>

### 13.1.4 Data Types

Each column that you define in your data model has a data type that is appropriate to the information being stored by that column.

Enterprise Architect contains definitions of the core datatypes for a number of standard DBMS products. However, data types vary from one DBMS product to another, and they also vary between versions of a given database product. Therefore Enterprise Architect provides you with tools to:

- Define new data types for a new version of your DBMS product
- Define data types for a new, non-standard database product
- Automatically convert data types from one defined DBMS product to another
- Define how the automatic conversion should proceed from or to a non-standard DBMS product.

#### Learn more

- [Supported Databases](#) <sup>[2375]</sup>
- [DBMS Product Conversion for a Package](#) <sup>[2349]</sup>
- [Data Type Conversion For a Table](#) <sup>[2350]</sup>
- [Add New Datatypes](#) <sup>[2350]</sup>
- [Map Data Types Between DBMS Products](#) <sup>[2353]</sup>

#### **13.1.4.1 DBMS Product Conversion for a Package**

Using the DBMS Package mapper, you can automatically convert a package of database tables from one supported DBMS type to another supported DBMS type. You can also change the DBMS type for individual tables.

If one of the DBMS types is non-standard or otherwise not supported by Enterprise Architect, you should ensure that the mapping of datatypes from one DBMS type to the other has been defined.

**Access** [Project Browser Package context menu](#) | [Code Engineering](#) | [Reset DBMS Options](#)

#### Map the DBMS data types of a package to the data types of another DBMS

Field/Button	Action	See also
<b>Convert DBMS Type</b>	Select the checkbox.	
<b>Current DBMS</b>	Click on the drop-down arrow and select the current DBMS.	
<b>New DBMS</b>	Click on the drop-down arrow and select the target DBMS.	
<b>Process Child Packages</b>	If there are child packages that also require changing, select the checkbox.	
<b>OK</b>	Click on this button to map all tables in the selected packages to the new DBMS.	

#### Learn more

- [Data Type Conversion For a Table](#) <sup>[2350]</sup>
- [Map Data Types Between DBMS Products](#) <sup>[2353]</sup>
- [Manage DBMS Settings](#) <sup>[2347]</sup>

### 13.1.4.2 Data Type Conversion For a Table

Once a database schema has been set up on an Enterprise Architect diagram (either by importing through ODBC or manually setting up the tables), the DBMS can be changed to another type and the column datatypes are mapped accordingly.

You might follow this procedure if you have copied a small number of tables into the project from elsewhere, but if you have many tables you can also convert all of them at once within their parent package.

If one of the DBMS types is non-standard or otherwise not supported by Enterprise Architect, you should ensure that the mapping of datatypes from one DBMS type to the other has been defined.

#### Map the DBMS type of a table to another DBMS type

Step	Action	See also
1	Double-click on the Table element in a diagram.  The Table Properties dialog displays, with the <b>Database</b> field showing the current DBMS for this table.	
2	To map the datatypes to another DBMS, click on the <b>Database</b> drop-down arrow and select the target DBMS.	
3	Click on the <b>Apply</b> button.  The datatypes are converted to match those of the new DBMS, and these are reflected in any DDL generated from this table.	

#### Learn more

- [DBMS Product Conversion for a Package](#)  <sup>2349</sup>
- [Map Data Types Between DBMS Products](#)  <sup>2353</sup>

### 13.1.4.3 Add New Datatypes

Using Enterprise Architect's Database Datatypes dialog, you can add to the set of data types that are available for a particular DBMS.

#### Access **Settings | Database Datatypes**

#### Use to

- Identify the DBMS in use and, if required, set this as the model default
- Include any new datatypes that are supported by later versions of the DBMS and not yet included with Enterprise Architect
- Remove any datatypes that are no longer relevant
- Add a new DBMS product and its built-in data types if, for example, you want to create a physical data model for a DBMS product that is not yet supported natively by Enterprise Architect

- Change the DBMS in use and begin to map the datatypes to the new DBMS

### Reference

Use the Database Datatypes dialog to add a new datatype to a DBMS product, as described below:

Field/Button	Action	See also
<b>Product Name</b>	Click on the drop-down arrow and select an existing DBMS.	
<b>Add Product</b>	If your DBMS is not listed, click on this button to add it. An Input prompt displays, in which you type the DBMS name; click on the <b>OK</b> button to add the name to the drop-down list.	
<b>Set as Default</b>	Select the checkbox to set the selected DBMS as the default for your database engineering and modeling.  Once you set the default database, when you create any new Table element the Database type is automatically pre-set to this default.  You can also set the default database type through the Code Generation Toolbar.	<a href="#">Physical Data Model</a> <sup>[2335]</sup>  <a href="#">Code Generation Toolbar</a> <sup>[144]</sup>
<b>Datatype</b>	Type a name for the data type.	
<b>Size</b>	Select the appropriate radio button and, if appropriate, specify the default and maximum values.	
<b>Common Type</b>	Click on the drop-down arrow and select a common type to which your data type applies.	
<b>Save</b>	Click on the button to create your data type and add it to the <b>Defined Datatypes for Databases</b> list.	
<b>Delete</b>	Select a data type in the <b>Defined Datatypes for Databases</b> list and click on this button to remove the data type.	
<b>Datatype Map</b>	If you have changed the DBMS or technology for which you have defined the data types from or to an unsupported DBMS type, click on this button to define how to automatically remap the data types to your new DBMS or technology.	<a href="#">Map Data Types Between DBMS Products</a> <sup>[2353]</sup>

You can transport these data types between models as Enterprise Architect Reference Data.

Learn more

- [Add MySQL Data Types](#) <sup>[2352]</sup>
- [Customize Oracle Data Types](#) <sup>[2352]</sup>
- [Export Data Types as Reference Data](#) <sup>[376]</sup>
- [Import Data Types as Reference Data](#) <sup>[380]</sup>

Learning Center topics

- (Alt+F1) | **Enterprise Architect | Database Engineering | Physical Data Model | Data Type Map**

**13.1.4.3.1 MySQL Data Types**

MySQL supports the **ENUM** and **SET** data types, which must be added to your Enterprise Architect model before you can use them as the types for columns.

Access   **Settings | Database Datatypes**

How to

To add the ENUM and SET data types for MySQL

Step	Action	See also
<b>1</b>	Select the <b>Database Datatypes</b> menu option. The Database Datatypes dialog displays.	
<b>2</b>	In the <b>Product Name</b> field select <b>MySQL</b> .	
<b>3</b>	Add the data types <b>ENUM</b> and <b>SET</b> .	

When using these data types later in a Column's **Initial** field, type the values as a comma-separated list, in the format:

*('one','two','three')*

If one value is the default, use the format:

*('one','two','three') default 'three'*

**13.1.4.3.2 Oracle Data Types**

The Oracle data types **NUMBER** and **VARCHAR** have additional properties that you can model.

Reference



Data Type	Detail	See also
<b>NUMBER</b>	<p>The <b>NUMBER</b> data type requires precision and scale properties.</p> <p>The <b>Precision</b> and <b>Scale</b> fields are displayed on the Attributes dialog when the data type is set to <b>NUMBER</b>; if you enter information into these fields, it is displayed on your diagrams.</p> <p>For example:</p> <ul style="list-style-type: none"> <li>create NUMBER by setting Precision = <b>0</b> and Scale = <b>0</b></li> <li>create NUMBER(8) by setting Precision = <b>8</b> and Scale = <b>0</b></li> <li>create NUMBER(8,2) by setting Precision = <b>8</b> and Scale = <b>2</b></li> </ul>	
<b>VARCHAR</b>	<p>Oracle <b>VARCHAR2(15 CHAR)</b> and <b>VARCHAR2(50 BYTE)</b> data types can be created by adding the Tagged Value <b>LengthType</b> with the value <b>CHAR</b> or <b>BYTE</b>.</p>	

#### Learn more

- [How to Add a Tagged Value](#) 

### 13.1.4.4 Map Data Types Between DBMS Products

One of the advantages of using visual physical data models to hide implementation details is that you can more easily change the target technology or platform when required.

After reverse engineering your database into a physical data model, you must remap the data types before generating the schema for the new DBMS product. Enterprise Architect provides a set of default mappings for standard, supported DBMS products, to help you automate the conversion process.

However, you might want to customize the default mappings to suit your specific project requirements, or where the mapping of one data type to another is not currently defined. For example, in your migration from one DBMS platform to another, one of the platforms might be non-standard or otherwise not supported by Enterprise Architect.

**Access**    **Settings | Database Datatypes: Datatype Map**

#### Use to

- Streamline conversion of one implementation-specific database product to another
- Map data types from one DBMS product to another
- Maximize portability of your database designs
- Customize default mappings

#### Reference

On the Database Datatypes Mapping dialog, map the data types between databases as indicated below:

Field/Button	Action	See also
<b>From Product Name</b>	Click on the drop-down arrow and select the DBMS product to map data types from.	

Field/Button	Action	See also
<b>Defined Datatypes for Databases</b>	Displays all the defined data types for the product and, where appropriate, their sizes and values.  Click on the data type to map - this must have a defined size unit and value.  The <b>Datatype</b> and <b>Common Type</b> fields under the <b>From Product Name</b> field display this data type.	
<b>To Product Name</b>	Click on the drop-down arrow and select the DBMS product to map data types to.  The <b>Datatype</b> and <b>Common Type</b> fields under this field display the values corresponding to those in the fields for the 'From' product.	
<b>Size</b>	Click on the radio button for the appropriate size unit and type the default values in the corresponding data fields.	
<b>Save</b>	Click on this button to save the mapping.	

Repeat this process for all the data types to map.

Once you are satisfied with the data type mappings, you can convert either individual tables or an entire package of tables to the new target DBMS product.

#### Learn more

- [DBMS Product Conversion for a Package](#)<sup>[2349]</sup>
- [Data Type Conversion For a Table](#)<sup>[2350]</sup>

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Database Engineering | Physical Data Model | DBMS Migration**

### 13.1.5 Database Keys

Two types of key are used to access tables in a relational database: *Primary Keys* and *Foreign Keys*.

- A Primary Key uniquely identifies a record in a table
- A Foreign Key accesses data in some other related table via its Primary Key

#### Topics

Topic	Detail	See also
<b>General Usage</b>	In Enterprise Architect, Primary Keys and Foreign Keys are represented by stereotyped UML attributes and operations.	

Topic	Detail	See also
<b>Example</b>	<p>The following example shows a data model that includes both Primary Key and Foreign Key columns.</p> <p>A Physical data model defined using a UML Profile that includes Primary and Foreign Keys</p>	
<b>Primary Key</b>	In each table, one of the columns is designated as the Primary Key, indicated by the <i>PK</i> label beside the column, and a corresponding Primary Key Operation.	<a href="#">Primary Key</a> <sup>[2355]</sup>
<b>Foreign Keys</b>	Foreign Keys are shown denoted by the <i>FK</i> labels.	<a href="#">Foreign Key</a> <sup>[2355]</sup>
<b>Association Notation</b>	<p>The Association that defines a Foreign Key relationship between <i>SalesPerson</i> and <i>Customer</i> uses the <b>Information Engineering</b> notation; this is one of three notations that Enterprise Architect supports to help Data Modelers distinguish cardinality in relationships.</p> <p>Enterprise Architect also supports:</p> <ul style="list-style-type: none"> <li>• IDEF1X notation</li> <li>• UML notation</li> </ul>	<a href="#">Data Modeling Notations</a> <sup>[2354]</sup>

#### Learn more

- [Primary Keys](#)<sup>[2355]</sup>
- [Foreign Keys](#)<sup>[2355]</sup>

### 13.1.5.1 Primary Key

A Primary Key uniquely identifies a record in a table and can consist of one or more columns. In Enterprise Architect, a:

- **Simple** Primary Key (single column) is defined as the **attribute** of a stereotyped operation
- **Complex** Primary Key (several columns) is defined as the **stereotyped operation** itself; for example, a column *LastName* might not be unique within a table, so a Primary Key is created from the *LastName*, *FirstName* and *DateOfBirth* columns
- **SQL Server** Primary Key **Fill Factor** is defined as the **Tagged Value FILLFACTOR** with a numerical value, within the Primary Key operation

Learn more

- [Create a Primary Key](#)<sup>[2356]</sup>
- [Indexes](#)<sup>[2365]</sup>
- [Define a Primary Key Name Template](#)<sup>[2357]</sup>
- [Foreign Keys](#)<sup>[2358]</sup>

Learning Center topics

- (Alt+F1) | [Enterprise Architect](#) | [Database Engineering](#) | [Physical Data Model](#) | [Add Primary Key](#)

**13.1.5.1.1 Create a Primary Key**How to

To define a simple Primary Key consisting of a single column

Step	Action	See also
1	Right-click on the table in a diagram and select the <b>Attributes</b> context menu option.  The Attributes dialog displays.	
2	Select the column that makes up the Primary Key.	
3	Select the <b>Primary Key</b> checkbox and click on the <b>Save</b> button.  A stereotyped operation is automatically created; it is this operation that defines the Primary Key for the table.  To remove a Primary Key, simply delete this operation.	

To define a complex Primary Key, consisting of more than one column

Step	Action	See also
1	Follow the steps above to create a Simple Primary Key; it doesn't matter which column you choose.	
2	Right-click on the table in a diagram and select the <b>Operations</b> context menu option.  The Operations dialog displays.	
3	Select the Primary Key operation (its name begins with <b>PK_</b> ) and then click on the <b>Column</b> tab.	

Step	Action	See also
4	Click on the <b>New</b> button to add a column to the Primary Key, select a column from the <b>Column Name</b> list box, and then click on the <b>Save</b> button.	
5	Click on the <b>Hand</b> buttons (up and down arrow) to change the order of columns in the Primary Key, if necessary.	

#### Learn more

- [Creating Non-Clustered Primary Keys for SQL Server](#)<sup>[2358]</sup>

#### 13.1.5.1.2 Define a Primary Key Name Template

If you have specific modeling conventions that stipulate how Primary Keys are documented, you can customize the way in which Primary Keys are named by default. By changing the **Primary Key Name Template**, you adjust the name that the system assigns to the UML Operation representing the Primary Key.

Access   **Tools | Options | Source Code Engineering | Code Editors: DDL Name Templates**

#### Define the name template for a Primary Key

Step	Action	See also
1	Click on the <b>DDL Name Templates</b> button. The DDL Name Template dialog displays, showing the default name templates.	
2	Edit or replace the template in the <b>Primary Key Name</b> field. For example, if you want to display the Primary Key description as <i>PK_tablename_columnname</i> then change the <b>Primary Key Name</b> field to <i>PK_%tablename%_%columnname%.</i>	
3	Click on the <b>Save</b> button.	

### 13.1.5.1.3 SQL Server Non Clustered Keys

When you create a primary key in SQL Server, it is created with a 'clustered index' by default. Therefore when you model a primary key for SQL Server in an Enterprise Architect data model, the corresponding DDL creates a clustered index for that primary key by default.

Clustered indexes provide improved performance for accessing the column(s) involved, but only one clustered index is allowed per table.

In some situations, you might be more interested in the performance of columns other than the Primary Key, and therefore must change the default assignment of the clustered index. Enterprise Architect allows you to model this.

#### How to

To define a Primary Key as non-clustered for a SQL Server table

Step	Action	See also
1	Right-click on the table in a diagram and select the <b>Operations</b> context menu option. The Table Operations dialog displays.	
2	Highlight the Primary Key Operation and click on the <b>Extended Properties</b> button. The Database Operation Properties dialog displays.	
3	Select the <b>SQL Server Non Clustered Primary Key</b> checkbox.	
4	Click on the <b>Save &amp; Close</b> button.	

Subsequently, you can model an index for that table and define it as clustered instead.

#### Learn more

- [Indexes](#) 

### 13.1.5.2 Foreign Key

A Foreign Key consists of a collection of columns (UML Attributes) that together have some operational meaning - they enforce a relationship to access data in a related table via its Primary Key.

Foreign keys are modeled in Enterprise Architect as operations with the stereotype **FK**. The operation's parameters become the columns involved in the key.

It is not necessary to define a Foreign Key **just** to access another table through its Primary Key. Foreign Keys are a feature of some database management systems, providing 'extras' such as referential integrity checking that prevents the deletion of a record if its Primary Key value exists in some other table's Foreign Key. The same thing can be achieved programmatically.

Learn more

- [Create a Foreign Key](#)<sup>[2359]</sup>
- [Composite Foreign Key](#)<sup>[2361]</sup>
- [Define a Foreign Key Name Template](#)<sup>[2362]</sup>
- [Define a Foreign Key Index Template](#)<sup>[2363]</sup>

Learning Center topics

- (Alt+F1) | [Enterprise Architect](#) | [Database Engineering](#) | [Physical Data Model](#) | [Add Foreign Key](#)

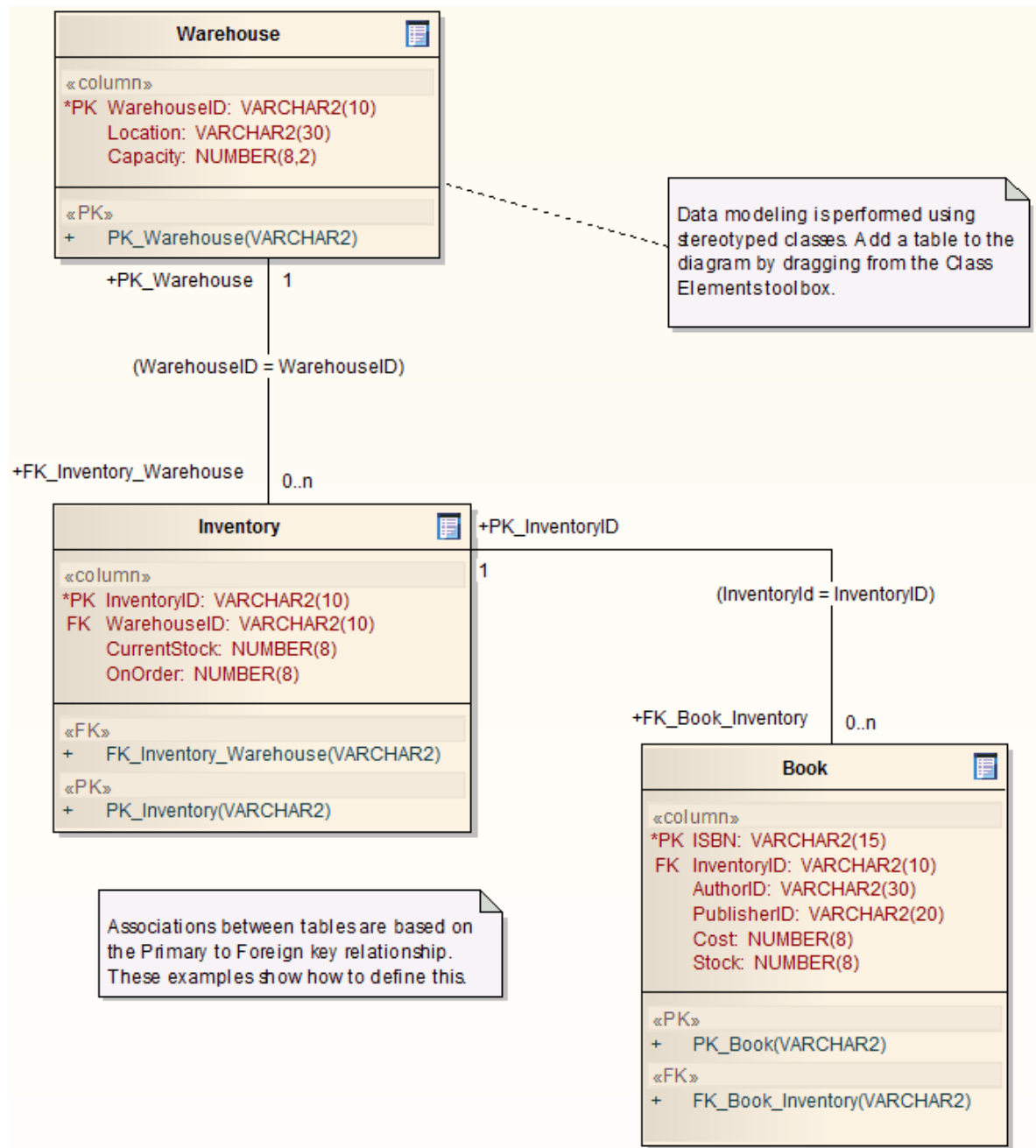
**13.1.5.2.1 Create a Foreign Key**Create a Foreign Key

Step	Action	See also
1	Locate the required Tables in a diagram. Both tables must have <b>defined database types</b> .	<a href="#">Set The Database Type</a> <sup>[2341]</sup>
2	Select an <i>Associate</i> connector in the Class Relationships page of the Toolbox.	
3	Click on the Table to contain the Foreign Key (source) and draw the connector to the target Table.	
4	Right-click on the connector and select the <b>Foreign Keys</b> context menu option. The Foreign Key Constraint dialog displays.	
5	The default foreign key name is set by the Foreign Key Name Template. To change the name to something other than the default provided by the template, select the <b>Override Template</b> checkbox and edit the foreign key name. Optionally, you can create a Foreign Key Index automatically using the <b>Create Index on Foreign Key</b> field. You can control the name of the automatically generated Foreign Key Index by setting the template.	<a href="#">Define Foreign Key Name Template</a> <sup>[2362]</sup>  <a href="#">Define Foreign Key Index template</a> <sup>[2363]</sup>
6	In the <b>Source:</b> panel and the <b>Target:</b> panel, click on the name of each of the two columns involved in the Foreign Key relationship.	
7	From the <b>On Delete</b> and/or <b>On Update</b> combo boxes, select the appropriate referential integrity constraint.	
8	Click on the <b>Apply</b> or <b>OK</b> buttons to automatically generate the Foreign Key	

Step	Action	See also
	operations.	

### Example

This example shows what a Foreign Key looks like in a diagram:



### Notes



- For MySQL databases, Foreign Keys are supported for InnoDB tables only



InnoDB is the default MySQL storage engine as of version 5.5; if you are modeling a MySQL database that is earlier than version 5.5 and you want to use Foreign Keys, you must set the **Engine** Tagged Value to **InnoDB**

#### Learn more

- [Set MySQL Options](#) <sup>[2342]</sup>

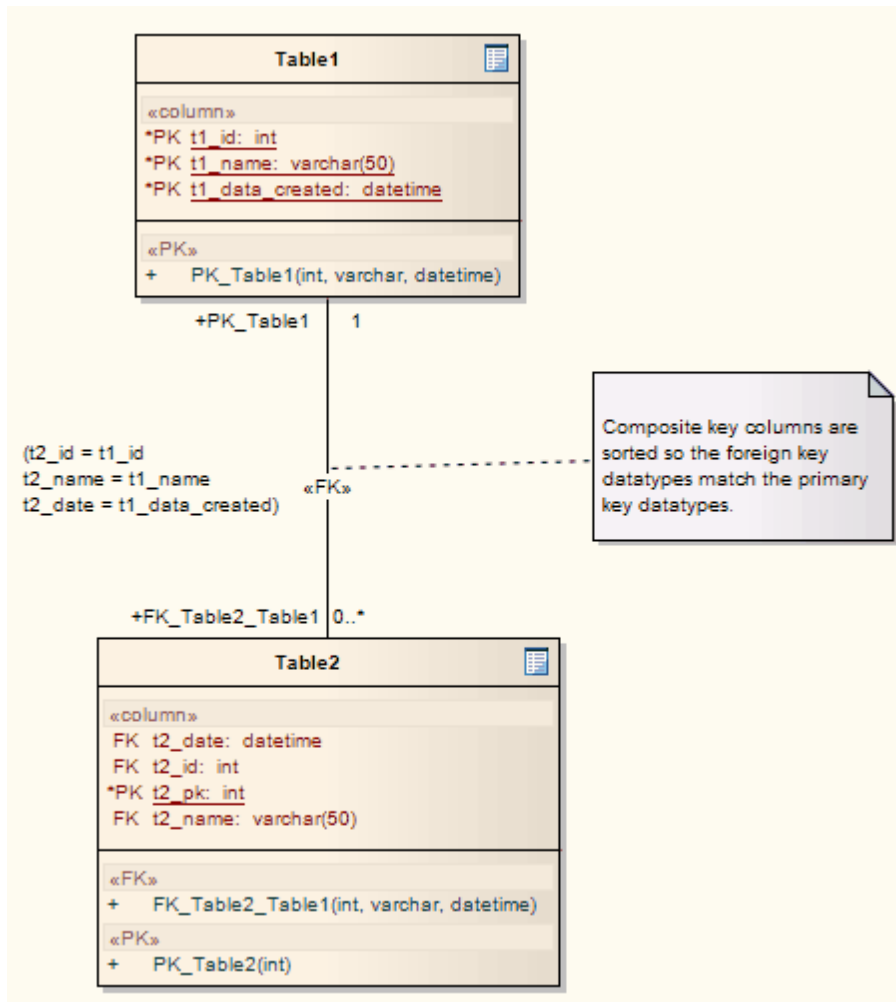
### 13.1.5.2.2 Composite Foreign Key

#### Create a composite Foreign Key

Step	Action	See also
1	Locate the required Tables in a diagram. Both tables must have <b>defined database types</b> .	<a href="#">Set The Database Type</a> <sup>[2341]</sup>
2	Select an <i>Associate</i> connector in the Class Relationships page of the Toolbox.	
3	Click on the Table to contain the Foreign Key (source) and draw the connector to the other Table (target).	
4	Right-click on the connector and select the <b>Foreign Keys</b> context menu option. The Foreign Key Constraint dialog displays.	
5	The default foreign key name is set by the Foreign Key Name Template. To change the name to something other than the default provided by the template, select the <b>Override Template</b> checkbox and edit the foreign key name.	<a href="#">Define Foreign Key Name Template</a> <sup>[2362]</sup>
6	In the <b>Source:</b> panel and the <b>Target:</b> panel, click on the names of the required columns in each panel.	
7	Click on the <b>OK</b> button. The Foreign Key columns are sorted according to data type to match the data types of the targeted composite Primary Key.	
8	If required, you can change the order of the key columns by clicking on the  and  buttons.	

### Example

This example shows what a Composite Foreign Key looks like in a diagram:



#### 13.1.5.2.3 Define a Foreign Key Name Template

If you have specific modeling conventions that stipulate how Foreign Keys are documented, you can customize the way Foreign Key relationships are named by default.

By changing the **Foreign Key Name Template**, you adjust the name that the system assigns to the Association connector and Operation that define the Foreign Key relationship.

**Access** [Tools | Options | Source Code Engineering | Code Editors: DDL Name Template](#)

#### Define the name template for a Foreign Key

Step	Action	See also
1	Click on the <b>DDL Name Template</b> button.	

Step	Action	See also
	The DDL Name Template dialog displays, showing the default name templates.	
<b>2</b>	Edit or replace the name template in the <b>Foreign Key Index</b> field. For example, if you want to display the Foreign Key description as: <i>FK_foreigntablename_FKcolumnname_primarytablename_PKcolumnname</i> then change the <b>Foreign Key Name</b> field to: <i>FK_%foreigntablename%_%fkcolumnname%_%primarytablename%_%pkcolumnname%</i>	
<b>3</b>	Click on the <b>Save</b> button.	

#### 13.1.5.2.4 Define a Foreign Key Index Template

If you have specific modeling conventions that stipulate how **Foreign Key Indexes** are documented, you can customize the way the indexes are named by default.

By changing the **Foreign Key Index Template**, you adjust the name the system assigns to the Index Operation that you can optionally define within the Foreign Key relationship.

**Access** [Tools](#) | [Options](#) | [Source Code Engineering](#) | [Code Editors](#): DDL Name Template

#### Define the Index template for a Foreign Key

Step	Action	See also
<b>1</b>	Click on the <b>DDL Name Template</b> button. The DDL Name Template dialog displays, showing the default name templates.	
<b>2</b>	Edit or replace the name template in the <b>Foreign Key Index</b> field. For example, if you want to create Foreign Key Indexes with the same name as the Foreign Keys, but include a prefix of <b>IX</b> , then change the <b>Foreign Key Index</b> field to: <i>IX%foreignkeyname%</i>	
<b>3</b>	Click on the <b>Save</b> button.	

#### Learn more

- [Create a Foreign Key](#) 

- [Define a Foreign Key Name Template](#) <sup>[2362]</sup>

### 13.1.6 Stored Procedures

Stored Procedures are subroutines for processing table data on the DBMS server. These routines can be called from an external application.

#### Topics

Topic	Detail	See also
<b>What is a Stored Procedure?</b>	<p>A stored procedure is a group of SQL statements that perform a specific task; they are used to group a set of operations or queries that can be executed on a database server.</p> <p>Stored Procedures are often used where there is a need to reduce network usage between clients and the DBMS servers.</p> <p>Enterprise Architect models stored procedures using stereotyped UML Classes and Operations. These can be generated with a schema for loading onto the DBMS server.</p>	<p><a href="#">Import Database Schema</a> <sup>[2377]</sup></p> <p><a href="#">Create a Stored Procedure</a> <sup>[2364]</sup></p> <p><a href="#">DBMS Product Conversion for a Package</a> <sup>[2349]</sup></p>

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Database Engineering | Physical Data Model | Stored Procedures**

#### 13.1.6.1 Create a Stored Procedure

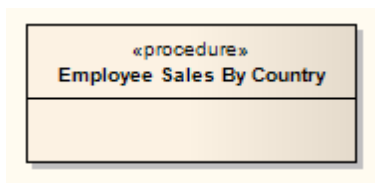
##### How to

To create a stored procedure as a UML Class, follow the steps below.

Step	Action	See also
1	Open the required diagram.	
2	From the Data Modeling page of the Toolbox ( <b>More tools   Data Modeling</b> ) drag the <i>Procedure</i> icon onto the diagram.	
3	If the Properties dialog does not automatically display, double-click on the element.	
4	<p>In the <b>Database</b> field click on the drop-down arrow and select the target DBMS to model.</p> <p>The field displays the default database if it has already been set.</p>	<a href="#">Set the Database Type</a> <sup>[2341]</sup>

Step	Action	See also
5	In the <b>Procedure definition</b> field, type the entire procedure text.	<a href="#">Languages supported</a> <sup>[2147]</sup> <a href="#">Options - Code Editors</a> <sup>[2250]</sup>
6	Click on the <b>OK</b> button.	
7	To rename the stored procedure, select the element, press <b>(F2)</b> and type the new name.	

### Example



### Notes

- Stored Procedures can also be defined under a single Class with the stereotype set to <<Stored Procedures>>, where each Stored Procedure is created as an operation under that Class; the scripting of these Stored Procedures can be viewed in the Operations dialog under **Behavior**

### Learn more

- [Generate DDL](#) <sup>[2380]</sup>
- [Import Database Schema](#) <sup>[2377]</sup>

## 13.1.7 Indexes

An **index** on a table is used to improve performance of lookups and sort operations on the table at the expense of additional storage. An index can be:




- (For all DBMSs) Column-based, for fast queries on table entries, using the columns specified in the index (the columns having already been defined in the table), or
- (For Oracle), function-based, as defined in a behavior expression
- (For SQL Server), include one or more columns in addition to the specified column, and a fill factor

An index is modeled as a stereotyped operation; on generating DDL, the necessary instructions for generating indexes are written to the DDL output.

**Access** **Project Browser | Right-click Table element | Operations Diagram | Right-click Table element | Features and Properties | Operations**

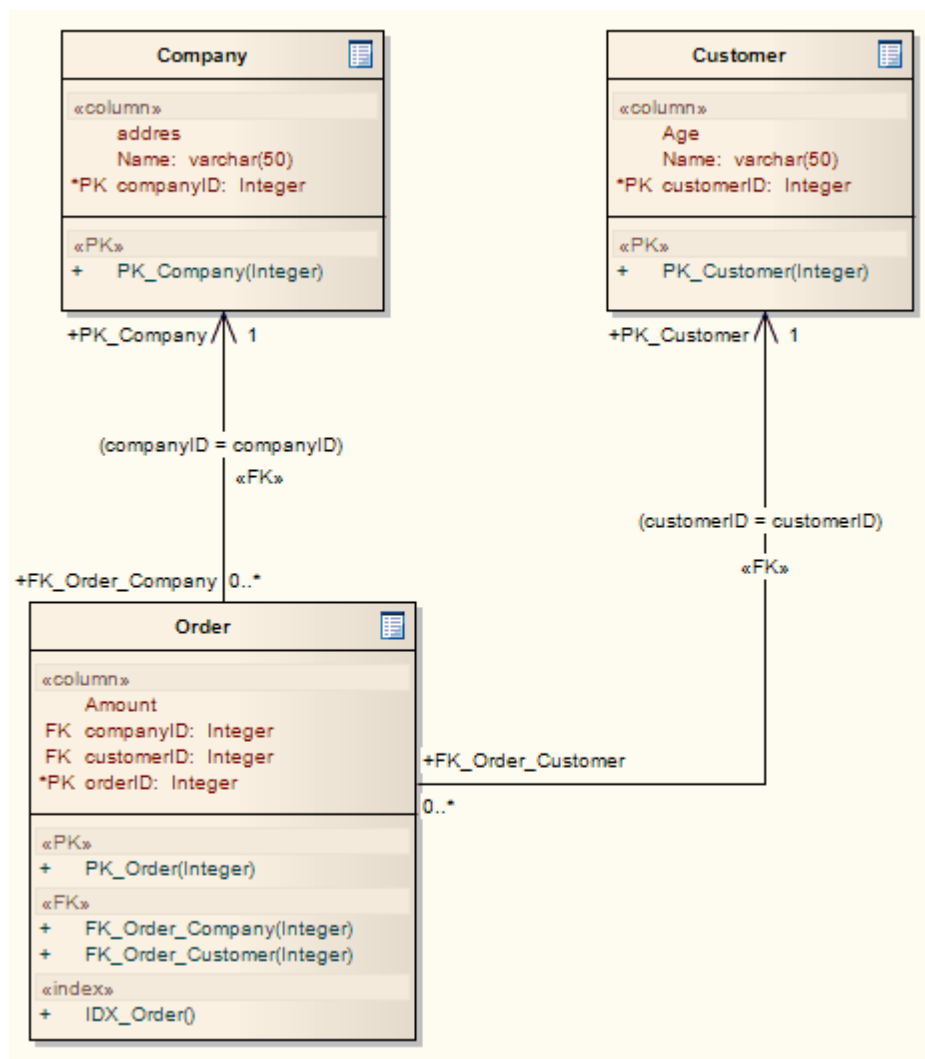
**Create an index**

Step	Action	See also
1	On the Operation dialog, add an operation with a name such as <i>IDX_CustomerID</i> . The <i>IDX_</i> prefix is optional, but helps to identify the operation.	
2	In the <b>Stereotype</b> field for the operation, click on the drop-down arrow and select <b>index</b> .	
3	Click on the <b>Save</b> button.	
4	By default, indexes are not created as <b>clustered</b> or <b>unique</b> , and are sorted in ascending order. If this is appropriate for your purposes, go to step 8.  Otherwise, click on the newly-created index operation in the <b>Operations</b> list and click on the <b>Extended Properties</b> button.  The Database Operation Properties dialog displays.	
5	Select the <b>Unique</b> and/or <b>Clustered</b> checkboxes as appropriate. <ul style="list-style-type: none"> <li>A Unique index cannot contain more than one instance of a combination of values across a set of columns</li> <li>Clustered indexes provide improved performance for accessing the columns; you can have one clustered index per table</li> </ul>	
6	Set the index sort order by selecting the <b>Ascending</b> or <b>Descending</b> radio button, as appropriate.	
7	Click on the <b>Save &amp; Close</b> button to return to the Operation dialog.	
8	If you prefer (and if the DBMS you are using supports it), you can type an index comment in the <b>Notes</b> field of the Operation dialog.	
9	It is not necessary to add any columns to an Oracle <b>function-based</b> index.  To model function-based indexes, click on the index operation in the <b>Operations</b> list and click on the Behavior page in the Properties list in the left hand panel of the dialog.  In the <b>Initial Code</b> field, type the index behavior expression, and click on the <b>Save</b> button.  Go to step 12.	
10	To add columns to a <b>column-based</b> index, click on the index operation in the <b>Operations</b> list and click on the Column page in the Properties list in the left hand	

Step	Action	See also												
	<p>panel of the dialog.</p> <p>Select the required columns from the <b>Column Name</b> drop-down list, in the required order. Click on the <b>Save</b> button after selecting each column from the drop-down list.</p> <p>If you prefer, you can adjust the sequence of columns using the 'up hand' and 'down hand' buttons.</p>													
11	<p>For SQL Server, to add included columns select the Tagged Values page from the Properties list and add the Tagged Values:</p> <ul style="list-style-type: none"><li>• <b>INCLUDE</b>, and in the <b>Value</b> field type the column names as a comma-separated list</li><li>• <b>FILLFACTOR</b>, and in the <b>Value</b> field type the numerical fill factor</li></ul> <p>For example:</p> <table><tr><th>Property</th><th>Value</th></tr><tr><td> <b>Constraint (IDX_CustomerID)</b></td><td></td></tr><tr><td>Is Unique</td><td>False</td></tr><tr><td>Is Clustered</td><td>False</td></tr><tr><td>Fill Factor</td><td>85</td></tr><tr><td>Include</td><td>Name, Address</td></tr></table>	Property	Value	 <b>Constraint (IDX_CustomerID)</b>		Is Unique	False	Is Clustered	False	Fill Factor	85	Include	Name, Address	
Property	Value													
 <b>Constraint (IDX_CustomerID)</b>														
Is Unique	False													
Is Clustered	False													
Fill Factor	85													
Include	Name, Address													
12	Click on the <b>Close</b> button.													

### Example

In the example below, the *Order* element shows what an index looks like in a diagram:

Learn more

- [Create Columns](#) 

Learning Center topics

- (Alt+F1) | **Enterprise Architect** | **Database Engineering** | **Physical Data Model** | **Add Index**

**13.1.8 Triggers**Topics

Topic	Detail	See also
<b>What is a Trigger?</b>	<p>A trigger is an operation automatically executed as a result of the modification of data in the database, and usually ensures consistent behavior of the database.</p> <p>For example, a trigger might be used to define validations that must</p>	



Topic	Detail	See also
	<p>be performed every time a value is modified, or might perform deletions in a secondary table when a record in the primary table is deleted.</p> <p>In Enterprise Architect, a trigger is modeled as a stereotyped operation.</p>	

#### Learn more

- [Create a Trigger](#) 



#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Database Engineering | Physical Data Model | Create Trigger**

### 13.1.8.1 Create a Trigger

#### How to

To create a Trigger

Step	Action	See also
1	Locate the required table in either a diagram or the Project Browser.	
2	Right-click on the table and select the <b>Operations</b> context menu option. The Operations dialog displays.	
3	Add an operation with a name such as <i>TRG_OnCustomerUpdate</i> . The <i>TRG_</i> prefix is optional, but helps to identify the operation.	
4	In the <b>Stereotype</b> field select <b>trigger</b> .	
5	Click the <b>Save</b> button.	
6	Select the trigger operation, then the Behavior tab.	
7	In the <b>Initial Code</b> field, enter the entire trigger code including the <i>CREATE_TRIGGER</i> statement.	<a href="#">Languages supported</a>  <a href="#">Options - Code Editors</a> 

Step	Action	See also
8	Click on the <b>Save</b> button.	

### 13.1.9 Create a Check Constraint

A **Check Constraint** enforces domain integrity by limiting the values that are accepted by a column.

**Access** [Project Browser or diagram](#) | [Table context menu](#) | [Operations](#)

#### Create a Constraint

Step	Action	See also
1	On the Operations dialog, add an operation with a name such as <i>CHK_ColumnName</i> . The <i>CHK_</i> prefix is optional, but helps to identify the operation.	
2	In the <b>Stereotype</b> field select <b>check</b> .	
3	Click on the <b>Save</b> button.	
4	Optionally, in the <b>Notes</b> field you can type a check constraint comment.	
5	Click on the check constraint operation in the <b>Operations</b> list, then select the Behavior tab from the Properties panel on the left of the dialog.	
6	In the <b>Initial Code</b> field, enter the entire check constraint clause. For example, <b>col1 &lt; 1000</b> .	
7	Click on the <b>Save</b> button.	

#### Learning Center topics

- (Alt+F1) | [Enterprise Architect](#) | [Database Engineering](#) | [Physical Data Model](#) | [Check Constraint](#)

### 13.1.10 Default Constraints

*Default Constraints* apply to SQL Server only, and are supported to maintain compatibility with earlier versions.

**Access**   **Right-click on Table | Operations**

#### Create a Default Constraint

Step	Action
1	On the Operations Properties dialog, in the <b>Name</b> field, type the new operation name in the format <b>DF_&lt;ColumnName&gt;</b> . The <b>DF_</b> is optional, but helps to identify the operation.
2	In the <b>Stereotype</b> field type or select the value <b>default</b> .
3	Click on the <b>Save</b> button.
4	In the <b>Operations</b> list, click on the default operation name, then click on <b>Column</b> in the left-hand panel.
5	Select the appropriate column from the <b>Column Name</b> drop down list.
6	Click on <b>Behavior</b> in the left-hand panel.
7	In the <b>Initial Code</b> field, type the default value.
8	Click on the <b>Save</b> button.

### 13.1.11 Views

A **Database View** defines a subset of the database, aggregated into a dynamic 'virtual' table. The information presented by a Database View is not physically replicated, rather it is automatically derived based on the query that defines the view.

In Enterprise Architect, you model Database Views as stereotyped UML Classes.

#### Learn more

- [Create a View](#)<sup>[2372]</sup>

Learning Center topics

- (Alt+F1) | **Enterprise Architect | Database Engineering | Physical Data Model | Create View**

**13.1.11.1 Create a View**Create a Database View

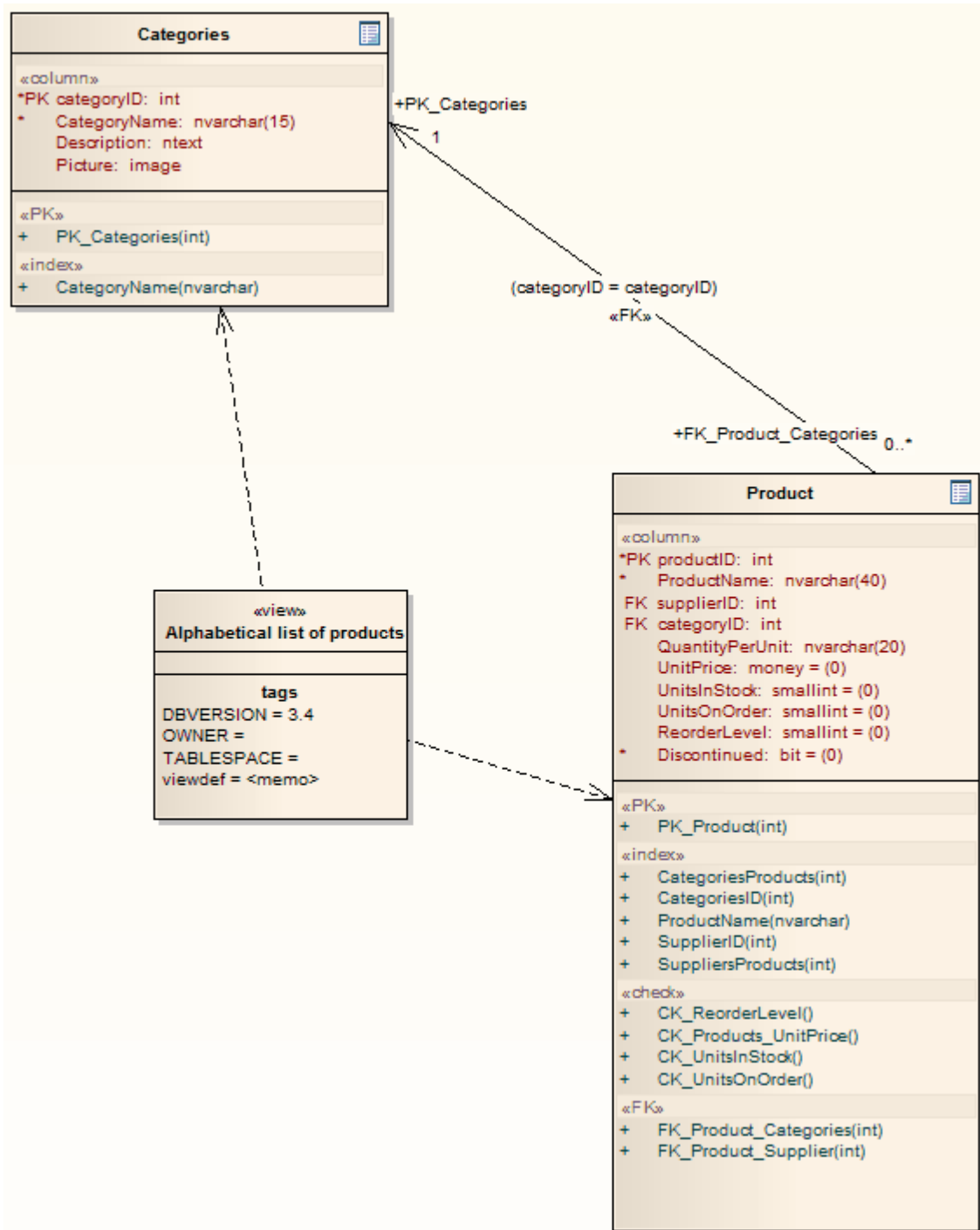
Step	Action	See also
1	On the Data Modeling page of the Toolbox ( <b>More tools</b>   Data Modeling), drag the View icon onto your Data Modeling diagram.	
2	If the View Properties dialog does not immediately display, double-click on the element.	
3	From the Database drop-down list, select the target DBMS to model. The default database displays if it has already been set.	<a href="#">Set the Database Type</a> [2341]
4	Click on the OK button.	
5	To rename the View, select the element, press (F2) and type the new name.	

Create a View definition

Step	Action	See also
1	Create a Dependency connector from the View to the table or tables on which the View depends.	
2	Double-click on the View to display the Properties dialog. The tables are now listed in the <b>Dependencies</b> field.	
3	In the <b>View definition</b> field, type the full view definition. The code editor provides intelli-sense for basic SQL keywords and functions.	<a href="#">Options - Code Editors</a> [2250] <a href="#">Intelli-sense</a> [2160]
4	Click on the <b>OK</b> button to save your definition.	

Step	Action	See also

### Example



### Notes

- The View definition and certain other parameters are held as Tagged Values; the View definition is held in the *viewdef* memo Tagged Value

You can select and view the *viewdef* Tagged Value in the Tagged Values window, and include it in document reports by inserting the **valueOf(viewdef)** field in the *Package::Package Element* or *Package::Element* sections.

#### Learn more

- [Tagged Values](#) <sup>[1134]</sup>
- [Report on Tagged Values](#) <sup>[2698]</sup>

### 13.1.12 Oracle Packages

#### Create an Oracle package

Step	Action	See also
1	Add a Class element to your data model.	
2	Open the Properties dialog for the element and, in the <b>Stereotype</b> field, type the value <b>Package</b> .	
3	For the package specification, create an Operation with the name <i>Specification</i> and with no return type.	
4	Open the Properties dialog for the <i>Specification Operation</i> and, on the Behavior tab, type the entire package specification into the <b>Initial Code</b> field.	
5	For the package body, create an Operation with the name <i>Body</i> and with no return type.	
6	Open the Properties dialog for the <i>Body Operation</i> and, on the Behavior tab, type the entire package body into the <b>Initial Code</b> field.	

#### Learn more

- [Initial Code](#) <sup>[1019]</sup>

## 13.2 Supported Databases

Modeling and reverse engineering of database schema is supported for the following DBMS products:

- DB2
- Firebird/InterBase\*
- Informix
- Ingres
- MS Access 97, 2000, 2003
- Access 2007
- MS SQL Server 2000, 2005, 2008, 2012
- SQL Server Express 2005 and 2008
- MySQL
- SQLite
- Oracle 9i, 10g and 11g
- PostgreSQL
- Sybase Adaptive Server Anywhere (Sybase ASA)
- Sybase Adaptive Server Enterprise (Sybase ASE)

### Notes

- \* Firebird 1.5 database tables can be modeled and generated as InterBase tables; Firebird tables can be imported, but are treated as InterBase tables
- You can download data types for SQL Server (2005, 2008, 2012) Informix and MS Access 2007 from the Resources page of the Sparx Systems web site

### Learn more

- [Online Resources](#)

## 13.3 Import Database Schema

Enterprise Architect provides facilities for reverse engineering DBMS schemas, enabling you to analyze and remodel legacy database systems and then export them to the existing or an alternative DBMS.

By connecting to a live database via ODBC, you can import the database schema into a standard UML model. Subsequent imports enable you to update your model from the live database.

Enterprise Architect supports importing database tables, stored procedures and views from an ODBC data source. Tables are imported as stereotyped Classes with suitable data definitions for the source DBMS.

### Import database schema and objects

Step	Action	See also
1	<p>Select a package in the Project Browser, into which to import the database schema.</p> <p>Create a data modeling diagram in this package.</p>	<a href="#">Create a Data Model Diagram</a> <sup>2336</sup>
2	<p>To import, either:</p> <ul style="list-style-type: none"> <li>• Right-click on the package and select the <b>Code Engineering   Import DB Schema from ODBC</b> context menu option</li> <li>• Right-click on the diagram and select the <b>Import DB schema from ODBC</b> context menu option, or</li> <li>• Select the <b>Tools   Database Engineering   Import DB schema from ODBC</b> menu option</li> </ul> <p>The Import DB Schema from ODBC Source dialog displays.</p>	
3	<p>In the <b>Database</b> field, click on the ( ... ) (<b>Browse</b>) button and select a suitable ODBC data source from the ODBC dialog (ODBC must be installed and configured on your machine for this to work correctly).</p> <p>When you have selected the data source, the <b>Database</b> field shows the DBMS, the database server ID and the database name, separated by full stops; that is:</p> <p><i>dbms.dbserver.database.</i></p>	<a href="#">Select a Suitable Data Source</a> <sup>2378</sup> See <i>Notes</i> below, concerning 32-bit ODBC drivers.
4	<p>You can filter objects to be retrieved from the database by schema or owner.</p> <p>In the <b>Schema/Owner</b> field, type the schema/owner as a single entry, or as a comma-separated list.</p> <p>The filter is useful for databases that support multiple schemas or owners, such as SQL Server 2005/2008/2012, Oracle, PostgreSQL and DB2 Express.</p> <p>By default, when importing database objects from Oracle, Enterprise Architect inserts the Oracle user name in this field to avoid unnecessary retrieval of large numbers of objects not owned by that user.</p>	



Step	Action	See also
5	Select the appropriate checkboxes to import system tables and views, user views, triggers and/or Oracle packages.	<a href="#">Tables and Columns</a> <sup>[2338]</sup> <a href="#">Views</a> <sup>[2371]</sup> <a href="#">Triggers</a> <sup>[2368]</sup>
6	<p>Select the appropriate checkboxes to import:</p> <ul style="list-style-type: none"> <li>• Stored Procedures</li> <li>• User Functions</li> <li>• User Sequences</li> </ul> <p>These items can be imported as a Class for each Stored Procedure, User Function or User Sequence. Alternatively, all Stored Procedures can be imported as a set of operations contained under a Class. A similar Class can be created for all User Functions and User Sequences.</p> <p><b>Import as individual Classes</b></p> <p>The script is imported into separate Classes. The Properties dialog for each of these Classes supports a context sensitive editor for editing the script.</p> <p><b>Import as Operations</b></p> <p>The selected items are imported to a single Class as Operations (methods); you can view and edit them through the Operations Properties dialog of the parent Class.</p>	<a href="#">Stored Procedures</a> <sup>[2364]</sup>
7	<p>In the Synchronization panel, select the appropriate option to determine whether the existing Classes are to be updated, or the database objects imported as new objects.</p> <p>If you select the <b>Synchronize existing classes</b> option, also select the appropriate checkboxes to determine whether model comments, column default values and/or table constraints are to be retained or overwritten with the comments, values and constraints of the imported objects.</p>	
8	Click on the <b>Import</b> button to start the import.	
9	Select the database objects to import.	<a href="#">Select Tables</a> <sup>[2378]</sup> <a href="#">Imported Class Elements</a> <sup>[2379]</sup>



**Notes**

- Enterprise Architect requires **32-bit ODBC** drivers to connect to a repository through ODBC; to set up

the ODBC configuration on **64-bit** clients, run the 32-bit ODBC Data Source Administrator from `C:\Windows\SysWOW64\odbcad32.exe`

- The ODBC connection should use the ODBC driver available from the DBMS vendor, such as MySQL's ODBC driver for MySQL, and Oracle's ODBC driver for Oracle; drivers provided by third-party vendors are not supported, including the Microsoft ODBC driver for Oracle
- You can import a suitable ODBC driver for SQLite from <http://www.ch-werner.de/sqliteodbc/>
- If setting up a ODBC connection for reverse engineering, the default settings are sufficient
- If you are importing database schema from an MS Access Jet 4.0 database, please ensure that you have selected the **Use Jet 4.0** checkbox on the General page of the Options dialog, otherwise the Jet 3.5 routines are loaded; you must restart Enterprise Architect after selecting the checkbox
- Additional data types are available from the *Data Modeling Data Types* section of the Resources page on the Sparx Systems website
- If no diagram is open, the **Package Only** radio button defaults to selected and the options are disabled; if the open diagram is in the selected package, you can select either option

#### Learn more

- [Import Source Code](#) 
- [General Options](#) 
- [Data Modeling Data Types](#) (Online Resource)

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Database Engineering | Import Schema**

### 13.3.1 Select a Data Source

#### Topics

Topic	Detail	See also
<b>General Usage</b>	<p>To import DDL from existing data sources, you must have a suitable ODBC connection installed and configured.</p> <p>From the Import DB Schema from ODBC Source dialog you can select the ODBC data source using the standard windows ODBC set-up dialog; click on the data source name and then click on the <b>OK</b> button.</p>	

### 13.3.2 Select Tables

When you have opened the ODBC data source, Enterprise Architect acquires a list of database objects suitable for importing. This is presented in a list form for you to select from.

Highlight the schema to import from the top list. Alternatively, select individual objects from the lower list.

#### Topics

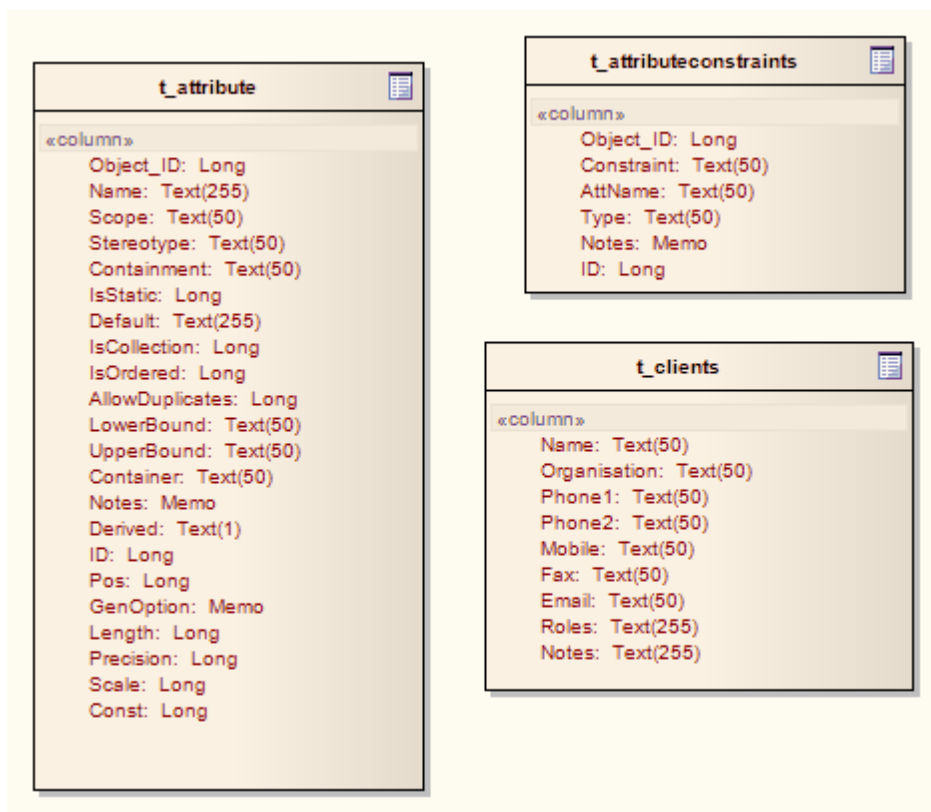
Topic	Detail	See also
<b>Selection Shortcuts</b>	<ul style="list-style-type: none"> <li>To select all objects, click on the <b>Select All</b> button</li> <li>To clear all objects, click on the <b>Select None</b> button</li> <li>Hold down ( <b>Ctrl</b> ) while clicking to select multiple objects</li> <li>Hold down ( <b>Shift</b> ) and click on the first and last objects in a range to select that range</li> </ul> <p>When you have selected the objects, click on the <b>OK</b> button.</p>	

### 13.3.3 The Imported Class Elements

When you import DDL table definitions they are converted to stereotyped Classes according the *UML Data Modeling Profile*.

The image below shows some example tables imported into the model using an ODBC data connection.

#### Example



## 13.4 Generate DDL

Enterprise Architect enables you to generate DDL from your model for Tables, Stored Procedures and Views. SQL script can be generated for a single element (a Table) or for a whole package.

Having generated DDL, you can view it using the Source Code editors, by clicking on the table or package in the Project Browser or a diagram, and pressing **(F12)**, **(Ctrl+E)** or **(Alt+7)**.

The editor cannot parse DDL, and therefore does not show a structure tree for a DDL file in the left-hand panel of the Source Code window or view.

### Learn more

- [Generate DDL for Tables](#)<sup>[2380]</sup>
- [Generate DDL for Packages](#)<sup>[2381]</sup>
- [Editing Source Code](#)<sup>[2146]</sup>

### Learning Center topics

- **(Alt+F1) | Enterprise Architect | Database Engineering | Generate DDL**



### 13.4.1 Generate DDL For a Table

**Access** **Diagram Table element context menu | Generate DDL**

### Reference

On the Generate DDL dialog, generate DDL for a table as indicated below:

Step	Action	See also
<b>Path</b>	Click on the ( ... ) (Browse) button to set up the file location and file name of the script to create.	
<b>Comment Level</b>	To include comments in the DDL, select the appropriate level. For example, select <b>Column</b> to include comments on columns, or <b>All</b> to include comments on all structures.	
<b>Option Checkboxes</b>	Select the checkbox for each type of inclusion you require; for example, to include a 'drop table'. command in the script, select the <b>Create Drop SQL</b> checkbox.  Deselect the checkboxes for inclusions you do not require.	
<b>Generate</b>	Click on this button to create the DDL.	
<b>View</b>	Click on this button to view the output.	<a href="#">Editing Source Code</a>

Step	Action	See also
	The viewer defaults to the Enterprise Architect default code editor. However, you can define an alternative default DDL editor on the Options dialog ( <b>Tools   Options   Source Code Engineering   Code Editors</b> ).	 <a href="#">Options - Code Editors</a> 

**Notes**

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Generate Source Code and DDL** permission to generate DDL
- Some checkboxes display only if the appropriate database is defined for the table; for example, **IF EXISTS** displays only if the database for the table is PostgreSQL, and **SEQ\_** and **\_SEQ** radio buttons display only if the database for the table is Oracle
- For a PostgreSQL database, you must select the **Generate Sequences** checkbox to enable auto increment columns to be created
- If generating Oracle sequences, you must always select the **Generate Triggers** and **Generate Sequences** checkboxes - this ensures that a pre-insert trigger is generated to select the next sequence value to populate the column; also set the **AutoNum** property to **True** in the column properties
- If generating Oracle sequences, to generate the sequence name and trigger name with the syntax **SEQ\_<sequence\_name>** and **TRG\_<trigger\_name>**, select the **SEQ\_** option; conversely, to generate them with the syntax **<sequence\_name>\_SEQ** and **SET\_<trigger\_name>**, select the **\_SEQ** option

**Learn more**

- [Permission List](#) 
- [Create Columns](#) 

**13.4.2 Generate DDL for a Package**

When you have defined a package in your database model, you can generate DDL for that package, and also compare the DDL with the database.

**Access** **Project Browser package context menu | Code Engineering | Generate DDL**, or **Select package Tools | Database Engineering | Generate Package DDL**

**Generate DDL for a package**

Step	Action	See also
<b>1</b>	On the Generate Package DDL dialog, select the checkbox against each inclusion required.  Deselect the checkboxes for inclusions you do not require.	
<b>2</b>	To recursively generate DDL, select the <b>Include All Child Packages</b> checkbox.	

Step	Action	See also
3	<p>Select the <b>Save Generated Order</b> checkbox to save the order in which the objects are generated.</p> <p>This is useful where the order is changed to resolve object dependencies; the order is saved when you click on the <b>Generate</b> button.</p>	
4	If you click the <b>Refresh</b> button with the <b>Save Generated Order</b> checkbox <i>un</i> checked, the objects are restored to their original order.	
5	<p>Click on the <b>Generate</b> button to proceed.</p> <p>Enterprise Architect prompts you for file names as the process executes.</p>	
6	<p>To view the output, click on the <b>View</b> button.</p> <p>The viewer defaults to the Enterprise Architect default code editor. However, you can define an alternative default DDL editor on the Options dialog (<b>Tools   Options   Source Code Engineering   Code Editors</b>).</p>	<a href="#">Editing Source Code</a> <sup>[2146]</sup> <a href="#">Options - Code Editors</a> <sup>[2250]</sup>

#### Compare the generated DDL with the database

Step	Action	See also
1	<p>On the Generate Package DDL dialog, click on the <b>Compare</b> button.</p> <p>The Compare With Database dialog displays.</p>	
2	Click on the <b>Browse ( ... )</b> button and locate the required database on the Select Data Source dialog.	
3	For an Oracle database, if required you can also specify the Owner in the <b>Schema/Owner</b> field.	<a href="#">Set up an Oracle ODBC Driver</a> <sup>[254]</sup>
4	<p>Click on the <b>OK</b> button to perform the comparison.</p> <p>The Comparison Database dialog displays with the results of the comparison.</p> <p>Click on each table name to review information on that table.</p>	
5	<p>Click on the:</p> <ul style="list-style-type: none"> <li>• <b>Save</b> button to save the altered DDL statements to a file</li> <li>• <b>View</b> button to open the saved file in a text editor</li> <li>• <b>Copy</b> button to copy the altered DDL statements to the clipboard</li> </ul>	

--	--	--

### Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Generate Source Code and DDL** permission to generate DDL
- Some checkboxes display only if the appropriate database is defined for the table; for example, **IF EXISTS** displays only if the database for the table is PostgreSQL, and **SEQ\_** and **\_SEQ** radio buttons display only if the database for the table is Oracle
- For a PostgreSQL database, you must select the **Generate Sequences** checkbox to enable auto increment columns to be created
- If generating Oracle sequences, you must always select the **Generate Triggers** and **Generate Sequences** checkboxes - this ensures that a pre-insert trigger is generated to select the next sequence value to populate the column; also set the **AutoNum** property to **True** in the column properties
- If generating Oracle sequences, to generate the sequence name and trigger name with the syntax **SEQ\_<sequence\_name>** and **TRG\_<trigger\_name>**, select the **SEQ\_** option; conversely, to generate them with the syntax **<sequence\_name>\_SEQ** and **SET\_<trigger\_name>**, select the **\_SEQ** option

### Learn more

- [Permission List](#) <sup>[329]</sup>
- [Create Columns](#) <sup>[2345]</sup>

## 13.5 Data Modeling Notations

Enterprise Architect supports three diagram notations for data modeling; UML 2.1, IDEF1X and Information Engineering.

**Access** [Diagram](#) | [Properties](#) > [Connectors](#) | [Connector Notations](#)

### Topics

Topic	Detail	See also
<b>UML 2.1</b>	The standard UML 2.1 notation for connectors.	
<b>Information Engineering</b>	The Information Engineering (IE) connection style.	
<b>IDEF1X</b>	Integrated Definition Methods IDEFX1 connection style.	

### Notes

- The default notation for the Data Modeling diagram is Information Engineering

### Learn more

- [Connectors Tab](#)<sup>[831]</sup>
- [Entity Relationship Diagrams \(ERDs\)](#)<sup>[1939]</sup>



**Part**

---



## 14 SOA and XML

Service Oriented Architecture (SOA) is an architectural paradigm for defining how people, organizations and systems provide and use services to achieve results.

A service is an offer of value to another through a well-defined interface and available to a community (which may be the general public). A service results in work provided to one by another.





Service Oriented Architecture (SOA) is a way of organizing and understanding ( representations of ) organizations, communities and systems to maximize agility, scale and interoperability. The SOA approach is simple - people, organizations and systems provide services to each other. These services allow us to get something done without doing it ourselves or even without knowing how to do it - enabling us to be more efficient and agile. Services also enable us to offer our capabilities to others in exchange for some value - thus establishing a community, process or marketplace. The SOA paradigm works equally well for integrating existing capabilities as for creating and integrating new capabilities.

(Derived from *Service oriented architecture Modeling Language (SoaML) - Specification for the UML Profile and Metamodel for Services (UPMS)* (OMG document ad/2008-11-01); pp. 25-26.)

In modeling and developing a complete Service Oriented Architecture in Enterprise Architect, you can work with any or all of:

- **XML Schema Definition (XSD)**, also known as **XML Schema** - an XML technology that is used to specify the rules to which an XML document must adhere
- **Web Services Description Language 1.1 (WSDL)** - a key XML-based language for describing **web** services
- **Service oriented architecture Modeling Language (SoaML)** - a standard method of designing and modeling SOA solutions using the Unified Modeling Language (UML)
- **Service-Oriented Modeling Framework (SOMF)** - a service-oriented development life cycle methodology, offering a number of modeling practices and disciplines that contribute to successful service-oriented life cycle management and modeling

### Learn more

- [XML Schema - XSD](#) <sup>[2387]</sup>
- [Web Services - WSDL](#) <sup>[2423]</sup>
- [SoaML](#) <sup>[2449]</sup>
- [SOMF 2.1](#) <sup>[2454]</sup>

### Learning Center topics

- (Alt+F1) | **Enterprise Architect | SOA and XML Engineering**

## 14.1 XML Schema - XSD

**XML Schema Definition** (XSD), also known as **XML Schema**, is a World Wide Web Consortium (**W3C**) XML technology that is used to specify the rules to which an XML document must adhere. XSD support is critical for the development of a complete **Service Oriented Architecture** (SOA), and the coupling of UML 2.4.1 and XML provides the natural mechanism for specifying, constructing and deploying XML-based SOA artifacts within an organization.

The **UML Profile for XSD** specifies a set of stereotypes, Tagged Values and constraints that can be applied to the UML model in order to change specific aspects of the resulting schema. Enterprise Architect provides native support for the XSD Profile through the **XML Schema** page of the **Diagram Toolbox**. The XSD Profile supported by Enterprise Architect is an adaptation of the profile defined in the publication *Modeling XML Applications with UML*.

Working with the XSD Profile through Enterprise Architect, you can rapidly model, forward engineer and reverse engineer XML Schema.

### Learn more

- [Model XSD](#) <sup>[2387]</sup>
- [XML from Abstract Class Models](#) <sup>[2413]</sup>
- [Generate XSD](#) <sup>[2417]</sup>
- [Import XSD](#) <sup>[2420]</sup>
- [W3C XML Schema](#) (Online Resource)
- [XML Schema Generation](#) (Online Resource)
- *Modeling XML Applications with UML - Practical e-Business Applications*, Object Technology Series, David Carlson, UK, April 2001

### Learning Center topics

- (Alt+F1) | **Enterprise Architect | SOA and XML Engineering | XSD | Getting Started**

















### 14.1.1 Model XSD






You can model XML schemas at two levels, using UML Class diagrams that:

- Have no XML schema-specific implementation details, to be generated directly by Enterprise Architect's **Schema Generator**; the generator applies a set of **default mappings** to convert the abstract model package to a **W3C XML Schema** (XSD) file
- Are refined with XML schema-specific definitions using the XML Schema pages of the Diagram Toolbox, which provides the structures of the **UML profile for XSD**

### Model an XML Schema

Step	Action	See also
1	In the Project Browser, create the top-level project structure you need (Model and Views), and click on the appropriate View.	<a href="#">Models</a> <sup>[753]</sup> <a href="#">Views</a> <sup>[769]</sup>

2	Click on the <b>New Package</b> icon in the Project Browser toolbar. The New Model Package dialog displays.	<a href="#">Project Browser Toolbar</a> 
3	In the <b>Name</b> field type the name of the new package, and select the <b>Automatically add new diagram</b> and <b>Open new diagram</b> checkboxes. Click on the <b>OK</b> button. The New Diagram dialog displays.	<a href="#">Add a Package</a> 
4	In the <b>Name</b> field type the name of the new diagram. In the Select From panel select <b>UML Structural</b> , and in the Diagram Types panel select <b>Class</b> .	<a href="#">Add New Diagrams</a> 
5	Click on the <b>OK</b> button. The new Class diagram opens in the Diagram View, with the Class pages displaying in the Diagram Toolbox. At this point you can either: <ul style="list-style-type: none"> <li>Create a Class diagram using the Class toolbox icons (click on the link to the right), or</li> <li>Create a tailored XML Schema diagram using the XML Schema pages of the Diagram Toolbox (continue to step 6)</li> </ul>	<a href="#">Diagram Toolbox</a>  <a href="#">Class Toolbox</a>  <a href="#">XML from Abstract Class Models</a> 
6	Click on the <b>More tools</b> option in the Toolbox and select <b>Extended   XML Schema</b> to display the XML Schema Toolbox pages.	<a href="#">XML Schema Toolbox</a> 
7	Click on the <b>Schema</b> icon from the Toolbox and drag it into the Class diagram. The XSD schema Properties dialog displays. Complete this dialog, and click on the <b>OK</b> button. The New Diagram dialog displays.	<a href="#">Schema Package</a> 
8	Again, in the <b>Name</b> field type the name of the new diagram. In the Select From panel select <b>UML Structural</b> , and in the Diagram Types panel select <b>Class</b> . Click on the <b>OK</b> button.	
9	An <i>XSDschema</i> stereotyped package is created in the Project Browser and on the diagram, with a child Class diagram. Double-click on the Package on the diagram to open the child Class diagram, and use the constructs from the XML Schema Toolbox to model the XML Schema.	<a href="#">Global Element</a>  <a href="#">Local Element</a>  <a href="#">Global Attribute</a>  <a href="#">Local Attribute</a>  <a href="#">Attribute Group</a>  <a href="#">Complex Type</a>  <a href="#">Model Group</a>  <a href="#">Group</a> 

		<a href="#">Simple Type</a>  <sup>[2402]</sup> <a href="#">Union</a>  <sup>[2408]</sup> <a href="#">Enumeration</a>  <sup>[2412]</sup> <a href="#">Any</a>  <sup>[2405]</sup> <a href="#">Any Attribute</a>  <sup>[2407]</sup>
--	--	---

### Notes

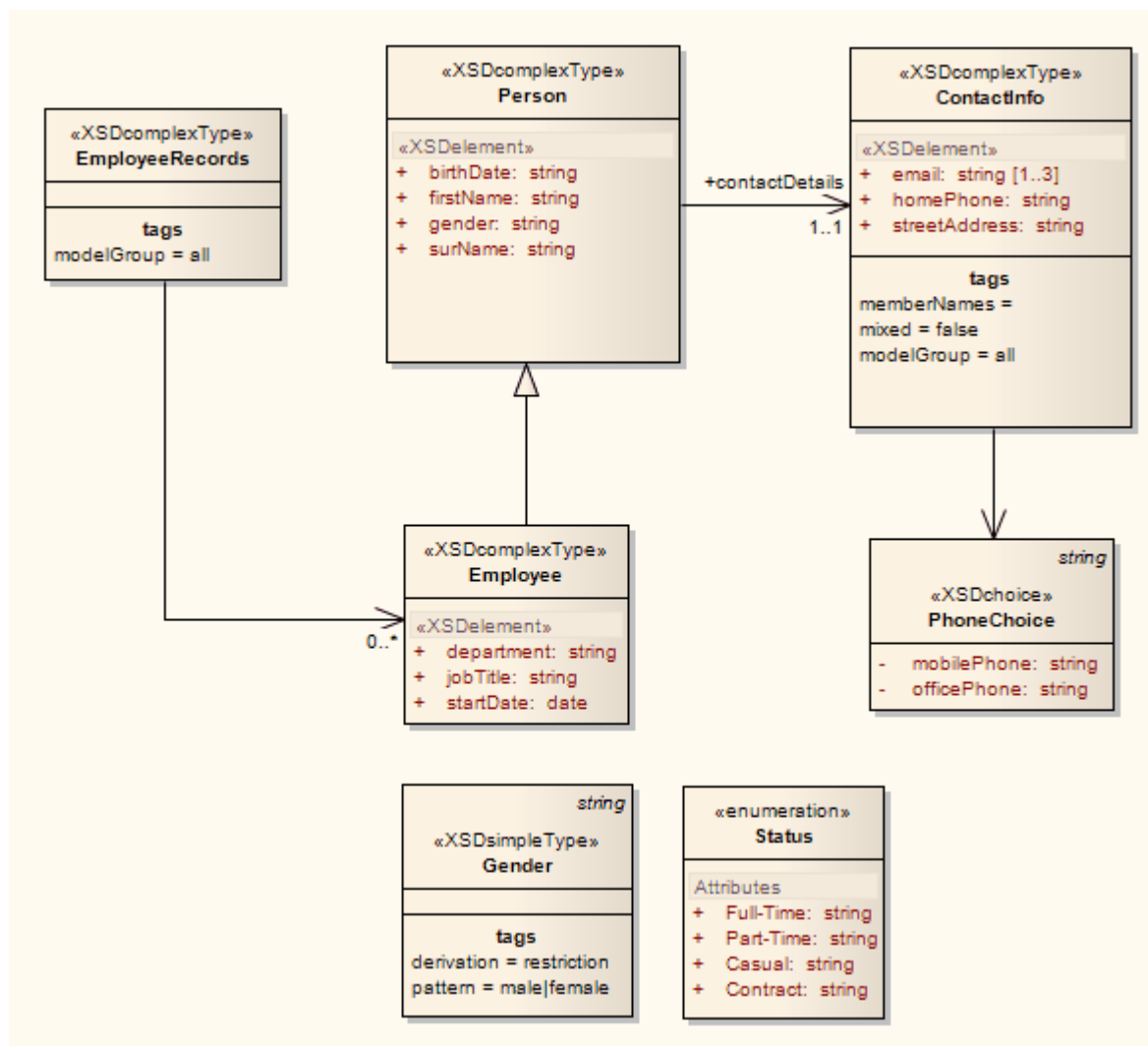
- The UML attributes of the Classes map directly to XML elements or attributes
- Classes in an XML Schema model have no methods since there is no meaningful correspondence between Class methods and XSD constructs
- **Modeling Restrictions** - the following XML Schema constructs cannot be modeled in Enterprise Architect:
  - appinfo
  - field
  - key
  - keyref
  - notation
  - redefine
  - selector
  - substitutionGroup
  - unique

### Learn more

- [Example XML Schema Diagram](#)  <sup>[2389]</sup>

#### 14.1.1.1 Example XML Schema Diagram

This example diagram shows a Class diagram containing XSD-specific elements created using the XSD Schema pages of the Diagram Toolbox. The diagram models an employee records system.



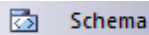
#### Learn more

- [XML Schema Toolbox](#)<sup>[818]</sup>
- [Complex Type](#)<sup>[2401]</sup>
- [Simple Type](#)<sup>[2402]</sup>
- [Model Group](#)<sup>[2410]</sup>
- [Local Element](#)<sup>[2394]</sup>
- [Enumeration](#)<sup>[2412]</sup>

#### 14.1.1.2 Schema Package

An **«XSDschema»** stereotyped package acts as a container for the XSD constructs, from which XML Schema can be generated. All Classes in the package are defined within one schema; the Schema element provides the default schema-wide settings. You can create an **«XSDschema»** package by dragging the **Schema** icon from the XML Schema Toolbox and dropping it directly onto a diagram.


#### Toolbox Icon



**Access** **Drag icon onto diagram**, or  
**Project Browser | Double-click on «XSDschema» stereotyped Package**, or  
**Diagram | Right-click on «XSDschema» stereotyped Package | Properties**

#### Define Properties


Field/Button	Action	See also										
Schema Name	If you do not want to use the default name of the schema package, overtype it with another name.											
Target Namespace	(Optional) Type in the target namespace for this Schema package.											
Prefix	(Optional) Type in the abbreviated text to represent the Target Namespace.											
Default Namespace	(Optional) Type in the default namespace for all non-prefixed XSDelements and XSDattributes.											
Schema File	Type in or browse for (click on ( ... )) the file path where the XML Schema file for this package is to be generated.											
XMLNS	<div>Identify the additional namespace or namespace-prefix pairs used in this Schema package.</div> <div>To <b>add</b> a namespace or namespace-prefix pair, click on the <b>New</b> button; to <b>edit</b> an existing entry, double-click on it. In either case, the Namespace Details dialog displays.</div> <table><tr><th>Field</th><th>Action</th></tr><tr><td>Prefix</td><td>Type in the abbreviated text to represent the Na</td></tr><tr><td>Namespace</td><td>Type in the name of the Namespace.</td></tr><tr><td>OK</td><td>Click on this button to <b>save</b> the new information Namespace Details dialog.</td></tr><tr><td>Cancel</td><td>Click on this button to <b>discard</b> the new informa Namespace Details dialog.</td></tr></table>	Field	Action	Prefix	Type in the abbreviated text to represent the Na	Namespace	Type in the name of the Namespace.	OK	Click on this button to <b>save</b> the new information Namespace Details dialog.	Cancel	Click on this button to <b>discard</b> the new informa Namespace Details dialog.	
Field	Action											
Prefix	Type in the abbreviated text to represent the Na											
Namespace	Type in the name of the Namespace.											
OK	Click on this button to <b>save</b> the new information Namespace Details dialog.											
Cancel	Click on this button to <b>discard</b> the new informa Namespace Details dialog.											

	<div> <div>Help</div> <div>Click on this button to display this Help topic.</div> </div>	
	To remove an entry from the list, click on it and click on the <b>Delete</b> button.	
OK	Click on this button to <b>save</b> the schema data entered and close the XSD schema Properties dialog.	
Cancel	Click on this button to <b>discard</b> the schema data entered and close the XSD schema Properties dialog.	
Help	Click on this button to display this Help topic.	
UML	<p>This button is displayed when you are <b>editing</b> existing Schema package information.</p> <p>Click on the button to open the UML element Properties dialog for the Schema element.</p>	<a href="#">Properties</a> 

#### Notes

- The default schema-wide settings are defined by Tagged Values, which you can review on the Tagged Values page of the schema element Properties dialog, or the Tagged Values window for the element; you can edit the schema-wide settings if you need to, or provide element-specific overrides in the properties and Tagged Values of the **individual XSD construct elements**

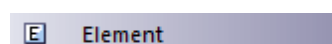
#### Learn more

- [Tagged Values](#)  (as you click on each Tagged Value, check the text field at the bottom of the Tagged Values window)

#### 14.1.1.3 Global Element

An «XSDtopLevelElement» stereotyped Class acts as a XSD global **element**. You can create it by dragging the **Element** icon from the XML Schema Toolbox and dropping it **directly** onto a diagram.

#### Toolbox Icon



**Access** Drag icon onto diagram, or Project Browser | Double-click on «XSDtopLevelElement» stereotyped Class, or Diagram | Right-click on «XSDtopLevelElement» stereotyped Class | Properties



**Define Properties**

Field/Button	Action	See also
<b>Name</b>	If you do not want to use the default name of the global element, overtype it with another name.	
<b>Type</b>	Either: <ul style="list-style-type: none"> <li>Type the name of a data type, or</li> <li>Click on the drop-down arrow and select an <b>XSD built-in dataType</b> from the list, or</li> <li>Click on the browse ( <b>...</b> ) button and browse for an existing <b>XSD classifier</b> element, or</li> <li>Select one of the two checkboxes below</li> </ul>	
<b>Nested complexType</b>	Select this checkbox to create an <b>XSDcomplexType</b> as a child of this global element.	<a href="#">Complex Type</a> <small>[2401]</small>
<b>Nested simpleType</b>	Select this checkbox to create an <b>XSDsimpleType</b> as a child of this global element.	<a href="#">Simple Type</a> <small>[2402]</small>
<b>Value</b>	(Optional) If you have entered an XSD built-in data type in the <b>Type</b> field, type in a value.	
<b>Default</b>	Select this radio button to set the <b>Value</b> as a default value.	
<b>Fixed</b>	Select this radio button to set the <b>Value</b> as a fixed value.	
<b>Annotation</b>	(Optional) Type in any notes you need for this element.	
<b>OK</b>	Click on this button to <b>save</b> the element data entered and close the XSD element Properties dialog.	
<b>Cancel</b>	Click on this button to <b>discard</b> the element data entered and close the XSD element Properties dialog.	
<b>Help</b>	Click on this button to display this Help topic.	
<b>UML</b>	This button is displayed when you are <b>editing</b> existing XSD element information.  Click on the button to open the UML element Properties dialog for the global element.	<a href="#">Properties</a> <small>[956]</small>

--	--	--

### Notes

- The fields **Type**, **Nested complexType** and **Nested simpleType** are mutually exclusive; selecting one disables the others
- The fields **Nested complexType** and **Nested simpleType** are available in the dialog only when **creating** a new global element (and not when **editing** the global element)
- A Global element
  - Cannot contain any UML **attributes**
  - Cannot be the **source** of an Association connector
  - **Can** be the **target** of an Association connector from a Complex Type Class or Group Class element
  - Cannot be the **target** of a Generalization connector
  - Can be the **source** of **one** Generalization connector to a Complex Type Class or Simple Type Class

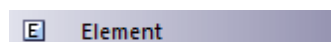
### Learn more

- [Local Element](#)<sup>2394</sup>

#### 14.1.1.4 Local Element

A **Local element** is an «*XSDelement*» stereotyped UML **attribute** that acts as a local XSD **element**. You can create it by dragging the **Element** icon from the XML Schema Toolbox and dropping it onto an «*XSDcomplexType*» or «*XSDgroup*» stereotyped **Class**.

### Toolbox Icon



**Access** **Drag icon onto «XSDcomplexType» or «XSDgroup» stereotyped Class**, or **Project Browser | Double-click on «XSDelement» stereotyped attribute**, or **Diagram | Highlight element | Highlight «XSDelement» stereotyped attribute | double-click on attribute**

### Define Properties

Field/Button	Action	See also
<b>Name</b>	If you do not want to use the default name of the local element, overwrite it with another name.	
<b>Type</b>	Either: <ul style="list-style-type: none"> <li>• Type the name of a data type, or</li> <li>• Click on the drop-down arrow and select an <b>XSD built-in dataType</b> from the list, or</li> </ul>	

	<ul style="list-style-type: none"> <li>Click on the browse ( <b>( ... )</b> ) button and browse for an existing <b>XSDcomplexType</b> or <b>XSDsimpleType</b> element as a classifier</li> </ul>	
<b>Reference</b>	(Optional) Specify whether to use the <i>ref</i> attribute (instead of the <i>type</i> attribute) to refer to the <b>XSDcomplexType</b> or <b>XSDsimpleType</b> element you selected in the <b>Type</b> field, in the generated XSD.	
<b>Value</b>	(Optional) If you have entered an XSD built-in data type in the <b>Type</b> field, type in a value.	
<b>Default</b>	Select this radio button to set the <b>Value</b> as a default value.	
<b>Fixed</b>	Select this radio button to set the <b>Value</b> as a fixed value.	
<b>MinOccurs</b>	(Optional) Type the minimum number of times this element must occur in the Class.  Type <b>0</b> to indicate that the element is optional.  The default value is <b>1</b> .	
<b>MaxOccurs</b>	(Optional) Type the maximum number of times this element can occur in the Class.  Type <b>unbounded</b> to indicate that there is no limit to the number of times the element can occur.  The default value is <b>1</b> .	
<b>Form</b>	(Optional) Click on the drop-down arrow and select whether or not to qualify the element: <ul style="list-style-type: none"> <li><b>qualified</b> - Use the <b>Prefix</b> defined in the Schema package to qualify this element</li> <li><b>unqualified</b> - Do not qualify this element</li> </ul>	<a href="#">Schema Package</a> <small>[2390]</small>
<b>Annotation</b>	(Optional) Type in any notes you need for this local element.	
<b>OK</b>	Click on this button to <b>save</b> the element data entered and close the XSD element Properties dialog.	
<b>Cancel</b>	Click on this button to <b>discard</b> the element data entered and close the XSD element Properties dialog.	
<b>Help</b>	Click on this button to display this Help topic.	

<b>UML</b>	<p>This button is displayed when you are <b>editing</b> existing XSD element information.</p> <p>Click on the button to open the UML attribute Properties dialog for the local element.</p>	<a href="#">General Properties of Attributes</a> <sup>[1001]</sup>
------------	---	--

### Notes

- Only «*Complex Type*», «*Group*» and «*Model Group*» stereotyped elements can have this UML Attribute

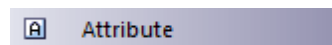
### Learn more

- [Complex Type](#) <sup>[2401]</sup>
- [Group](#) <sup>[2404]</sup>
- [Model Group](#) <sup>[2410]</sup>
- [Global Element](#) <sup>[2392]</sup>

#### 14.1.1.5 Global Attribute

A **Global Attribute** is an «*XSDtopLevelAttribute*» stereotyped Class. You can create it by dragging the **Attribute** icon from the XML Schema Toolbox and dropping it **directly** onto a diagram.




### Toolbox Icon



**Access** **Drag icon onto diagram**, or  
**Project Browser | Double-click on «XSDtopLevelAttribute» stereotyped element**, or  
**Diagram | right-click on «XSDtopLevelAttribute» stereotyped element | Properties**

### Define Properties

Field/Button	Action	See also
<b>Name</b>	If you do not want to use the default name of the global attribute, overwrite it with another name.	
<b>Type</b>	<p>Either:</p> <ul style="list-style-type: none"> <li>Type the name of a data type, or</li> <li>Click on the drop-down arrow and select an <b>XSD built-in dataType</b> from the list, or</li> <li>Click on the browse ( ( ... ) ) button and browse for an existing <b>XSDsimpleType</b> element as a classifier</li> </ul> <p>Alternatively, select the <b>Nested simpleType</b> checkbox (below).</p>	
<b>Nested</b>	Select this checkbox to create an <b>XSDsimpleType</b> element as a child of	<a href="#">Simple Type</a>

<b>simpleType</b>	this global attribute element.	
<b>Value</b>	(Optional) If you have selected an XSD built-in dataType in the <b>Type</b> field, type in a value.	
<b>Default</b>	Select this radio button to set the <b>Value</b> field as a <b>default</b> value.	
<b>Fixed</b>	Select this radio button to set the <b>Value</b> field as a <b>fixed</b> value.	
<b>Form</b>	(Optional) Click on the drop-down arrow and select: <ul style="list-style-type: none"> <li><b>qualified</b> to use any <b>Prefix</b> supplied on the Schema package to qualify this attribute, or</li> <li><b>unqualified</b> to show no qualifying prefix on the attribute name</li> </ul>	<a href="#">Schema Package</a> 
<b>Annotation</b>	(Optional) Type in any notes you need for this attribute.	
<b>OK</b>	Click on this button to <b>save</b> the attribute data entered and close the XSD attribute Properties dialog.	
<b>Cancel</b>	Click on this button to <b>discard</b> the attribute data entered and close the XSD attribute Properties dialog.	
<b>Help</b>	Click on this button to display this Help topic.	
<b>UML</b>	This button is displayed when you are <b>editing</b> existing XSD element information.  Click on the button to open the UML element Properties dialog for the global attribute Class.	<a href="#">Properties</a> 

### Notes

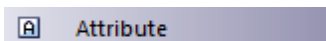
- The field **Nested simpleType** is available in the dialog only when creating a new global attribute (and not when editing the global attribute)
- The fields **Type** and **Nested simpleType** are mutually exclusive; selecting one disables the other
- A Global attribute:
  - Cannot contain any UML attributes
  - Cannot be the **source** of an Association connector
  - **Can** be the **target** of an Association connector from a Complex Type Class
  - Cannot be the **target** of a Generalization connector
  - **Can** be the **source** of one Generalization connector to a Simple Type Class

Learn more

- [Local Attribute](#) <sup>[2398]</sup>

**14.1.1.6 Local Attribute**

A **local attribute** is an «*XSDattribute*» stereotyped UML **attribute**. You can create it by dragging the **Attribute** icon from the XML Schema Toolbox and dropping it onto an «*XSDcomplexType*» or «*XSDattributeGroup*» stereotyped Class.

Toolbox Icon

Access **Drag icon onto «XSDcomplexType» or «XSDattributeGroup» stereotyped Class, or Project Browser | Double-click on «XSDattribute» stereotyped attribute, or Diagram | Highlight element | Highlight «XSDattribute» stereotyped attribute | double-click on attribute**

Define Properties

Field/Button	Action	See also
<b>Name</b>	If you do not want to use the default name of the local attribute, overtype it with another name.	
<b>Type</b>	Either: <ul style="list-style-type: none"> <li>• Type the name of a data type, or</li> <li>• Click on the drop-down arrow and select an <b>XSD built-in dataType</b> from the list, or</li> <li>• Click on the browse ( ( ... ) ) button and browse for an existing <b>XSDsimpleType</b> element as a classifier</li> </ul>	<a href="#">Simple Type</a> <sup>[2402]</sup>
<b>Reference</b>	(Optional) Specify whether to use the <i>ref</i> attribute (instead of the <i>type</i> attribute) to refer to the <b>XSDsimpleType</b> element you selected in the <b>Type</b> field, in the generated XSD.	
<b>Value</b>	(Optional) If you have entered an XSD built-in data type in the <b>Type</b> field, type in a value.	
<b>Default</b>	Select this radio button to set the <b>Value</b> as a <b>default</b> value.	
<b>Fixed</b>	Select this radio button to set the <b>Value</b> as a <b>fixed</b> value.	

<b>Form</b>	(Optional) Click on the drop-down arrow and select whether or not to qualify the attribute: <ul style="list-style-type: none"> <li>• <b>qualified</b> - Use the <b>Prefix</b> defined in the Schema package to qualify this attribute</li> <li>• <b>unqualified</b> - Do not qualify this attribute</li> </ul>	<a href="#">Schema Package</a> <small>[2390]</small>
<b>Annotation</b>	(Optional) Type in any notes you need for this local attribute.	
<b>OK</b>	Click on this button to <b>save</b> the attribute data entered and close the XSD attribute Properties dialog.	
<b>Cancel</b>	Click on this button to <b>discard</b> the element data entered and close the XSD attribute Properties dialog.	
<b>Help</b>	Click on this button to display this Help topic.	
<b>UML</b>	This button is displayed when you are <b>editing</b> existing XSD attribute information.  Click on the button to open the UML attribute Properties dialog for the local attribute.	<a href="#">General Properties of Attributes</a> <small>[1001]</small>

**Notes**

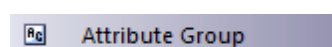
- Only **Complex Types** and **Attribute Groups** can have this UML attribute

**Learn more**

- [Complex Type](#) [2401]
- [Attribute Group](#) [2399]
- [Global Attribute](#) [2396]

**14.1.1.7 Attribute Group**

An **Attribute Group** Class is used to group a set of «*XSDattribute*» stereotyped UML attributes and Simple Type Classes that can be referenced from an «*XSDcomplexType*» stereotyped Class. You can create it by dragging the **Attribute Group** icon from the XML Schema Toolbox and dropping it **directly** onto a diagram.

**Toolbox Icon**

**Access**    **Drag icon onto diagram, or**  
**Project Browser | Double-click on «XSDattributeGroup» stereotyped element, or**  
**Diagram | right-click on «XSDattributeGroup» stereotyped element | Properties**

Define Properties

Field/Button	Action	See also
<b>Name</b>	If you do not want to use the default name of the Attribute Group, overwrite it with another name.	
<b>Annotation</b>	(Optional) Type in any notes you need for this Attribute Group.	
<b>OK</b>	Click on this button to <b>save</b> the attribute group data entered and close the XSD Attribute Group Properties dialog.	
<b>Cancel</b>	Click on this button to <b>discard</b> the attribute group data entered and close the XSD Attribute Group Properties dialog.	
<b>Help</b>	Click on this button to display this Help topic.	
<b>UML</b>	This button is displayed when you are <b>editing</b> existing XSD attribute group information.  Click on the button to open the UML element Properties dialog for the attribute group.	<a href="#">Properties Dialog</a> [956]

Notes

- An Attribute Group element:
  - Cannot be the **child** of any other XSD Class
  - Can contain **only** «XSDattribute» stereotyped UML attributes and Simple Type Classes
  - Can be the **source** of an Association connector to another Attribute Group
  - Can be the **target** of an Association connector from a Complex Type Class
  - Cannot be the source **or** target of a Generalization connector

Learn more

- [Complex Type](#) [2401]
- [Simple Type](#) [2402]
- [Local Attribute](#) [2398]



### 14.1.1.8 Complex Type

An «XSDcomplexType» stereotype is applied to a generic UML Class, to tailor the generation of a *complexType* definition in the Schema. You can create an «XSDcomplexType» stereotyped Class by dragging the **Complex Type** icon from the XML Schema Toolbox and dropping it **directly** onto a diagram.

#### Toolbox Icon



**Access**    **Drag icon onto diagram**, or  
**Project Browser | Double-click on «XSDcomplexType» stereotyped element**, or  
**Diagram | right-click on «XSDcomplexType» stereotyped element | Properties**

#### Define Properties

Field/Button	Action	See also
<b>Name</b>	If you do not want to use the default name of the complexType Class, overtype it with another name.	
<b>Model Group</b>	Click on the down-arrow and select the option that defines how the child elements of this complexType should occur in the Schema. <ul style="list-style-type: none"> <li>• <b>sequence</b> - the child elements must occur in the specified order</li> <li>• <b>choice</b> - only one of the child elements can occur</li> <li>• <b>all</b> - the child elements can occur in any order</li> </ul>	
<b>MinOccurs</b>	(Optional) Type the minimum number of times this element must occur in the Class.  Type <b>0</b> to indicate that the element is optional.  The default value is <b>1</b> .	
<b>MaxOccurs</b>	(Optional) Type the maximum number of times this element can occur in the Class.  Type <b>unbounded</b> to indicate that there is no limit to the number of times the element can occur.  The default value is <b>1</b> .	
<b>Annotation</b>	(Optional) Type any notes you need for this element.	
<b>Abstract</b>	(Optional) Select this checkbox to use this complexType in an <b>instance XML</b> file.	

<b>Mixed</b>	(Optional) Select this checkbox to allow character data to display among the child elements.	
<b>OK</b>	Click on this button to <b>save</b> the complexType data entered and close the XSD complexType Properties dialog.	
<b>Cancel</b>	Click on this button to <b>discard</b> the complexType data entered and close the XSD complexType Properties dialog.	
<b>Help</b>	Click on this button to display this Help topic.	
<b>UML</b>	<p>This button is displayed when you are <b>editing</b> existing XSD complexType information.</p> <p>Click on the button to open the UML element Properties dialog for the complexType Class.</p>	<a href="#">Properties</a> [956]

#### Notes

- A complexType can:
  - Contain both *XSDelement* and *XSDattribute* stereotyped UML attributes
  - Contain other complexTypes as child elements
  - Be a child of a Global Element
  - Be the **source** of Association connectors to other complexTypes, Simple Types, Attribute Groups, Groups and Model Groups
  - Be the **source** of a maximum of **one** Generalization connector to either another complexType or a Simple Type Class

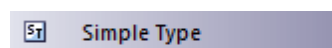
#### Learn more

- [Simple Type](#) [2402]
- [Attribute Group](#) [2399]
- [Group](#) [2404]
- [Model Group](#) [2410]

#### 14.1.1.9 Simple Type

An «*XSDsimpleType*» stereotype is applied to a generic UML Class, to tailor the generation of a simpleType definition in the Schema. You can create an «*XSDsimpleType*» Class by dragging the **Simple Type** icon from the XML Schema Toolbox and dropping it **directly** onto a diagram.

#### Toolbox Icon



**Access** **Drag icon onto diagram**, or  
**Project Browser | Double-click on «XSDsimpleType» stereotyped element**, or  
**Diagram | right-click on «XSDsimpleType» stereotyped element | Properties**

#### Define Properties

Field/Button	Action	See also
<b>Name</b>	If you do not want to use the default name of the simpleType element, overtype it with another name.	
<b>Type</b>	Either: <ul style="list-style-type: none"> <li>Type the name of a data type, or</li> <li>Click on the drop-down arrow and select an <b>XSD built-in dataType</b> from the list, or</li> <li>Click on the browse ( <b>( ...)</b> ) button and browse for an existing «XSDsimpleType» element as a classifier</li> </ul>	
<b>Restriction</b>	Select this radio button to restrict the value of this simpleType to that of the selected <b>Type</b> .  The various restrictions (facets) on the simpleType are available as Tagged Values on this Class.	<a href="#">Tagged Values</a> <sup>[1134]</sup>
<b>List</b>	Select this radio button to specify this simpleType as a list of values of the selected <b>Type</b> .	
<b>Annotation</b>	(Optional) Type any notes you need for this element.	
<b>OK</b>	Click on this button to <b>save</b> the simpleType data entered and close the XSD simpleType Properties dialog.	
<b>Cancel</b>	Click on this button to <b>discard</b> the simpleType data entered and close the XSD simpleType Properties dialog.	
<b>Help</b>	Click on this button to display this Help topic.	
<b>UML</b>	This button is displayed when you are <b>editing</b> existing XSD simpleType information.  Click on the button to open the UML element Properties dialog for the simpleType Class.	<a href="#">Properties</a> <sup>[956]</sup>

**Notes**

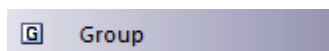
- A simpleType:
  - Cannot contain any «XSDelement» or «XSDattribute» stereotyped UML attributes
  - Cannot contain any child Classes
  - Cannot be the **source** of an Association connector
  - Can be the **target** of a Generalization connector
  - Can have at the most **one** Generalization connector to another simpleType Class

**Learn more**

- [Complex Type](#) 


**14.1.1.10 Group**

The **Group** Class is used to group a set of «XSDelement» stereotyped UML attributes, **Complex Type** Classes and **Simple Type** Classes that can be **referenced from** an «XSDcomplexType» Class. You can create this type of element by dragging the **Group** icon from the XML Schema Toolbox and dropping it **directly** onto a diagram.

**Toolbox Icon**

**Access** **Drag icon onto diagram**, or  
**Project Browser | Double-click on «XSDgroup» stereotyped element**, or  
**Diagram | right-click on «XSDgroup» stereotyped element | Properties**

**Define Properties**

Field/Button	Action	See also
<b>Name</b>	If you do not want to use the default name of the Group element, overwrite it with another name.	
<b>Model Group</b>	Click on the drop-down arrow and select the value that defines how the child elements of this group should occur in the Complex Type Class: <ul style="list-style-type: none"> <li>• <b>sequence</b> - to specify that the child elements must occur in the specified order</li> <li>• <b>choice</b> - to specify that only one of the child elements can occur</li> <li>• <b>all</b> - to specify that the child elements can occur in any order</li> </ul>	<a href="#">Complex Type</a> 
<b>Annotation</b>	(Optional) Type any notes you need for this element.	
<b>OK</b>	Click on this button to <b>save</b> the Group data entered and close the XSD	

	group Properties dialog.	
<b>Cancel</b>	Click on this button to <b>discard</b> the Group data entered and close the XSD group Properties dialog.	
<b>Help</b>	Click on this button to display this Help topic.	
<b>UML</b>	This button is displayed when you are <b>editing</b> existing XSD Group information.  Click on the button to open the UML element Properties dialog for the Group Class.	<a href="#">Properties</a> [956]

**Notes**

- A Group element can:
  - Contain only «*XSDelement*» stereotyped UML attributes
  - Contain Complex Types and Simple Types as child elements
  - Be the **source** of Association connectors to other Complex Types, Simple Types and Groups
  - Be the **target** of an Association connector from a Complex Type element
  - **Not** be the **source** or **target** of a Generalization connector

**Learn more**

- [Local Element](#) [2394]
- [Simple Type](#) [2402]

**14.1.1.11 Any**

An «*XSDany*» stereotyped **Class** allows a Complex Type Class to contain **elements** that are not specified in the Schema package. You can create it by dragging the **Any** icon from the XML Schema Toolbox and dropping it **directly** onto a diagram.

**Toolbox Icon**

**Access**    **Drag icon onto diagram**, or  
**Project Browser | Double-click on «XSDany» stereotyped element**, or  
**Diagram | right-click on «XSDany» stereotyped element | Properties**

**Define Properties**

Field/Button	Action	See also
--------------	--------	----------

<b>Name</b>	If you do not want to use the default name of the Any element, overtype it with another name.	
<b>Namespace</b>	(Optional) Type the namespace to contain the elements that can be used in the Complex Type.	<a href="#">Complex Type</a> <sup>[2401]</sup>
<b>ProcessContents</b>	(Optional) Click on the drop-down arrow and select the value that defines how the XML Parser should validate these elements: <ul style="list-style-type: none"> <li>• <b>lax</b> - to attempt to validate the elements against their Schema; no error is flagged when the Schema cannot be obtained</li> <li>• <b>skip</b> - to skip validating the elements</li> <li>• <b>strict</b> - to validate the elements against their Schema and flag an error if the Schema is unobtainable</li> </ul>	
<b>Annotation</b>	(Optional) Type any notes you need for this element.	
<b>OK</b>	Click on this button to <b>save</b> the Any element data entered and close the XSD any Properties dialog.	
<b>Cancel</b>	Click on this button to <b>discard</b> the Any element data entered and close the XSD group Properties dialog.	
<b>Help</b>	Click on this button to display this Help topic.	
<b>UML</b>	This button is displayed when you are <b>editing</b> existing «XSDany» element information.  Click on the button to open the UML element Properties dialog for the Any Class.	<a href="#">Properties</a> <sup>[956]</sup>

### Notes

- An Any Class:
  - Cannot **contain** any UML Attributes or child XSD Classes
  - Cannot be the **child** of any XSD Class
  - Cannot be the **target** of a Generalization connector
  - Cannot be the **source** of an Association or Generalization connector
  - Can be the **target** of Association connectors from Complex Types, Groups and Model Groups
  - **Must** be the **target** of at least **one** incoming Association connector

### Learn more

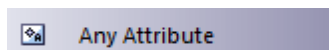
- [Schema Package](#) <sup>[2390]</sup>

- [Any Attribute](#) 

#### 14.1.1.12 Any Attribute



The «XSDany» stereotyped UML **attribute** allows a Complex Type element or an Attribute Group element to contain **attributes** that are not specified in the Schema package. You can create it by dragging the **Any Attribute** icon from the XML Schema Toolbox and dropping it onto an «XSDcomplexType» or «XSDataAttributeGroup» stereotyped Class.

##### Toolbox Icon



**Access** Drag icon onto «XSDcomplexType» or «XSDataAttributeGroup» stereotyped Class, or Project Browser | Double-click on «XSDany» stereotyped attribute, or Diagram | Highlight element | Highlight «XSDany» stereotyped attribute | double-click on attribute

##### Define Properties

Field/Button	Action	See also
<b>Name</b>	If you do not want to use the default name of the attribute, overwrite it with another name.	
<b>Namespace</b>	(Optional) Type the namespace to contain the attributes that can be used in the Complex Type or Attribute Group elements.	<a href="#">Complex Type</a>  <a href="#">Attribute Group</a> 
<b>ProcessContents</b>	(Optional) Click on the drop-down arrow and select the value that defines how the XML Parser should validate these attributes: <ul style="list-style-type: none"> <li>• <b>lax</b> - to attempt to validate the attributes against their Schema; no error is flagged when the Schema cannot be obtained</li> <li>• <b>skip</b> - to skip validating the attributes</li> <li>• <b>strict</b> - to validate the attributes against their Schema and flag an error if the Schema is unobtainable</li> </ul>	
<b>Annotation</b>	(Optional) Type any notes you need for this attribute.	
<b>OK</b>	Click on this button to <b>save</b> the attribute data entered and close the XSD anyAttribute Properties dialog.	
<b>Cancel</b>	Click on this button to <b>discard</b> the attribute data entered and close the XSD anyAttribute Properties dialog.	

<b>Help</b>	Click on this button to display this Help topic.	
<b>UML</b>	<p>This button is displayed when you are <b>editing</b> existing «XSDany» attribute information.</p> <p>Click on the button to open the UML attribute Properties dialog for the «XSDany» attribute.</p>	<a href="#">General Properties of Attributes</a> <sup>[1001]</sup>

### Notes

- Only Complex Type and Attribute Group elements can have this UML attribute

### Learn more

- [Any](#) <sup>[2405]</sup>

## 14.1.1.13 Union

A **Union** Class is a Simple Type element that defines a **collection** of Simple Types. You can create it by dragging the **Union** icon from the XML Schema Toolbox and dropping it **directly** on a diagram.

### Toolbox Icon




**Access** **Drag icon onto diagram, or Project Browser | Double-click on «XSDunion» stereotyped element, or Diagram | right-click on «XSDunion» stereotyped element | Properties**

### Define Properties

Field/Button	Action	See also				
<b>Name</b>	If you do not want to use the default name of the Union, overtype it with another name.					
<b>Member Types</b>	<div><div>Click on the <b>Browse</b> button (( ... )) to display the XSD Union Members dialog, and select built-in XSD datatypes and Simple Type elements to be members of the collection.</div><table><tr><th>Field/Button</th><th>Action</th></tr><tr><td><b>Class Name</b></td><td>Either:<ul style="list-style-type: none"><li>Type in an existing or proposed Class name</li></ul></td></tr></table></div>	Field/Button	Action	<b>Class Name</b>	Either: <ul style="list-style-type: none"><li>Type in an existing or proposed Class name</li></ul>	<a href="#">Simple Type</a> <sup>[2402]</sup>
Field/Button	Action					
<b>Class Name</b>	Either: <ul style="list-style-type: none"><li>Type in an existing or proposed Class name</li></ul>					



	Field/Button	Action
		<ul style="list-style-type: none"> <li>Click on the down arrow and select an XSD d</li> </ul> <p>Select the <b>Accept classifier even if not in model</b> checkbox and click the <b>Add</b> button.</p> <p>These options are generally used to specify objects to the same package as the Union element.</p>
	<b>Choose</b>	<p>Instead of typing or selecting values in the <b>Class Name</b> field, click the <b>Choose</b> button to display the Select Classifier browser and locate the Simple Type element.</p> <p>Click on the <b>OK</b> button to close the browser and immediately add the selected element to the <b>Type Details</b> list.</p> <p>This option is generally used to specify objects that are in the same package as the Union element, but you can select objects from a different package also.</p>
	<b>Add</b>	Click on this button to add the data type or element selected in the <b>Name</b> field to the <b>Type Details</b> list.
	<b>Accept classifier even if not in model</b>	Select this checkbox to include elements or data types that are named but that are not present in the same model package as the Union element.
	<b>Type Details</b>	Review the list of selected elements or data types. If you want to remove an object from the list, highlight it and click on the <b>Selected</b> button (below).
	<b>Delete Selected</b>	Click on this button to remove the currently-selected object from the <b>Type Details</b> list.
	<b>Close</b>	Click on this button to close the XSD Union Members dialog. This will remove the selected elements and data types in the <b>Member</b> list.
<b>Annotation</b>	(Optional) Type any notes you need for this element.	
<b>OK</b>	Click on this button to <b>save</b> the attribute data entered and close the XSD union Properties dialog.	
<b>Cancel</b>	Click on this button to <b>discard</b> the attribute data entered and close the XSD union Properties dialog.	

<b>Help</b>	Click on this button to display this Help topic.	
<b>UML</b>	<p>This button is displayed when you are <b>editing</b> existing «XSDunion» element information.</p> <p>Click on the button to open the UML element Properties dialog for the «XSDunion» element.</p>	<a href="#">Properties</a> 

### Notes

- When you click on the **Close** button on the XSD union Properties dialog, a Generalization connector is added to the diagram from the XSD Union element to each of the member elements on the same diagram; any elements **not** on the same diagram are listed in the top right corner of the XSD Union element
- If the Member Types that are not on the same diagram as the Union element are **not** listed, select **Tools | Options | Diagram | Behavior** and select the **Show 'Hidden' Parents** checkbox
- A Union element:
  - Cannot contain any child Classes
  - Cannot contain any «XSDelement» or «XSDattribute» stereotyped UML attributes
  - Cannot be the **source** of an Association connector
  - Can be the **target** of an Association connector from a Complex Type element
  - Can be the **target** of a Generalization connector from a Simple Type element

### Learn more

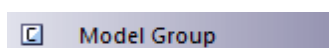
- [Complex Type](#) 

#### 14.1.1.14 Model Group

You can create an «XSDsequence», «XSDchoice» or «XSDall» stereotyped Class by dragging the **Model Group** icon from the XML Schema Toolbox and dropping it **directly** onto a diagram.

An «XSDsequence» model group (the default model group type) is a container for the attributes and associations owned by the Class. The model group is in turn added to the model groups of the Class's respective owners. Tagged Values specified by owners of the Class persist through to the child elements of the model group; if **memberNames** are unqualified for a **complexType**, so are the children of this model group when added to that complexType.

### Toolbox Icon



**Access** **Drag icon onto diagram**, or  
**Project Browser | Double-click on «XSDsequence», «XSDchoice» or «XSDall» stereotyped element**, or  
**Diagram | right-click on «XSDsequence», «XSDchoice» or «XSDall» stereotyped element | Properties**

**Define Properties**

Field/Button	Action	See also
<b>Name</b>	If you do not want to use the default name of the Model Group, overwrite it with another name.	
<b>Model Group</b>	<p>Click on the drop-down arrow and select the value that defines how the child elements of this group should occur in the Complex Type Class:</p> <ul style="list-style-type: none"> <li>• <b>sequence</b> - to specify that the child elements must occur in the specified order; creates an «<i>XSDsequence</i>» stereotyped Class</li> <li>• <b>choice</b> - to specify that only one of the child elements can occur; creates an «<i>XSDchoice</i>» stereotyped Class</li> <li>• <b>all</b> - to specify that the child elements can occur in any order; creates an «<i>XSDall</i>» stereotyped Class</li> </ul>	<a href="#">Complex Type</a> <small>[240]</small>
<b>MinOccurs</b>	<p>(Optional) Type the minimum number of times this element must occur in the Class.</p> <p>Type <b>0</b> to indicate that the element is optional.</p> <p>The default value is <b>1</b>.</p>	
<b>MaxOccurs</b>	<p>(Optional) Type the maximum number of times this element can occur in the Class.</p> <p>Type <b>unbounded</b> to indicate that there is no limit to the number of times the element can occur.</p> <p>The default value is <b>1</b>.</p>	
<b>Annotation</b>	(Optional) Type any notes you need for this element.	
<b>OK</b>	Click on this button to <b>save</b> the Model Group data entered and close the XSD element Properties dialog.	
<b>Cancel</b>	Click on this button to <b>discard</b> the Model Group data entered and close the XSD element Properties dialog.	
<b>Help</b>	Click on this button to display this Help topic.	
<b>UML</b>	<p>This button is displayed when you are <b>editing</b> existing Model Group element information.</p> <p>Click on the button to open the UML element Properties dialog for the Model Group Class.</p>	<a href="#">Properties</a> <small>[956]</small>

**Notes**

- A Model Group:
  - Can contain only «*XSDelement*» stereotyped UML attributes
  - Can contain Complex Types and Simple Types as child elements
  - Can be the **source** of Association connectors to Complex Type, Simple Type, Group and Model Group elements
  - Must be the **target** of least **one** incoming Association connector from a Complex Type
  - Cannot be the **source** or **target** of a Generalization connector

**Learn more**

- [Local Element](#)<sup>[2394]</sup>
- [Simple Type](#)<sup>[2402]</sup>
- [Group](#)<sup>[2404]</sup>

**14.1.1.15 Enumeration**


An **Enumeration** defines a list of acceptable values for the Class. You can create an Enumeration element by dragging the **Enum** icon from the XML Schema Toolbox and dropping it directly onto a diagram.

**Toolbox Icon**

**Access** **Drag icon onto diagram**, or  
**Project Browser | Double-click on Enumeration element**, or  
**Diagram | Right-click on element | Properties**

**Define Properties**



Field/ Button	Action	See also
<b>Name</b>	If you do not want to use the default name of the Enumeration, overwrite it with another name.	
<b>Type</b>	Either: <ul style="list-style-type: none"> <li>• Type the name of a data type, or</li> <li>• Click on the drop-down arrow and select an <b>XSD built-in dataType</b> from the list, or</li> <li>• Click on the browse ( ( ... ) ) button and browse for an existing <b>XSDsimpleType</b> element</li> </ul>	<a href="#">Simple Type</a> <sup>[2402]</sup>
<b>Values</b>	Type each of the values, separated by commas, for the selected <b>Type</b> .	

	These values are listed on the element as attributes.	
<b>Annotation</b>	(Optional) Type any notes you need for this element.	
<b>OK</b>	Click on this button to <b>save</b> the Enumeration element data entered and close the XSD enumeration Properties dialog.	
<b>Cancel</b>	Click on this button to <b>discard</b> the Enumeration element data entered and close the XSD enumeration Properties dialog.	
<b>Help</b>	Click on this button to display this Help topic.	
<b>UML</b>	<p>This button is displayed when you are <b>editing</b> existing Enumeration element information.</p> <p>Click on the button to open the UML element Properties dialog for the Enumeration Class.</p>	<a href="#">Properties</a> 

### Notes

- An Enumeration:
  - Cannot contain any «*XSDelement*» or «*XSDattribute*» stereotyped UML attributes
  - Cannot contain any child Classes
  - Cannot be the **source** of an Association connector
  - Can be the **target** of a Generalization connector
  - Can have at most **one** Generalization connector to a Simple Type Class

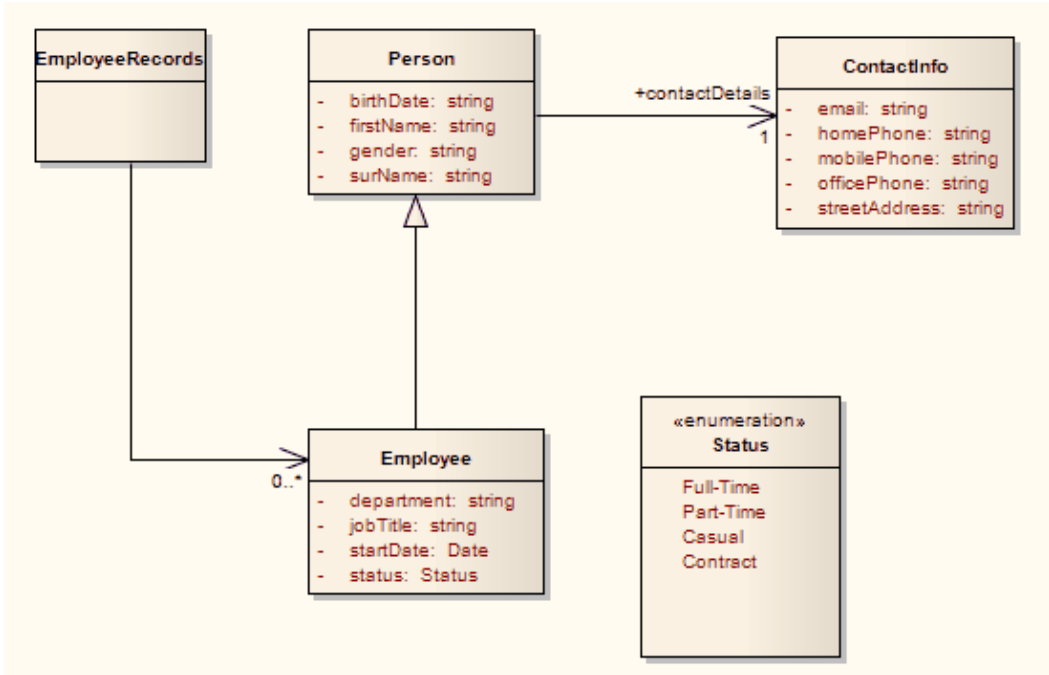
### Learn more

- [Local Element](#) 
- [Local Attribute](#) 

## 14.1.2 XML from Abstract Class Models

You can model XML schemas using only simple, abstract Class models. This makes it possible for an architect, for example, to start working at a higher level of abstraction without concern for the implementation details of a Schema. Whilst such an abstract model can subsequently be refined using the XML Schema pages of the Toolbox, it can be also be generated **directly** by Enterprise Architect's **Schema Generator**, in which case the Schema Generator applies a set of **default mappings** to convert the abstract model to an XSD file.

### Example

Structure	Detail
<b>Diagram</b>	<p>This is a simple Class element version of the earlier <i>Employee Details</i> example model. It does not use XSD-specific stereotypes or Tagged Values.</p>  <pre> classDiagram     class EmployeeRecords     class Person {         - birthDate: string         - firstName: string         - gender: string         - surName: string     }     class Employee {         - department: string         - jobTitle: string         - startDate: Date         - status: Status     }     class ContactInfo {         - email: string         - homePhone: string         - mobilePhone: string         - officePhone: string         - streetAddress: string     }     class Status {         «enumeration»         Full-Time         Part-Time         Casual         Contract     }     EmployeeRecords "0..*" --&gt; Employee     Employee &lt; -- Person     Person "1" --&gt; ContactInfo : +contactDetails   </pre>
<b>Schema</b>	<p>This schema fragment can be generated from the above model:</p> <pre> &lt;?xml version="1.0"?&gt; &lt;xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"&gt;   &lt;xs:simpleType name="Status"&gt;     &lt;xs:restriction base="xs:string"&gt;       &lt;xs:enumeration value="Full-Time"/&gt;       &lt;xs:enumeration value="Part-Time"/&gt;       &lt;xs:enumeration value="Casual"/&gt;       &lt;xs:enumeration value="Contract"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt;   &lt;xs:element name="Person" type="Person"/&gt;   &lt;xs:complexType name="Person"&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="firstName" type="xs:string"/&gt;       &lt;xs:element name="surName" type="xs:string"/&gt;       &lt;xs:element name="birthDate" type="xs:string"/&gt;       &lt;xs:element name="gender" type="xs:string"/&gt;       &lt;xs:element name="contactDetails" type="ContactInfo"/&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt;   &lt;xs:element name="Employee" type="Employee"/&gt;   &lt;xs:complexType name="Employee"&gt;     &lt;xs:complexContent&gt;       &lt;xs:extension base="Person"&gt;         &lt;xs:sequence&gt;           &lt;xs:element name="status" type="Status"/&gt;           &lt;xs:element name="jobTitle"   </pre>

Structure	Detail
	<pre> type="xs:string"/&gt;                                 &lt;xs:element name="startDate" type="xs:date"/&gt;                                 &lt;xs:element name="department" type="xs:string"/&gt;                                 &lt;/xs:sequence&gt;                                 &lt;/xs:extension&gt;                                 &lt;/xs:complexContent&gt; &lt;/xs:complexType&gt; &lt;xs:element name="EmployeeRecords" type="EmployeeRecords"/&gt; &lt;xs:complexType name="EmployeeRecords"&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="Employee" type="Employee" minOccurs="0" maxOccurs="unbounded"/&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; &lt;xs:element name="ContactInfo" type="ContactInfo"/&gt; &lt;xs:complexType name="ContactInfo"&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="homePhone" type="xs:string"/&gt;     &lt;xs:element name="mobilePhone" type="xs:string"/&gt;   &gt;     &lt;xs:element name="officePhone" type="xs:string"/&gt;   &gt;     &lt;xs:element name="email" type="xs:string"/&gt;     &lt;xs:element name="streetAddress" type="xs: string"/&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; &lt;/xs:schema&gt; </pre>

#### Learn more

- [Example XML Schema Diagram](#) <sup>[2389]</sup>
- [Default UML to XML Mappings](#) <sup>[2415]</sup>
- [Generate XSD](#) <sup>[2417]</sup>

#### 14.1.2.1 Default UML to XSD Mappings

When you are defining simple schemas using **abstract Class models**, the Enterprise Architect Schema Generator translates the UML information to XSD using a **default mapping of UML to XSD constructs**. These defaults are also used by the Schema Generator to generate **unstereotyped** elements in an abstract model.

When you model XML Schema using the XML Schema pages of the Diagram Toolbox, the stereotypes and Tagged Values of the Toolbox elements override the default mappings.

#### Constructs

UML Construct	Default XSD Production Rules
<b>Package</b>	A <b>Schema</b> element is generated for the target package. If the target package includes Classes from another package, which has the Tagged Values <b>targetNamespace</b> and <b>targetNamespacePrefix</b> set, these are included as attributes of the Schema element.

UML Construct	Default XSD Production Rules
	<p>In addition, an <b>import</b> or <b>include</b> element is created for each referenced package:</p> <ul style="list-style-type: none"> <li>An <b>include</b> element is used if the external package shares the same <b>targetNamespace</b> Tagged Value as the target package</li> <li>An <b>import</b> element is used where the <b>targetNamespaces</b> differ</li> </ul>
<b>Class</b>	A root-level element declaration and <b>complexType</b> definition are generated. The element name and type are the same as the Class name. An XSD <b>sequence Model Group</b> is also generated, to contain UML attributes generated as elements.
<b>Attribute</b>	<p>An <b>element</b> is declared for each Class attribute. The element name is set to that of the UML attribute name. This is prefixed with the <b>Class</b> name to make the element unique. The <b>minOccurs</b> and <b>maxOccurs</b> attributes are set to reflect the attribute cardinality.</p> <p>If the attribute refers to another Class, the element declaration is followed by a <b>complexType</b> definition, which contains a reference to the appropriate complexType.</p>
<b>Association</b>	An <b>element</b> is declared for each Association owned by a Class. The element name is set to that of the Association <b>role</b> . The <b>minOccurs</b> and <b>maxOccurs</b> attributes reflect the cardinality of the Association.
<b>Generalization (Inheritance)</b>	For single inheritances, an <b>extension element</b> is generated with the base attribute set to the base Class name. The UML attributes of the child Class are then appended to an <b>XSDall Model Group</b> within the extension element.
<b>Enumeration</b>	A <b>simpleType</b> element is declared for the Enumeration with the name attribute set to the Enumeration name. A Restriction element is generated with base set to <b>string</b> . Each of the Enumeration attributes is appended to the Restriction element as XSD Enumeration elements with value set to the UML attribute name. Any type specification for the UML attributes is ignored by the schema generator.

#### Notes

- If left unspecified, the **minOccurs** and **maxOccurs** attributes default to **1**
- If the direction of the Association is unspecified, the owner is assumed to be the source

#### Learning Center topics

- (Alt+F1) | [Enterprise Architect | SOA and XML Engineering | XSD | Order Elements](#)
- (Alt+F1) | [Enterprise Architect | SOA and XML Engineering | XSD | Refer External Schema](#)



### 14.1.3 Generate XSD

When you have developed your XML Schema model, either as an abstract Class model or a tailored XSD Class model, you can forward-engineer it into W3C XML Schema (XSD) files using the **Generate XML Schema** feature. As an XML Schema corresponds to a UML package in Enterprise Architect, XML Schema generation is a package-level operation.

You define the location of the file into which the XML Schema is to be generated, in the Schema Package element in your model.

**Access** [Project Browser](#) | [Click on source Package](#) | [Tools](#) | [XML Schema](#) | [Generate XML Schema](#), or [Project Browser](#) | [Right-click source Package](#) | [Code Engineering](#) | [Generate XML Schema](#)

#### Generate Schema files

Field/Option/Button	Action	See also
<b>Encoding</b>	Either: <ul style="list-style-type: none"> <li>Click on the drop-down arrow and select the XML encoding scheme to use, or</li> <li>Click on the <b>Default</b> button to apply the default encoding scheme (UTF-8)</li> </ul>	
<b>Generate global element for all global ComplexTypes ('Garden of Eden' style)</b>	Selected by default to generate Schema in the <b>Garden of Eden</b> style, containing a global element.  Clear the checkbox if you want to omit the global element.	<a href="#">Generate Global Element</a> <sup>[2418]</sup>
<b>Generate XSD for Referenced packages</b>	Select the checkbox to generate Schema for packages that are <b>referenced</b> by any of the packages selected on this dialog.	
<b>Prompt when missing Filename</b>	Select the checkbox to prompt, during Schema generation, for a filename for a referenced package if the path into which to generate the Schema file is missing.  This option is not available if the <b>Generate XSD for Referenced packages</b> option is not selected.	
<b>Use relative-path to reference XSDs (if 'schemaLocation' tag is empty)</b>	Select the checkbox to use a relative-path in the <i>XSD import</i> (or <i>XSD include</i> ) statement when referencing external packages, provided that the <i>schemaLocation</i> tag is empty on the referenced packages.  You set the <b>Schema File</b> field on the XSD Schema Properties dialog (the element Properties dialog for a Schema element) for the referenced and referencing <i>XSDschema</i> stereotyped packages, so that the relative-path is correctly determined.	<a href="#">Schema Package</a> <sup>[2390]</sup>

Field/Option/Button	Action	See also
<b>Generate XSD for Child packages</b>	<p>Select the checkbox to generate schema for child packages of the selected package, and then select either:</p> <ul style="list-style-type: none"> <li>• <b>Include all packages</b> - to list all child packages under the parent package in the list box, or</li> <li>• <b>Include &lt;XSDschema&gt; packages</b> - to list only those packages that have the stereotype «<i>XSDschema</i>»</li> </ul> <p>The list-box shows, for each package, the package name and the file path into which the Schema file can be generated.</p> <p>To change the file path for a package, double-click on the entry in the list-box, and type in or browse for the new file path in the prompt field.</p> <p>Select the checkbox against each package for which to generate Schema.</p>	
<b>Generate</b>	Click on this button to generate the Schema for each of the packages selected in the list-box.	
<b>Close</b>	Click on this button to close the dialog, without saving your option selections.	
<b>Help</b>	Click on this button to display this Help topic.	
<b>View Schema</b>	Click on this button to view the generated Schema for a package highlighted in the list-box.	
<b>Progress</b>	Check the progress of Schema generation.	

#### Learn more

- [Model XSD](#) <sup>[2387]</sup>
- [XML from Abstract Class Models](#) <sup>[2413]</sup>

#### Learning Center topics

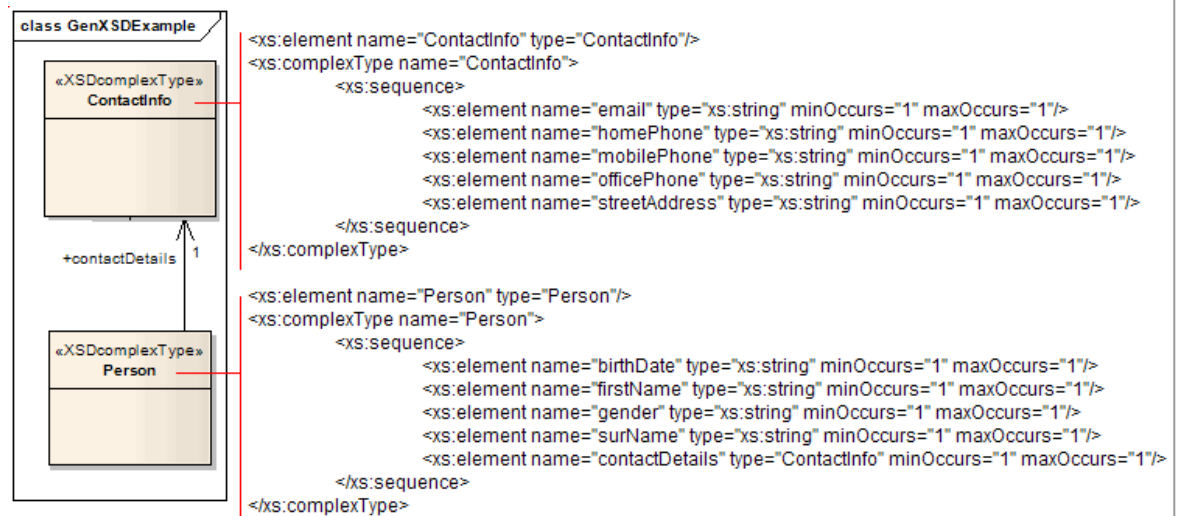
- (Alt+F1) | **Enterprise Architect | SOA and XML Engineering | XSD | Generate XML Schema**

### 14.1.3.1 Generate Global Element

Enterprise Architect, by default, generates XML Schema in the **Garden of Eden** style. For every global *XSDcomplexType* stereotyped Class, the system generates a global element.

#### Example

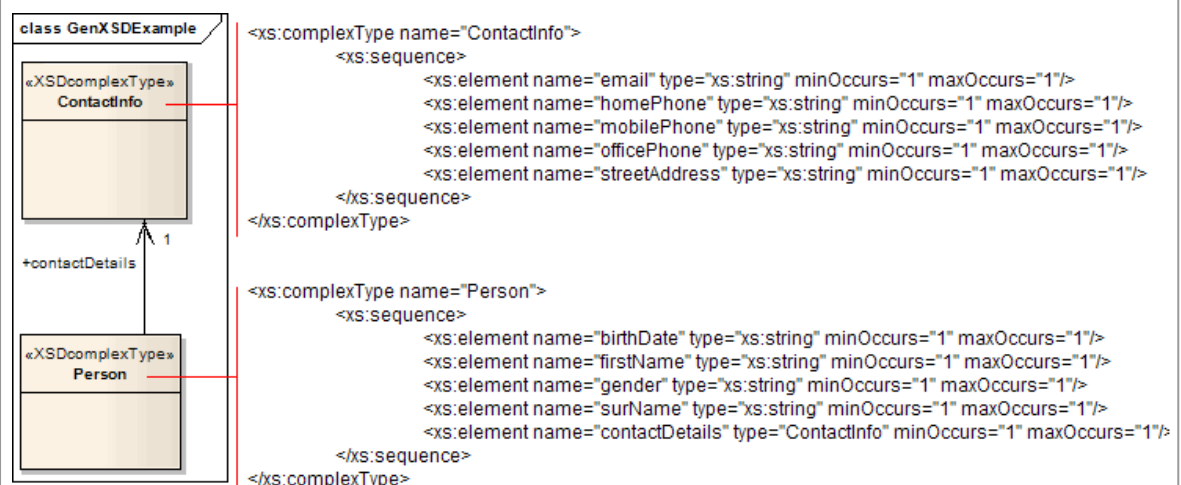
## Diagram



You can change the above specified default behaviour by de-selecting the **Generate global element for all global ComplexTypes** checkbox on the Generate XML Schema dialog. Then, the generated XSD no longer contains the global element; that is, it no longer has the lines:

- `<xs:element name="ContactInfo" type="ContactInfo"/>` and
- `<xs:element name="Person" type="Person"/>`

## Diagram



Learn more


- [Generate XSD](#) 

### 14.1.4 Import XSD

To reverse engineer a W3C XML Schema (XSD) file to create or overwrite a package of your UML Class model, you can use the **XML Schema Import** facility.

Access   **Project Browser | Click on target Package | Tools | XML Schema | Import XML Schema**, or  
**Project Browser | Right-click target Package | Code Engineering | Import XML Schema**

Import Schema files

Field/Option/ Button	Action	See also
<b>Root Package</b>	Displays the name of the selected package.	
<b>Directory</b>	Type in or browse for (click on ( ... )) the directory containing the source XSD file(s).	
<b>Selected File(s)</b>	Lists the XML Schema(s) currently available for import. <ul style="list-style-type: none"> <li>• To select a single file, click on it</li> <li>• To select several individual files ( <b>Ctrl</b> ) + <b>click</b> on each file</li> <li>• To select a range of files, press ( <b>Shift</b> ) and select the first and last file in the range</li> </ul>	
<b>Import global elements with "Type" postfix</b>	Select this checkbox to treat the global element and the <i>ComplexType</i> it is referring to as two separate entities.	<a href="#">Global Element and ComplexType</a> 
<b>Import referenced XML Schema(s)</b>	Select this checkbox to import any XML Schema that is being referenced by any of the files selected in the <b>Selected File(s)</b> field.	
<b>Create Diagram for XML Schema(s)</b>	Select this checkbox to create a <b>Class</b> diagram under each imported <i>XSDschema</i> package.	
<b>Import XSD Elements/ Attributes as</b>	Select the appropriate radio button to indicate how the inline <i>XSDelements</i> and <i>XSDattributes</i> are to be imported into a Class, either as: <ul style="list-style-type: none"> <li>• UML Associations or</li> <li>• UML attributes</li> </ul>	

<b>Import</b>	Click on this button to begin the XSD import.	
<b>Close</b>	Click on this button to close the dialog, without saving your option selections.	
<b>Help</b>	Click on this button to display this Help topic	
<b>Progress</b>	<p>Displays system messages indicating the progress of the Schema import.</p> <p>On imports containing a large number of external references, it can be useful to capture the progress messages to check exactly what has been imported. To do this, right-click on the messages and:</p> <ul style="list-style-type: none"> <li>• Copy the selected messages to the clipboard (select the <b>Copy Selected to Clipboard</b> menu option)</li> <li>• Copy all the messages to the clipboard (select the <b>Copy All to Clipboard</b> menu option), or</li> <li>• Save all the messages to a file (select the <b>Save to File</b> menu option)</li> </ul>	

### Notes

- If an XML Schema file being imported already exists in the model, Enterprise Architect skips importing the file
- References to XSD *Primitive Types* are always imported as UML attributes
- References to XSD constructs in external Schema files are always imported as UML attributes
- Enterprise Architect uses the *schemaLocation* attribute in the *XSD Import* and *XSD Include* elements of an XML Schema to determine the dependencies between the files; this attribute must be set to a valid file path (and not a URL) for the dependent XML Schema(s) to be imported correctly

### Learning Center topics

- (Alt+F1) | **Enterprise Architect | SOA and XML Engineering | XSD | Import XML Schema**

#### 14.1.4.1 Global Element and ComplexType

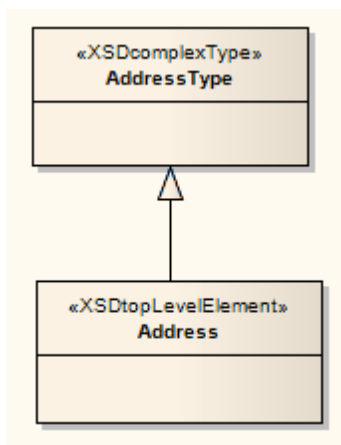
Some XML Schemas have **ComplexType** elements with the same name as the referring global elements, but with the suffix *Type*, as shown below:

```
<xs:element name="Address" type="AddressType" />
<xs:complexType name="AddressType">
  <xs:sequence/>
</xs:complexType>
```

On XSD import, by default, Enterprise Architect treats this global element and its bounding ComplexType as a single entity, and creates a single **XSDcomplexType** stereotyped Class with the same name as the global element, as shown below:



You can change this default behaviour by selecting the **Import global elements with "Type" postfix** checkbox on the Import XML Schema dialog. When you select this option, the system treats the global element and the ComplexType it is referring to as two separate entities. For the above example, the system creates an «XSDtopLevelElement» stereotyped Class for the global element and an «XSDcomplexType» stereotyped Class for the ComplexType, connected as shown:



### Notes

- Enterprise Architect treats the following as two separate entities irrespective of whether the **Import global elements with "Type" postfix** checkbox is selected or unselected:

```

<xs:element name="HomeAddress" type="AddressType" />
<xs:complexType name="AddressType">
  <xs:sequence />
</xs:complexType>
  
```

### Learn more

- [Import XML Schema](#)  [2420]

## 14.2 Web Services - WSDL

**Web Services Description Language 1.1 (WSDL)** is a key XML-based, World Wide Web Consortium (**W3C**) language for describing web services. WSDL support is critical for the development of a complete **Service Oriented Architecture (SOA)**, and the coupling of UML 2.4.1 and XML provides the natural mechanism for specifying, constructing and deploying XML-based SOA artifacts within an organization.

Using Enterprise Architect, you can rapidly model, forward engineer and reverse engineer WSDL files.

### Learn more

- [W3C WSDL Specification](#) (Online Resource)
- [WSDL 1.1 Model Structure](#) <sup>[2425]</sup>
- [Model WSDL](#) <sup>[2426]</sup>
- [WSDL Model Transformation](#) <sup>[2043]</sup>
- [Generate WSDL](#) <sup>[2445]</sup>
- [Import WSDL](#) <sup>[2447]</sup>

### Learning Center topics

- (Alt+F1) | **Enterprise Architect | SOA and XML Engineering | WSDL**

### 14.2.1 WSDL 1.1 Model Structure

A Web Service Description Language (WSDL), under specification 1.1, is defined within a «*WSDLnamespace*» stereotyped Package, which represents the top-level container for the WSDL elements. Conceptually it maps to the *targetNamespace* in a WSDL definition element.

When you create a WSDL model, Enterprise Architect creates the Namespace and provides a set of sub-packages, each containing a diagram on which to define the constituent elements of the model, with an Overview diagram to navigate between the sub-packages. You work through the sub-packages in sequence, to define the objects that are used by later objects, themselves called into still later objects.

### WSDL Structure Development

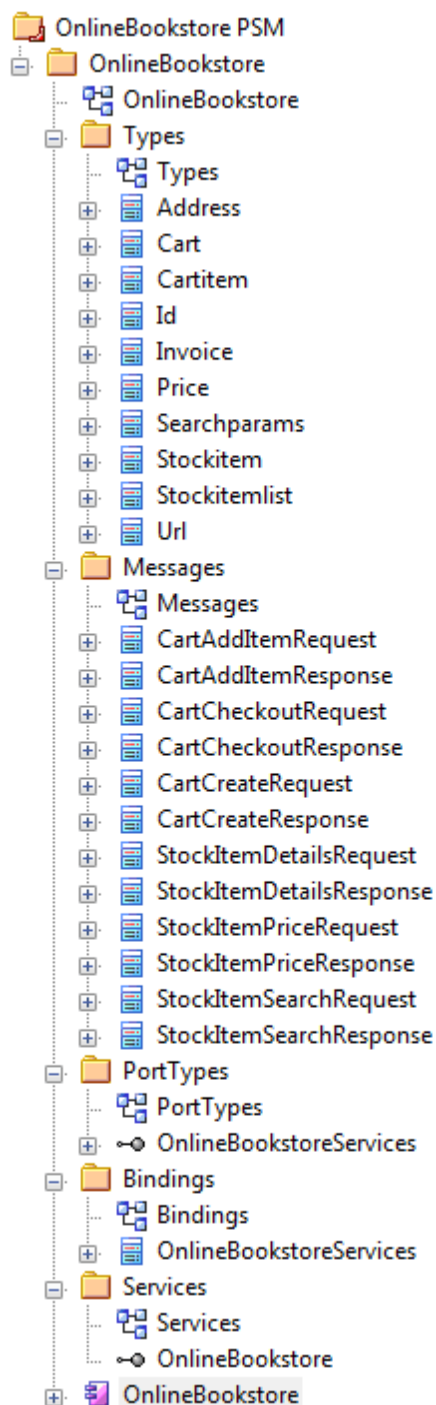
WSDL Element Type	Description	See also
<b>Types</b>	Defined in an <b>XSD Schema</b> , these are the <b>XSD data types</b> used by the web service and communicated by WSDL <b>Messages</b> ; you drag « <i>XSDElement</i> », « <i>XSDsimpleType</i> » and « <i>XSDcomplexType</i> » stereotyped elements onto the Types diagram from the XML Schema page of the Diagram Toolbox.	<a href="#">Model XSD</a> <sup>[2387]</sup>
<b>Messages</b>	WSDL <b>Messages</b> identify the data being communicated by a web service. Each Message element contains one or more <b>Message Parts</b> , which are attributes that each identify an XSD <b>data type</b> being communicated.	<a href="#">WSDL Message</a> <sup>[2430]</sup> <a href="#">WSDL Message Part</a> <sup>[2431]</sup>

<b>Port Types</b>	WSDL <b>Port Types</b> are the essential core of the web service, defining the <b>interfaces</b> of the service. Each Port Type consists of a set of <b>Port Type Operations</b> , each of which identifies an exchange of Messages (data input to and output from the interface as that operation). The Port Type Operation can also identify Messages acting as <b>Fault</b> indicators.	<a href="#">WSDL Port Type</a> <sup>[2433]</sup> <a href="#">WSDL Port Type Operation</a> <sup>[2434]</sup>
<b>Bindings</b>	A <b>Binding</b> specifies the protocol and data format for the operations and messages defined for of a particular <b>Port Type</b> . Each « <i>WSDLbinding</i> » Class implements (realizes) the operations specified by the « <i>WSDLportType</i> » Interface - the <b>Port Type Operations</b> in the Port Type element are automatically copied into the Binding element as <b>Binding operations</b> .	<a href="#">WSDL Binding</a> <sup>[2437]</sup> <a href="#">WSDL Binding Operation</a> <sup>[2438]</sup>
<b>Services</b>	A WSDL <b>Service</b> defines a formal interface of the web service. It describes the collection of <b>Port Types</b> that expose a particular <b>Binding</b> , having an Association to each exposed Binding. It therefore encapsulates a set of the other data structures - if not all the data structures - defined in the model.	<a href="#">WSDL Service</a> <sup>[2441]</sup>
<b>Documents</b>	WSDL Documents are represented by <b>Components</b> having the stereotype « <i>WSDL</i> ». This is the element from which you <b>generate</b> the <b>WSDL file</b> .  You can create more than one Document to re-use the schema Types, Messages, Port Types, Bindings and Services of a Namespace across multiple physical WSDL documents, either in the same configuration or in different configurations.	<a href="#">WSDL Document</a> <sup>[2443]</sup>

### Example

This figure shows an example WSDL namespace, *OnlineBookstore PSM*, which includes a single WSDL document, *OnlineBookstore* (at the bottom of the hierarchy).





### Notes

- You can also generate a WSDL Package structure from a UML Interface using the **WSDL Model Transformation**

### Learn more

- [Web Services - WSDL](#) <sup>[2423]</sup>
- [Model WSDL](#) <sup>[2426]</sup>

- [WSDL Model Transformation](#) <sup>[2043]</sup>
- [Generate WSDL](#) <sup>[2445]</sup>
- [Import WSDL](#) <sup>[2447]</sup>

#### Learning Center topics












- (Alt+F1) | **Enterprise Architect** | **SOA and XML Engineering** | **WSDL** | **Getting Started**

## 14.2.2 Model WSDL

You can quickly and easily model the elements in a Web Service Definition using the **WSDL page** of the Diagram Toolbox. As a first step, you can create an example **WSDL Package** structure in the Project Browser, using the **Namespace** icon from the WSDL page. You can use this example package structure as a template for developing your WSDL.

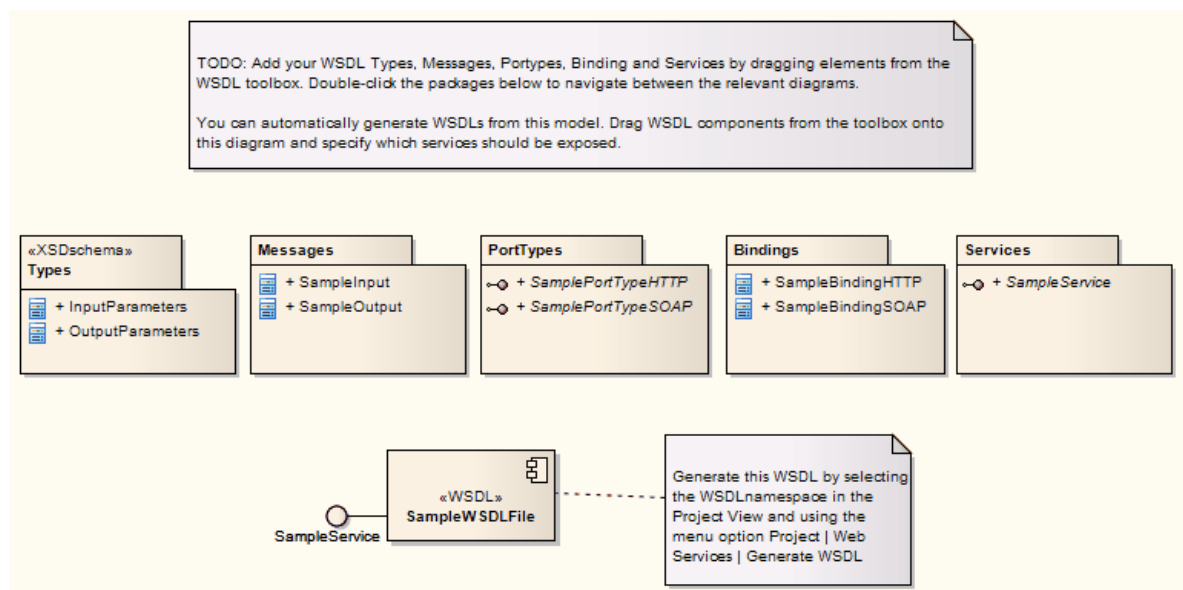
#### Create a new WSDL Package structure

Step	Action	See also
1	In the Project Browser, create the top-level project structure you need (Model and Views), and click on the appropriate View.	<a href="#">Models</a> <sup>[753]</sup> <a href="#">Views</a> <sup>[769]</sup>
2	Click on the <b>New Package</b> icon in the Project Browser toolbar. The New Model Package dialog displays.	<a href="#">Project Browser Toolbar</a> <sup>[669]</sup>
3	In the <b>Name</b> field type the name of the new package, and select the <b>Automatically add new diagram</b> and <b>Open new diagram</b> checkboxes.	<a href="#">Add a Package</a> <sup>[772]</sup>
4	Click on the <b>OK</b> button. The New Diagram dialog displays.	
5	In the <b>Name</b> field type the name of the new diagram. In the Select From panel select <b>UML Structural</b> , and in the Diagram Types panel select <b>Class</b> .	<a href="#">Add New Diagrams</a> <sup>[822]</sup>
6	Click on the <b>OK</b> button. The new Class diagram opens in the Diagram View, with the Class pages displaying in the Diagram Toolbox.	<a href="#">Diagram Toolbox</a> <sup>[792]</sup>
7	Click on the <b>More tools</b> option in the Toolbox and select <b>Extended   WSDL</b> . The WSDL Toolbox page displays.	<a href="#">WSDL Toolbox</a> <sup>[817]</sup>

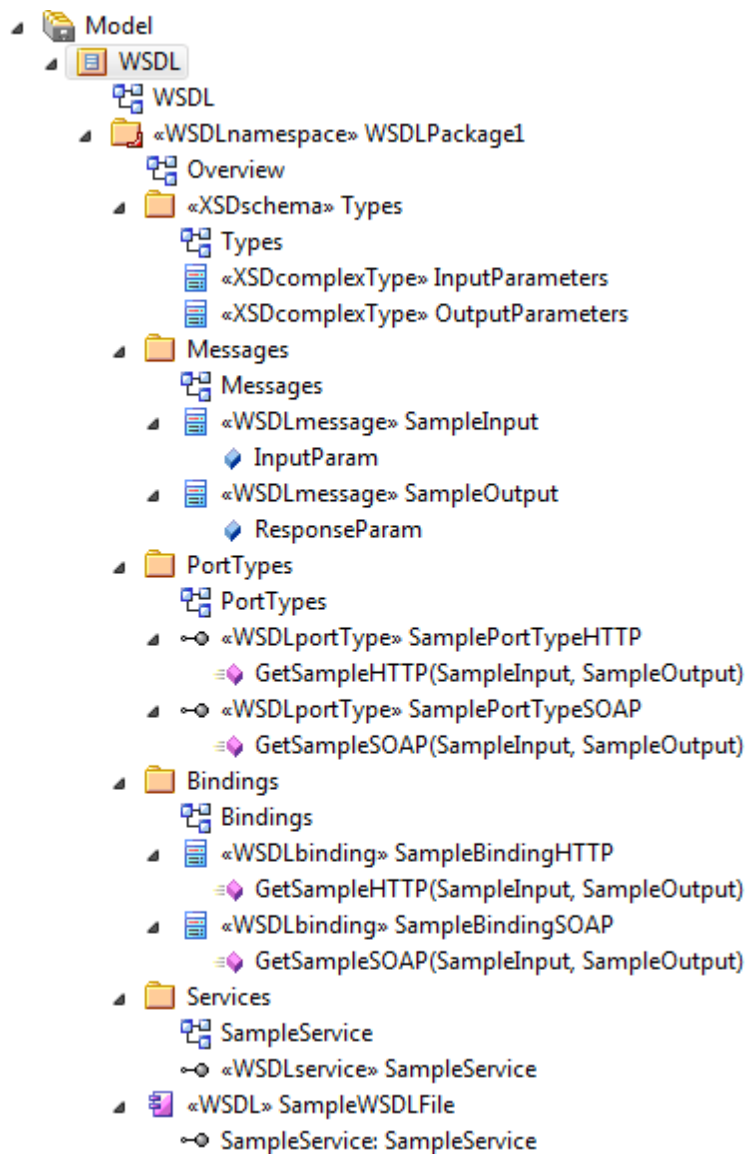
8	<p>Click on the <b>Namespace</b> icon from the Toolbox and drag it into the Class diagram.</p> <p>The WSDL Namespace Properties dialog displays.</p> <p>Type in a <b>WSDL Package Name</b> and the URL of the <b>Target Namespace</b>. You can edit these values later.</p>	<a href="#">WSDL Namespace</a>  [2428]
9	<p>Click on the <b>OK</b> button.</p> <p>The sample «<i>WSDLnamespace</i>» stereotyped package structure is created on the diagram, and the full model structure is displayed, expanded, in the Project Browser (see below).</p> <p>The model structure consists of the following sub-packages, with an Overview diagram to navigate between the sub-packages:</p> <ul style="list-style-type: none"> <li>• <b>Types</b>: Contains the XSD types for the data communicated by the web service, on a <b>Types</b> diagram</li> <li>• <b>Messages</b>: Contains the WSDL <b>Messages</b>, modeled as UML Classes marked with the stereotype «<i>WSDLmessage</i>»</li> <li>• <b>PortTypes</b>: Contains the WSDL <b>Port Types</b>, modeled as UML interfaces marked with the stereotype «<i>WSDLportType</i>»</li> <li>• <b>Bindings</b>: Contains the WSDL <b>Bindings</b>, modeled as UML Classes that realize the PortTypes</li> <li>• <b>Services</b>: Contains the WSDL <b>Services</b>, modeled as UML interfaces with Associations to each exposed Binding</li> </ul>	<a href="#">WSDL 1.1 Model Structure</a>  [2423] <a href="#">Model XSD</a>  [2387]
10	<p>Model each of the WSDL constructs in their corresponding Packages.</p>	<a href="#">WSDL Message</a>  [2430] <a href="#">WSDL Message Part</a>  [2431] <a href="#">WSDL Port Type</a>  [2433] <a href="#">WSDL Port Type Operation</a>  [2434] <a href="#">WSDL Binding</a>  [2437] <a href="#">WSDL Binding Operation</a>  [2438] <a href="#">WSDL Service</a>  [2441] <a href="#">WSDL Document</a>  [2443]

### Template WSDL Model - Diagram

The *WSDLnamespace* Package acts as a container for the WSDL structure.



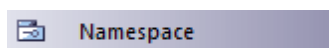
### Template WSDL Model - Project Browser Hierarchy



### 14.2.2.1 WSDL Namespace

A «WSDLnamespace» stereotyped Package represents the top-level container for the WSDL constructs in Enterprise Architect. You can create the Namespace package by dragging the **Namespace** icon from the WSDL Toolbox page and dropping it directly onto a diagram.

#### Toolbox Icon



**Access**    **Drag icon onto diagram** or  
**Project Browser | Double-click on «WSDLnamespace» stereotyped Package element**

#### Define Properties

Field/Option/ Button	Action	See also
<b>WSDL Package Name</b>	Type in the name of the WSDL Namespace Package element.	
<b>Target Namespace</b>	(Optional) Type in the URL for the WSDL Namespace package.	
<b>OK</b>	Click on this button to save the values entered and close the WSDL Namespace Properties dialog.  If you have just created the Namespace, a new Package diagram opens containing the sample template WSDL model.	<a href="#">Model WSDL</a> <sup>[2426]</sup>
<b>Cancel</b>	Click on this button to <b>discard</b> the data entered and close the WSDL Namespace Properties dialog.	
<b>Help</b>	Click on this button to display this Help topic.	
<b>UML</b>	This button is displayed when you are <b>editing</b> existing WSDL Namespace element information.  Click on the button to open the UML element Properties dialog for the Namespace Package element.	<a href="#">Properties</a> <sup>[956]</sup>

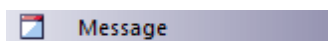
#### Learn more

- [Model WSDL](#) <sup>[2426]</sup>
- [WSDL 1.1 Model Structure](#) <sup>[2423]</sup>

### 14.2.2.2 WSDL Message

A «WSDLmessage» stereotyped Class represents a WSDL **Message** and acts as a container for one or more WSDL **Message Parts**. You can create WSDL Messages by dragging the **Message** icon from the WSDL Toolbox and dropping it directly onto the Messages diagram (under the **Messages package** in the WSDL Package structure).

#### Toolbox Icon



**Access** Drag icon onto diagram, or  
Project Browser | Double-click on «WSDLmessage» stereotyped element, or  
Diagram | right-click on «WSDLmessage» stereotyped element | Properties

#### Define Properties

Field/Option/ Button	Action	See also
<b>Name</b>	Type in the name of the WSDL Message.	
<b>Documentation</b>	(Optional) Type in any notes you need for this element.	
<b>OK</b>	Click on this button to <b>save</b> the data entered and close the WSDL Message dialog.	
<b>Cancel</b>	Click on this button to <b>discard</b> the data entered and close the WSDL Message dialog.	
<b>Help</b>	Click on this button to display this Help topic.	
<b>UML</b>	This button is displayed when you are <b>editing</b> existing WSDL Message element information.  Click on the button to open the UML Class Properties dialog for the element.	<a href="#">Properties</a> 956

### Notes

- WSDL Messages can only be created under the *Messages* package in the WSDL Package structure
- The name of the WSDL Message should be unique amongst all WSDL Messages within the WSDL

### Learn more

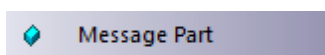
- [Message Part](#) <sup>[2431]</sup>
- [WSDL 1.1 Model Structure](#) <sup>[2423]</sup>
- [Model WSDL](#) <sup>[2426]</sup>

#### 14.2.2.2.1 WSDL Message Part

A **WSDL Message Part** is the segment of a **WSDL Message** that identifies the XSD data type of the data communicated by the Message. If a Message communicates data of more than one data type, each data type is identified in a **separate** Message Part.

In Enterprise Architect, a WSDL Message Part is represented by a UML **attribute** of the WSDL Message Class. You can create the Message Part attribute by dragging the **Message Part** icon from the WSDL Toolbox and dropping it onto a «WSDLmessage» stereotyped Class.

### Toolbox Icon



**Access** **Drag icon onto «WSDLmessage» stereotyped element**, or  
**Project Browser | Double-click on Message Part attribute**, or  
**Diagram | Highlight Message element | Highlight Message Part attribute | double-click on attribute**

#### Define Properties

Field/Option/ Button	Action	See also
<b>Name</b>	Type in the name of the WSDL Message Part attribute.	
<b>Type</b>	<p>Either:</p> <ul style="list-style-type: none"> <li>Type the name of a data type, or</li> <li>Click on the drop-down arrow and select an <b>XSD built-in dataType</b> from the list, or</li> <li>Click on the browse ( ( ... ) ) button and browse for an existing «<i>XSDelement</i>», «<i>XSDcomplexType</i>» or «<i>XSDsimpleType</i>» element as a classifier</li> </ul> <p>The XSD Types can be defined in:</p> <ul style="list-style-type: none"> <li>The <b>Types</b> package under the WSDL Package Structure or</li> <li>Any other package in the model</li> </ul>	<p><a href="#">XML Schema</a> <small>[2387]</small></p> <p><a href="#">WSDL Package Structure</a> <small>[2429]</small></p>
<b>OK</b>	Click on this button to <b>save</b> the data entered and close the WSDL Message Part dialog.	
<b>Cancel</b>	Click on this button to <b>discard</b> the data entered and close the WSDL Message Part dialog.	
<b>Help</b>	Click on this button to display this Help topic.	
<b>UML</b>	<p>This button is displayed when you are <b>editing</b> existing WSDL Message Part attribute information.</p> <p>Click on the button to open the UML Attribute Properties dialog for the Message Part.</p>	<a href="#">General Properties of Attributes</a> <small>[1001]</small>

#### Notes

- WSDLmessage* stereotyped Classes can effectively contain Message Part attributes only; if you add other attributes to the Class element, they are re-cast as Message Parts

#### Learn more



- [WSDL Message](#) <sup>[2430]</sup>
- [WSDL 1.1 Model Structure](#) <sup>[2423]</sup>
- [Model WSDL](#) <sup>[2426]</sup>

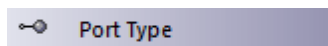
#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | SOA and XML Engineering | WSDL | Parts Referring External XSD**
- (Alt+F1) | **Enterprise Architect | SOA and XML Engineering | WSDL | Types Referring External XSD**
- (Alt+F1) | **Enterprise Architect | SOA and XML Engineering | WSDL | Message-Typing Attribute**

### 14.2.2.3 WSDL Port Type

A «WSDLportType» stereotyped **Interface** represents a WSDL PortType. It describes the operations exposed by the WSDL, acting as a container for one or more WSDL **Port Type Operations**. You can create a WSDL PortType element by dragging the **Port Type** icon from the WSDL Toolbox and dropping it directly onto the **PortTypes** diagram (under the *PortTypes* package in the WSDL Package structure).

#### Toolbox Icon



**Access** **Drag icon onto diagram**, or  
**Project Browser | Double-click on «WSDLportType» stereotyped element**, or  
**Diagram | right-click on «WSDLportType» stereotyped element | Properties**

#### Define Properties




Field/Option/ Button	Action	See also
<b>Name</b>	Type in the name of the WSDL PortType.	
<b>Documentation</b>	(Optional) Type in any notes you need for this element.	
<b>OK</b>	Click on this button to <b>save</b> the data entered and close the WSDL PortType dialog.	
<b>Cancel</b>	Click on this button to <b>discard</b> the data entered and close the WSDL PortType dialog.	
<b>Help</b>	Click on this button to display this Help topic.	
<b>UML</b>	This button is displayed when you are <b>editing</b> existing WSDL PortType element information.  Click on the button to open the UML element Properties dialog for the	<a href="#">Properties</a> <sup>[956]</sup>

	element.	
--	----------	--

### Notes

- WSDL PortTypes can only be created under the *PortTypes* package in the WSDL Package structure
- The name of the WSDL PortType should be unique amongst all the WSDL PortTypes within the WSDL

### Learn more

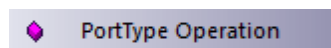
- [WSDL PortType Operation](#) 
- [WSDL 1.1 Model Structure](#) 
- [Model WSDL](#) 

#### 14.2.2.3.1 WSDL Port Type Operation

A **Port Type Operation** identifies an exchange of Messages (data input to and output from the interface as an operation). The Port Type Operation can also identify Messages acting as Fault indicators.

In Enterprise Architect, a WSDL PortType Operation is represented by a UML **Operation** of the **WSDL PortType Interface**. You can create a PortType Operation by dragging the **PortType Operation** icon from the WSDL Toolbox and dropping it onto a «WSDLportType» stereotyped Interface.

### Toolbox Icon




**Access** Drag icon onto «WSDLportType» stereotyped element, or  
Project Browser | Double-click on PortType operation, or  
Diagram | Highlight PortType element | Highlight PortType operation | double-click on operation

### Define Properties

Field/Option/ Button	Action	See also
<b>Name</b>	Type in the name of the WSDL PortType Operation.	
<b>Documentation</b>	(Optional) Type in any notes you need for this operation.	
<b>Operation Type</b>	Click on the drop-down arrow and select one of the supported PortType Operation types: <ul style="list-style-type: none"> <li>• <b>OneWay</b></li> <li>• <b>Request-Response</b></li> <li>• <b>Solicit-Response</b></li> <li>• <b>Notification</b></li> </ul>	

<b>Input</b>	<p>This section is grayed out if you have selected <b>Notification</b> as the operation type.</p> <table><tr><th>Field/Option</th><th>Action</th></tr><tr><td><b>Name</b></td><td>Defaults to a name that parallels the <b>Operation Type</b>. If you do not want to use the default, type an alternative input Message.</td></tr><tr><td><b>Message</b></td><td>Click on the drop-down arrow and select one of the W previously created in the <i>Message</i> package.</td></tr><tr><td><b>Documentation</b></td><td>(Optional) Type in any notes you need for this input Me</td></tr></table>	Field/Option	Action	<b>Name</b>	Defaults to a name that parallels the <b>Operation Type</b> . If you do not want to use the default, type an alternative input Message.	<b>Message</b>	Click on the drop-down arrow and select one of the W previously created in the <i>Message</i> package.	<b>Documentation</b>	(Optional) Type in any notes you need for this input Me	<a href="#">Message</a> <sup>[2430]</sup>
Field/Option	Action									
<b>Name</b>	Defaults to a name that parallels the <b>Operation Type</b> . If you do not want to use the default, type an alternative input Message.									
<b>Message</b>	Click on the drop-down arrow and select one of the W previously created in the <i>Message</i> package.									
<b>Documentation</b>	(Optional) Type in any notes you need for this input Me									
<b>Output</b>	<p>This section is grayed out if you have selected <b>OneWay</b> as the operation type.</p> <table><tr><th>Field/Option</th><th>Action</th></tr><tr><td><b>Name</b></td><td>Defaults to a name that parallels the <b>Operation Type</b>. If you do not want to use the default, type an alternative output Message.</td></tr><tr><td><b>Message</b></td><td>Click on the drop-down arrow and select one of the W previously created in the <i>Message</i> package.</td></tr><tr><td><b>Documentation</b></td><td>(Optional) Type in any notes you need for this output M</td></tr></table>	Field/Option	Action	<b>Name</b>	Defaults to a name that parallels the <b>Operation Type</b> . If you do not want to use the default, type an alternative output Message.	<b>Message</b>	Click on the drop-down arrow and select one of the W previously created in the <i>Message</i> package.	<b>Documentation</b>	(Optional) Type in any notes you need for this output M	
Field/Option	Action									
<b>Name</b>	Defaults to a name that parallels the <b>Operation Type</b> . If you do not want to use the default, type an alternative output Message.									
<b>Message</b>	Click on the drop-down arrow and select one of the W previously created in the <i>Message</i> package.									
<b>Documentation</b>	(Optional) Type in any notes you need for this output M									
<b>Faults</b>	<p>Review the details of the WSDL Messages that can act as Faults.</p> <p>Faults display in this list with the most recently-created at the top and the oldest at the end. If more than four Fault Messages are defined, use the vertical scroll bar to display the rest of the list.</p> <p>To add a Message, click on the <b>New</b> button. The WSDL PortType Operation Fault dialog displays.</p> <table><tr><th>Field/Option/Button</th><th>Action</th></tr><tr><td><b>Name</b></td><td>Defaults to <b>Fault&lt;n&gt;</b>.</td></tr></table>	Field/Option/Button	Action	<b>Name</b>	Defaults to <b>Fault&lt;n&gt;</b> .					
Field/Option/Button	Action									
<b>Name</b>	Defaults to <b>Fault&lt;n&gt;</b> .									

		If you do not want to use the default, type an alternative fault Message.	
	<b>Message</b>	Click on the drop-down arrow and select one of the W previously created in the <i>Message</i> package.	
	<b>Documentation</b>	(Optional) Type in any notes you need for this fault Message.	
	<b>OK</b>	Click on this button to <b>save</b> the data entered and close the PortType Operation Fault dialog.	
	<b>Cancel</b>	Click on this button to <b>discard</b> the data entered and close the PortType Operation Fault dialog.	
	<b>Help</b>	Click on this button to display this Help topic.	
	To remove a Message from the list, click on it and click on the <b>Delete</b> button.		
<b>OK</b>	Click on this button to <b>save</b> the data entered and close the WSDL PortType Operation dialog.		
<b>Cancel</b>	Click on this button to <b>discard</b> the data entered and close the WSDL PortType Operation dialog.		
<b>Help</b>	Click on this button to display this Help topic.		
<b>UML</b>	<p>This button is displayed when you are <b>editing</b> existing WSDL Port Type Operation information.</p> <p>Click on the button to open the UML operation Properties dialog for the element.</p>		<a href="#">General Properties of Operations</a> 

### Notes

- WSDL PortType Operations can only be contained by WSDL PortTypes
- The name provided for an **Input**, **Output** or **Fault** Message in a PortType Operation must be unique amongst the Input, Output and Fault Messages, respectively, across the WSDL **PortType**
- In the UML operation Properties dialog, the Messages identified as Input, Output and Fault can be examined as the parameters of the operation

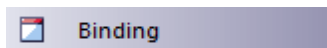
Learn more

- [WSDL Port Type](#) <sup>[2433]</sup>
- [WSDL 1.1 Model Structure](#) <sup>[2423]</sup>
- [Model WSDL](#) <sup>[2426]</sup>

**14.2.2.4 WSDL Binding**

A **WSDLbinding** element implements the operations specified by a particular «*WSDLportType*» stereotyped Interface and describes the message format and protocol details for the operations and messages defined by this WSDL PortType. A WSDL Binding element is represented by a «*WSDLbinding*» stereotyped Class.

You create a WSDL Binding element by dragging the **Binding** icon from the WSDL Toolbox directly onto the Bindings diagram under the Bindings package in the WSDL Package structure.

Toolbox Icon

Access    **Drag icon onto diagram**, or  
**Project Browser | Double-click on «WSDLbinding» stereotyped element**, or  
**Diagram | right-click on «WSDLbinding» stereotyped element | Properties**

Define Properties

Field/Option/ Button	Action	See also
<b>Name</b>	Type in the name of the WSDL Binding element.	
<b>PortType</b>	Click on the drop-down arrow and select the WSDL PortType to be implemented by this WSDL Binding.	<a href="#">Port Type</a> <sup>[2433]</sup>
<b>Protocol</b>	Click on the drop-down arrow and select the protocol for the transmission of the selected WSDL PortType's operations. The supported protocols are: <ul style="list-style-type: none"> <li>• <b>SOAP</b></li> <li>• <b>HTTP</b></li> </ul>	
<b>Transport</b>	This field is disabled if you have selected the <b>HTTP</b> protocol. Defaults to <b>http://schemas.xmlsoap.org/soap/http</b> . If necessary, type in an alternative URL for the SOAP protocol.	
<b>Style</b>	This field is disabled if you have selected the <b>HTTP</b> protocol. Click on the drop-down arrow and select the style of SOAP protocol.	

<b>Verb</b>	<p>This field is disabled if you have selected the <b>SOAP</b> protocol.</p> <p>Click on the drop-down arrow and select the appropriate HTTP verb. The supported verbs are:</p> <ul style="list-style-type: none"> <li>• <b>GET</b></li> <li>• <b>POST</b></li> </ul>	
<b>Documentation</b>	(Optional) Type in any notes you need for this element.	
<b>OK</b>	Click on this button to <b>save</b> the data entered and close the WSDL Binding dialog.	
<b>Cancel</b>	Click on this button to <b>discard</b> the data entered and close the WSDL Binding dialog.	
<b>Help</b>	Click on this button to display this Help topic.	
<b>UML</b>	<p>This button is displayed when you are <b>editing</b> existing WSDL Binding element information.</p> <p>Click on the button to open the UML element Properties dialog for the element.</p>	<a href="#">Properties</a> <small>956</small>

#### Notes

- A WSDL Binding must implement a WSDL PortType; therefore, WSDL PortTypes should be defined before you create WSDL Bindings
- WSDL Bindings can only be created under the **Bindings** package in the WSDL Package structure
- The name of the WSDL Binding should be unique amongst all the WSDL Bindings within the WSDL

#### Learn more

- [WSDL Binding Operation](#) 2438
- [WSDL 1.1 Model Structure](#) 2423
- [Model WSDL](#) 2426

#### 14.2.2.4.1 WSDL Binding Operation

When you save a newly-created «WSDLbinding» stereotyped Class, the system:

1. Adds the **WSDL Port Type** element implemented by the **WSDL Binding** to the Binding diagram.
2. Draws a **Realization** connector from the Binding to the PortType.
3. Automatically populates the Binding with all the **UML operations** from the PortType, as the **WSDL Binding Operations**.

**Access** **Project Browser | Double-click on WSDL Binding operation**, or  
**Diagram | Highlight Binding element | Highlight Binding operation | double-click on operation**

### Define Properties

Field/Option/ Button	Action	See also				
<b>Operation Name</b>	Displays the name of the Operation copied from the WSDL PortType element.  The value in this field cannot be edited.					
<b>Action</b>	If the protocol of the parent WSDL Binding element was defined as <b>HTTP</b> , this field is grayed out.  Type in the <b>SOAP</b> Action header ( <i>URL</i> ) for this operation.	<a href="#">WSDL Binding</a> <sup>[2437]</sup>				
<b>Style</b>	If the protocol of the parent WSDL Binding element was defined as <b>HTTP</b> , this field is grayed out.  Click on the drop-down arrow and select the SOAP style of the operation.	<a href="#">WSDL Binding</a> <sup>[2437]</sup>				
<b>Location</b>	If the protocol of the parent WSDL Binding element was defined as <b>SOAP</b> , this field is grayed out.  Type in the relative URL of this Operation.	<a href="#">WSDL Binding</a> <sup>[2437]</sup>				
<b>Documentation</b>	(Optional) Type in any notes you need for this operation.					
<b>Parameters</b>	<p>Click on this button to define the parameters for this operation.</p> <p>The WSDL Binding Operation Parameters dialog displays, showing the names of the operation Input, Output and Faults. You cannot change these names.</p> <p>Click on the <b>Details</b> button to specify the details for Input, Output and Fault operation (Message)parameters. Note that the <b>Details</b> button in the:</p> <ul style="list-style-type: none"><li>• <b>Input</b> section is disabled for the <b>Notification</b> Operation Type</li><li>• <b>Output</b> section is disabled for the <b>One-way</b> Operation Type</li><li>• <b>Fault</b> section is disabled if there are no Fault Messages</li></ul> <table><tr><th>Field/Option/Button</th><th>Action</th></tr><tr><td><b>Use</b></td><td>If the protocol of the parent WSDL Binding element was defined as <b>HTTP</b>, this field is grayed out.  Click on the drop-down arrow and select the encoding used.</td></tr></table>	Field/Option/Button	Action	<b>Use</b>	If the protocol of the parent WSDL Binding element was defined as <b>HTTP</b> , this field is grayed out.  Click on the drop-down arrow and select the encoding used.	
Field/Option/Button	Action					
<b>Use</b>	If the protocol of the parent WSDL Binding element was defined as <b>HTTP</b> , this field is grayed out.  Click on the drop-down arrow and select the encoding used.					

	<b>Encoding Style</b>	If the protocol of the parent WSDL Binding element is <b>HTTP</b> , this field is grayed out.  If <b>Use</b> is set to <b>encoded</b> , type in the style (URL) to apply.	
	<b>Namespace</b>	If the protocol of the parent WSDL Binding element is <b>HTTP</b> , this field is grayed out.  (Optional) Type in the namespace.	
	<b>Parts</b>	If the protocol of the parent WSDL Binding element is <b>HTTP</b> , this field is grayed out.  This field is also not applicable to Faults.  (Optional) Type in the Message Part attributes that apply to the SOAP Body portion.	
	<b>Header</b>	This field is not applicable to Faults.  (Optional) Type in the text of the SOAP/HTTP Header.	
	<b>Documentation</b>	(Optional) Type in any notes you need for this message.	
	<b>OK</b>	Click on this button to <b>save</b> the data entered and close the Binding Parameter Details dialog.	
	<b>Cancel</b>	Click on this button to <b>discard</b> the data entered and close the Binding Parameter Details dialog.	
	<b>Help</b>	Click on this button to display this Help topic.	
<b>OK</b>	Click on this button to <b>save</b> the data entered and close the WSDL Binding Operation Details dialog.		
<b>Cancel</b>	Click on this button to <b>discard</b> the data entered and close the WSDL Binding Operation Details dialog.		
<b>Help</b>	Click on this button to display this Help topic.		
<b>UML</b>	<p>This button is displayed when you are <b>editing</b> existing WSDL Binding Operation information.</p> <p>Click on the button to open the UML operation Properties dialog for the element.</p>		<a href="#">General Properties of Operations</a> <small>[1015]</small>



--	--	--

### Notes

- If you subsequently change the WSDL Port Type operations, you can refresh the Binding Operations by deleting the Realization connector and re-establishing it; the Overrides & Implementations dialog displays, on which you select the updated operations to establish
- You can review the parameters of a Binding Operation by highlighting the Operation in the diagram or Project Browser and expanding the entries in the Tagged Values window

### Learn more

- [WSDL Binding](#) <sup>[2437]</sup>
- [WSDL 1.1 Model Structure](#) <sup>[2425]</sup>
- [Model WSDL](#) <sup>[2426]</sup>
- [Override Parent Operations](#) <sup>[1023]</sup>
- [Tagged Values](#) <sup>[1134]</sup>

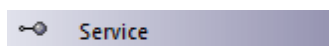
#### 14.2.2.5 WSDL Service

A **WSDL Service** is represented by a «*WSDLservice*» stereotyped Interface; it describes a collection of Ports that expose a particular Binding. You can create a WSDL Service element by dragging the **Service** icon from the WSDL Toolbox and dropping it **directly** onto a diagram in the *Services* package of your WSDL model.

When you save a newly-created «*WSDLservice*» stereotyped Interface, the system:

1. Adds the **WSDL Binding** elements exposed by the **WSDL Service** to the Service diagram.
2. Draws an **Association** connector from the Service element to each Binding element.
3. Labels each connector with the corresponding **Port** name.

### Toolbox Icon



**Access** **Drag icon onto diagram**, or  
**Project Browser | Double-click on «WSDLservice» stereotyped element**, or  
**Diagram | right-click on «WSDLservice» stereotyped element | Properties**

### Define Properties

Field/Option/ Button	Action	See also
<b>Name</b>	Type in the name of the WSDL Service.	
<b>Documentation</b>	(Optional) Type in any notes you need for this element.	



Ports	Identify the Ports (or endpoints) for this WSDL Service.  To add a Port to the list, click on the <b>New</b> button. The WSDL Port dialog displays.																	
	<table><tr><th>Field/Option/Button</th><th>Action</th></tr><tr><td><b>Port Name</b></td><td>Type in the name of the Port.</td></tr><tr><td><b>Binding</b></td><td>Click on the drop-down arrow and select a Binding element of all the WSDL Bindings created in the <i>Bindings</i> package.</td></tr><tr><td><b>Location</b></td><td>Type in the URL for the Port.</td></tr><tr><td><b>Documentation</b></td><td>(Optional) Type in any notes you need for this Port.</td></tr><tr><td><b>OK</b></td><td>Click on this button to <b>save</b> the values entered and close the dialog.</td></tr><tr><td><b>Cancel</b></td><td>Click on this button to <b>discard</b> the values entered and close the Port dialog.</td></tr><tr><td><b>Help</b></td><td>Click on this button to display this Help topic.</td></tr></table>	Field/Option/Button	Action	<b>Port Name</b>	Type in the name of the Port.	<b>Binding</b>	Click on the drop-down arrow and select a Binding element of all the WSDL Bindings created in the <i>Bindings</i> package.	<b>Location</b>	Type in the URL for the Port.	<b>Documentation</b>	(Optional) Type in any notes you need for this Port.	<b>OK</b>	Click on this button to <b>save</b> the values entered and close the dialog.	<b>Cancel</b>	Click on this button to <b>discard</b> the values entered and close the Port dialog.	<b>Help</b>	Click on this button to display this Help topic.	<a href="#">WSDL Binding</a> [2437]
	Field/Option/Button	Action																
	<b>Port Name</b>	Type in the name of the Port.																
	<b>Binding</b>	Click on the drop-down arrow and select a Binding element of all the WSDL Bindings created in the <i>Bindings</i> package.																
	<b>Location</b>	Type in the URL for the Port.																
	<b>Documentation</b>	(Optional) Type in any notes you need for this Port.																
	<b>OK</b>	Click on this button to <b>save</b> the values entered and close the dialog.																
	<b>Cancel</b>	Click on this button to <b>discard</b> the values entered and close the Port dialog.																
	<b>Help</b>	Click on this button to display this Help topic.																
	The Ports are organized in the list with the most recent at the top and the oldest at the end.  To remove an entry from the list, click on it and click on the <b>Delete</b> button.																	
OK	Click on this button to <b>save</b> the data entered and close the WSDL Service dialog.																	
Cancel	Click on this button to <b>discard</b> the data entered and close the WSDL Service dialog.																	
Help	Click on this button to display this Help topic.																	
UML	This button is displayed when you are <b>editing</b> existing WSDL Service element information.  Click on the button to open the UML element Properties dialog for the element.	<a href="#">Properties</a> [956]																

--	--	--

### Notes

- WSDL Services can only be created under the *Service* package in the WSDL Package structure
- The name of the WSDL Service should be unique amongst all the WSDL Services within the WSDL

### Learn more

- [WSDL 1.1 Model Structure](#) 
- [Model WSDL](#) 

## 14.2.2.6 WSDL Document

A **WSDL Document** encapsulates a Web Service defined within the «*WSDLnamespace*» stereotyped Package, and is the source from which the WSDL file is generated. It is represented by a «*WSDL*» stereotyped **Component** element as a direct child element of the «*WSDLnamespace*» stereotyped Package. You can have multiple WSDL Documents under a single WSDL Namespace, to reuse and expose the WSDL Services for that namespace across multiple WSDLs.

One «*WSDL*» stereotyped Component element is automatically created when you create the Namespace package structure. You can add further WSDL elements by dragging the **WSDL** icon from the WSDL Toolbox and dropping it directly onto the namespace Overview diagram.

### Toolbox Icon



**Access**    **Drag icon onto diagram**, or  
**Project Browser | Double-click on «WSDL» stereotyped Component element**, or  
**Diagram | right-click on «WSDL» stereotyped Component element | Properties**

### Define Properties

Field/Option/ Button	Action	See also
<b>Name</b>	Type in the name of the WSDL document.	
<b>File Name</b>	Type the file path into which the WSDL 1.1 file is to be generated.	
<b>Documentation</b>	(Optional) Type in any notes you need for this element.	
<b>XMLNS</b>	Identify the additional namespace or namespace-prefix pairs used in this WSDL Document.  To <b>add</b> a namespace or namespace-prefix pair, click on the <b>New</b> button; to <b>edit</b> an existing entry, double-click on it. In either case, the Namespace	

	<p>Details dialog displays.</p> <table><tr><th>Field/Option/Button</th><th>Action</th></tr><tr><td><b>Prefix</b></td><td>Type in the abbreviated text to represent the Namespace.</td></tr><tr><td><b>Namespace</b></td><td>Type in the name of the Namespace.</td></tr><tr><td><b>OK</b></td><td>Click on this button to <b>save</b> the new information and close the Namespace Details dialog.</td></tr><tr><td><b>Cancel</b></td><td>Click on this button to <b>discard</b> the new information and close the Namespace Details dialog.</td></tr><tr><td><b>Help</b></td><td>Click on this button to display this Help topic.</td></tr></table> <p>To remove an entry from the list, click on it and click on the <b>Delete</b> button.</p>	Field/Option/Button	Action	<b>Prefix</b>	Type in the abbreviated text to represent the Namespace.	<b>Namespace</b>	Type in the name of the Namespace.	<b>OK</b>	Click on this button to <b>save</b> the new information and close the Namespace Details dialog.	<b>Cancel</b>	Click on this button to <b>discard</b> the new information and close the Namespace Details dialog.	<b>Help</b>	Click on this button to display this Help topic.	
Field/Option/Button	Action													
<b>Prefix</b>	Type in the abbreviated text to represent the Namespace.													
<b>Namespace</b>	Type in the name of the Namespace.													
<b>OK</b>	Click on this button to <b>save</b> the new information and close the Namespace Details dialog.													
<b>Cancel</b>	Click on this button to <b>discard</b> the new information and close the Namespace Details dialog.													
<b>Help</b>	Click on this button to display this Help topic.													
<b>Services</b>	<p>Review the WSDL Services that exist in the <i>Services</i> package.</p> <p>Select the checkbox against the services to be included in the current WSDL file.</p>	<a href="#">WSDL Service</a> [244]												
<b>OK</b>	Click on this button to <b>save</b> the data entered and close the WSDL Document Properties dialog.													
<b>Cancel</b>	Click on this button to <b>discard</b> the data entered and close the WSDL Document Properties dialog.													
<b>Help</b>	Click on this button to display this Help topic.													
<b>UML</b>	<p>This button is displayed when you are <b>editing</b> existing WSDL Document element information.</p> <p>Click on the button to open the UML element Properties dialog for the element.</p>	<a href="#">Properties</a> [956]												

#### Learn more

- [WSDL 1.1 Model Structure](#) <sup>[2423]</sup>
- [Model WSDL](#) <sup>[2426]</sup>

### 14.2.3 Generate WSDL

If you have developed a WSDL model in UML, you can forward-engineer it into WSDL 1.1 files using the **Generate WSDL** feature. This feature acts on either a «*WSDLnamespace*» stereotyped Package or a «*WSDL*» stereotyped Component (**Document**), and generates any or all of the WSDL Components owned by the target «*WSDLnamespace*» structure.

**Access**    **Tools | Web Services | Generate WSDL**, or  
**Right-click «WSDL» stereotyped Component | Generate WSDL**

#### Generate WSDL 1.1 files

Field/Option/ Button	Action	See also								
<b>WSDL Package</b>	Displays the name of the WSDL Namespace containing the source Component(s) from which the WSDL file is to be generated.	<a href="#">WSDL Namespace</a> [2429]								
<b>Encoding</b>	Either: <ul style="list-style-type: none"><li>Click on the drop-down arrow and select the XML encoding scheme you need, or</li><li>Click on the <b>Default</b> button to apply the default encoding scheme (<b>UTF-8</b>)</li></ul>									
<b>Select Components To Generate</b>	<p>Click on the «<i>WSDL</i>» stereotyped Component(s) in the list for which the WSDL file is to be generated.</p> <p>To:</p> <ul style="list-style-type: none"><li>Select multiple individual Components use <b>Ctrl</b>+click</li><li>Select a range use <b>Shift</b>+click</li><li>Select all entries in the list click on the <b>Select All</b> button</li><li>Clear all entries in the list click on the <b>Select None</b> button</li><li>Provide a file path and name into which to generate the WSDL file for a component, double-click on the component name; the Component File Name dialog displays</li></ul> <table><tr><th>Field/Button</th><th>Action</th></tr><tr><td><b>Name</b></td><td>Displays the name of the selected «<i>WSDL</i>» stereotype</td></tr><tr><td><b>Prefix</b></td><td>If multiple prefixes have been specified, click on the dr and select the appropriate prefix for the WSDL Names</td></tr><tr><td><b>File Name</b></td><td>Type in or browse for (click on ( ... )) the file path and n WSDL file is to be generated.</td></tr></table>	Field/Button	Action	<b>Name</b>	Displays the name of the selected « <i>WSDL</i> » stereotype	<b>Prefix</b>	If multiple prefixes have been specified, click on the dr and select the appropriate prefix for the WSDL Names	<b>File Name</b>	Type in or browse for (click on ( ... )) the file path and n WSDL file is to be generated.	
Field/Button	Action									
<b>Name</b>	Displays the name of the selected « <i>WSDL</i> » stereotype									
<b>Prefix</b>	If multiple prefixes have been specified, click on the dr and select the appropriate prefix for the WSDL Names									
<b>File Name</b>	Type in or browse for (click on ( ... )) the file path and n WSDL file is to be generated.									

Field/Option/ Button	Action		See also
	<b>OK</b>	Click on this button to <b>save</b> the data entered and close File Name dialog.	
	<b>Cancel</b>	Click on this button to <b>discard</b> the data entered and close Component File Name dialog.	
	<b>Help</b>	Click on this button to display this Help topic.	
<b>Generate</b>	Click on this button to generate the WSDL files for the selected « <i>WSDL</i> » stereotyped Components.  A message displays when the generation is complete; click on the <b>OK</b> button on the message to close it.		
<b>View WSDL</b>	Click on this button to display the most recently generated WSDL.		
<b>Close</b>	Click on this button to close this dialog.		
<b>Help</b>	Click on this button to display this Help topic.		
<b>Progress</b>	Monitor the progress of the WSDL file generation.		

### Notes

- You can also generate WSDL files through the Automation Interface

### Learn more

- [WSDL 1.1 Model Structure](#) <sup>[2423]</sup>
- [Model WSDL](#) <sup>[2426]</sup>
- [WSDL Document](#) <sup>[2443]</sup>
- [Project Class](#) <sup>[2971]</sup>

### Learning Center topics

- (Alt+F1) | **Enterprise Architect** | **SOA and XML Engineering** | **WSDL** | **Generate WSDL**

### 14.2.4 Import WSDL

If you have WSDL 1.1 files external to Enterprise Architect that you want to reverse engineer into UML Class models, you can import them into the system using the **WSDL Import** facility.

**Access** **Project Browser** | **Click on root package to contain imported file**, then:  
**Tools** | **Web Services** | **Import WSDL**, or  
**Right-click on package** | **Code Engineering** | **Import WSDL**

#### Import a WSDL File

Field/Option/ Button	Action
<b>Filename</b>	Type in or browse for (click on ( ... )) the name and path of the WSDL file to import.
<b>Root Package</b>	Displays the name of the root Package under which the WSDL file is to be imported.
<b>Target Package</b>	Defaults to the name of the WSDL file being imported, as the name of the package to represent the imported file.  If you do not want to use the default name, type in a different name.
<b>Import</b>	Click on this button to start the WSDL Import.  A message displays when the import is complete; click on the <b>OK</b> button on the message to close it.
<b>Close</b>	Click on this button to close this dialog.
<b>Help</b>	Click on this button to display this Help topic.
<b>Progress</b>	Monitor the progress of the WSDL Import.

#### Notes

- Enterprise Architect cannot import a WSDL file that references WSDL constructs existing outside that file; if there are referenced constructs in other files, combine all files into a single file and import that combined file
  - Example of an importable file: [http://www.w3.org/TR/wsdl.html#\\_wsdl](http://www.w3.org/TR/wsdl.html#_wsdl)
  - Example of a non-importable file: [http://www.w3.org/TR/wsdl.html#\\_style](http://www.w3.org/TR/wsdl.html#_style); attempts to import this file result in the error message *Cannot Import Split Files*

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect** | **SOA and XML Engineering** | **WSDL** | **Import WSDL**



## 14.3 SoaML

**Service oriented architecture Modeling Language (SoaML)** is a standard method of designing and modeling **SOA** solutions using the Unified Modeling Language (**UML**).

The following text is derived from *Service oriented architecture Modeling Language (SoaML) - Specification for the UML Profile and metamodel for Services (UPMS)* (OMG document ad/2008-11-01); pp. 25-26:

A service is an offer of value to another through a well-defined interface and available to a community (which may be the general public). A service results in work provided to one by another.

Service Oriented Architecture (SOA) is a way of organizing and understanding (representations of) organizations, communities and systems to maximize agility, scale and interoperability. The SOA approach is simple - people, organizations and systems provide services to each other. These services allow us to get something done without doing it ourselves or even without knowing how to do it - enabling us to be more efficient and agile. Services also enable us to offer our capabilities to others in exchange for some value - thus establishing a community, process or marketplace. The SOA paradigm works equally well for integrating existing capabilities as for creating and integrating new capabilities.

SOA ... is an architectural paradigm for defining how people, organizations and systems provide and use services to achieve results. SoaML ... provides a standard way to architect and model SOA solutions using the Unified Modeling Language (UML). The profile uses the built-in extension mechanisms of UML to define SOA concepts in terms of existing UML concepts.

... the highest leverage of employing SOA comes from understanding a community, process or enterprise as a set of interrelated services and ... supporting that service oriented enterprise with service-enabled systems. SoaML enables business oriented and systems oriented services architectures to mutually and collaboratively support the enterprise mission. ... SoaML depends on Model Driven Architecture® (MDA®) to help map business and systems architectures, the design of the enterprise, to the technologies that support SOA, like web services and CORBA®.

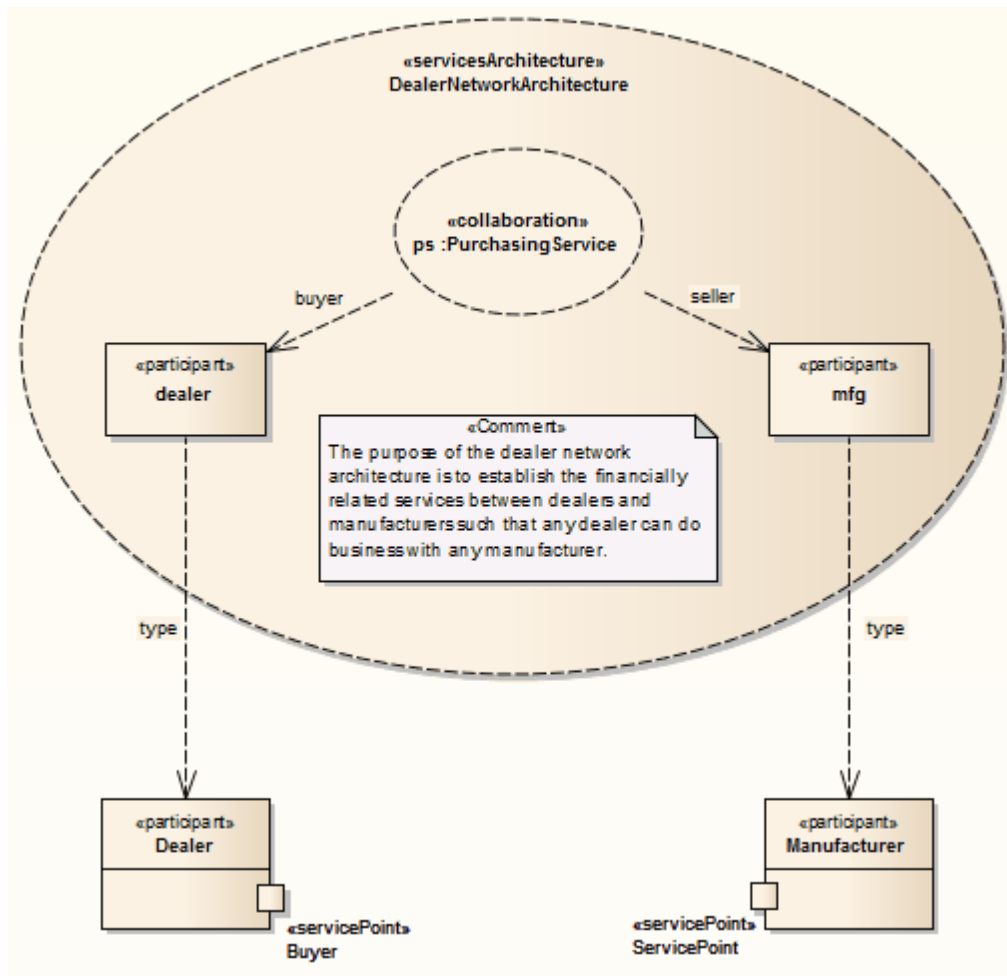
For further information on the concepts of SoaML, see the specification document on the OMG website [SoaML document page](#).

### SoaML in Enterprise Architect

In Enterprise Architect you can model **services architectures** quickly and simply through use of an **MDG Technology** integrated with the Enterprise Architect installer. The **SoaML** facilities are provided in the form of:

- Two SoaML **diagram types** - *SoaML Component Diagram* and *SoaML Sequence Diagram* - accessed through the New Diagram dialog
- SoaML pages in the **Diagram Toolbox**
- SoaML element and relationship entries in the **Toolbox Shortcut** menu and **Quick Linker**

### Example SoaML Diagram



### Disable SoaML

If you prefer not to use SoaML in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog (**Settings | MDG Technologies**).

### Notes

- Service Oriented Architecture Modeling Language (SoaML) is supported in the Corporate, Systems Engineering, Business and Software Engineering and Ultimate editions of Enterprise Architect

### Learn more

- [SOAML document page](#) (Online Resource)
- [SOA and XML](#) <sup>[2386]</sup>
- [New Diagram](#) <sup>[822]</sup>
- [Toolbox Shortcut Menu](#) <sup>[799]</sup>
- [SoaML Toolbox Pages](#) <sup>[2451]</sup>
- [Quick Linker](#) <sup>[896]</sup>
- [Manage MDG Technologies](#) <sup>[1477]</sup>

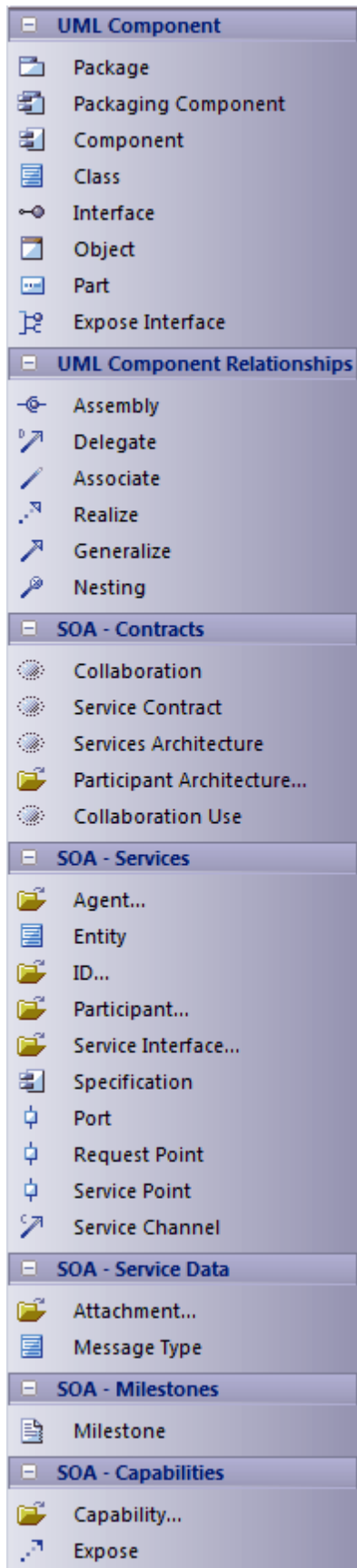
### 14.3.1 SoaML Toolbox Pages

You can create the elements and relationships of the **SoaML model** using the SoaML pages of the **Diagram Toolbox**. Each of the two SoaML diagram types has a separate set of pages, although the last five (SOA-specific) pages in the two sets are identical.

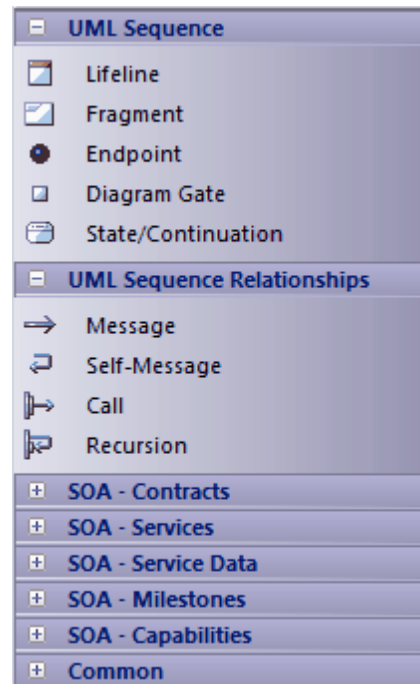
Access   **Diagram | Diagram Toolbox (Alt+F) | More tools | SoaML Component**, or  
**SoaML Sequence**

Toolbox Pages

## SoaML Component Diagram



## SoaML Sequence Diagram



**Learn more**

- [SoaML](#) [2449]
- [SOA and XML](#) [2386]

## 14.4 SOMF 2.1

The **Service-Oriented Modeling Framework (SOMF)** is a service-oriented development life cycle methodology, offering a number of modeling practices and disciplines that contribute to successful service-oriented life cycle management and modeling. The following text is derived from the extensive Wikipedia entry on Service Oriented Modeling:

The Service-Oriented Modeling Framework (SOMF) has been proposed by author Michael Bell as a holistic and anthropomorphic modeling language for software development that employs disciplines and a universal language to provide tactical and strategic solutions to enterprise problems. The term "holistic language" pertains to a modeling language that can be employed to design any application, business and technological environment, either local or distributed. This universality may include design of application-level and enterprise-level solutions, including SOA landscapes or Cloud Computing environments. The term "anthropomorphic", on the other hand, affiliates the SOMF language with intuitiveness of implementation and simplicity of usage.

SOMF ... illustrates the major elements that identify the "what to do" aspects of a service development scheme. These are the modeling pillars that will enable practitioners to craft an effective project plan and to identify the milestones of a service-oriented initiative- either a small or large-scale business or a technological venture.

### SOMF in Enterprise Architect

In Enterprise Architect, SOMF 2.1 is implemented as a profile within an MDG Technology that is integrated with the Enterprise Architect installer. The SOMF 2.1 facilities are provided in the form of:

- Eleven SOMF diagram types, accessed through the New Diagram dialog:

*Conceptual  
Analysis  
Cloud Computing  
Logical Design Relationship  
Logical Design Composition  
Business Integration  
Conceptual Architecture  
Asset Utilization  
Transaction  
Transaction Directory  
Reference Architecture*




- **SOMF pages in the Toolbox** - Enterprise Architect includes several Toolbox pages of modeling structures for each SOMF 2.1 diagram type, through the **More tools | SOMF 2.1** menu option; these provide a wide breadth of SOMF modeling capabilities
- SOMF element and relationship entries in the **Toolbox Shortcut** menu and **Quick Linker**

### Disable SOMF

If you prefer not to use SOMF in Enterprise Architect, you can disable it (and subsequently re-enable it) using the MDG Technologies dialog (**Settings | MDG Technologies**).

### Learn more

- [Service Oriented Modeling](#) (Online Resource)
- [Service-Oriented Modeling Framework™ in Enterprise Architect](#) product page (Online Resource)
- [New Diagram](#) <sup>[822]</sup>

- [Toolbox Shortcut Menu](#)  799
- [Quick Linker](#)  896
- [MDG Technologies](#)  1477

## 14.5 MOF

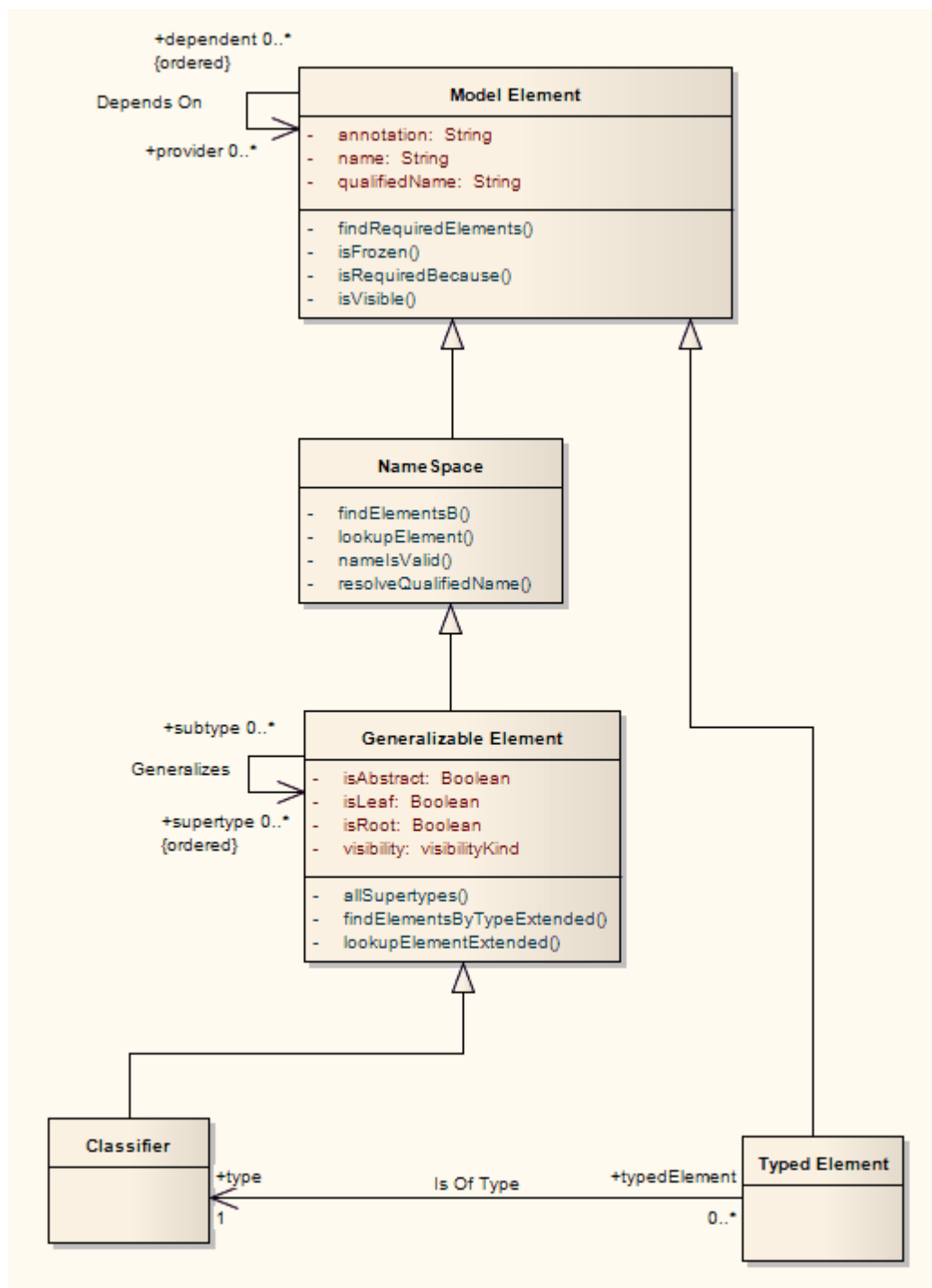
The **Meta-Object Facility (MOF)** is an Object Management Group (OMG) standard developed as a meta-modeling architecture to define the UML, and so provides a means to define the structure or abstract syntax of a language or of data. The MOF is designed as a **four-layered architecture**; being a **closed, strict meta-modeling** architecture, every model element on every layer is strictly an instance of a model element of the layer above.

Simplified, the MOF uses Classes to define concepts (model elements) on a meta-layer. These Classes (concepts) can then be instantiated through objects (instances) of the model layer below. Because an element on the M2 layer is an object (instance of an M3 model element) as well as a Class (an M2 layer concept) the notion of a **clabject** is used - a merge of the words *Class* and *Object*.

Because of the similarities between the MOF model and UML structure models, the MOF meta-models are usually modeled as UML Class diagrams. You can also use the **Metamodel** page of the Diagram Toolbox (**More tools | UML | Metamodel**) to create MOF model elements and connectors.

The layered architecture of the Meta Object Facility is illustrated in this diagram.





### Notes

- A supporting standard of MOF is **XMI**, which defines an XML-based exchange format
- From Enterprise Architect, you can **export** packages to XMI under the **MOF 1.3** or **MOF 1.4 XMI file specifications**
- A related standard is the **Object Constraint Language (OCL)**, which describes a formal language that can be used to define model constraints by means of predicate logic; OCL makes a MOF model more precise by associating assertions with its meta-elements

Learn more

- [Class Diagrams](#)<sup>[1184]</sup>
- [Metamodel Toolbox](#)<sup>[812]</sup>
- [Create MOF Diagrams](#)<sup>[2458]</sup>
- [Export MOF Model to XML](#)<sup>[2460]</sup>
- [Object Constraint Language \(OCL\) Specification](#) (Online Resource)

Learning Center topics

- (Alt+F1) | **Enterprise Architect | SOA and XML Engineering | MOF**

### 14.5.1 Create MOF Diagrams

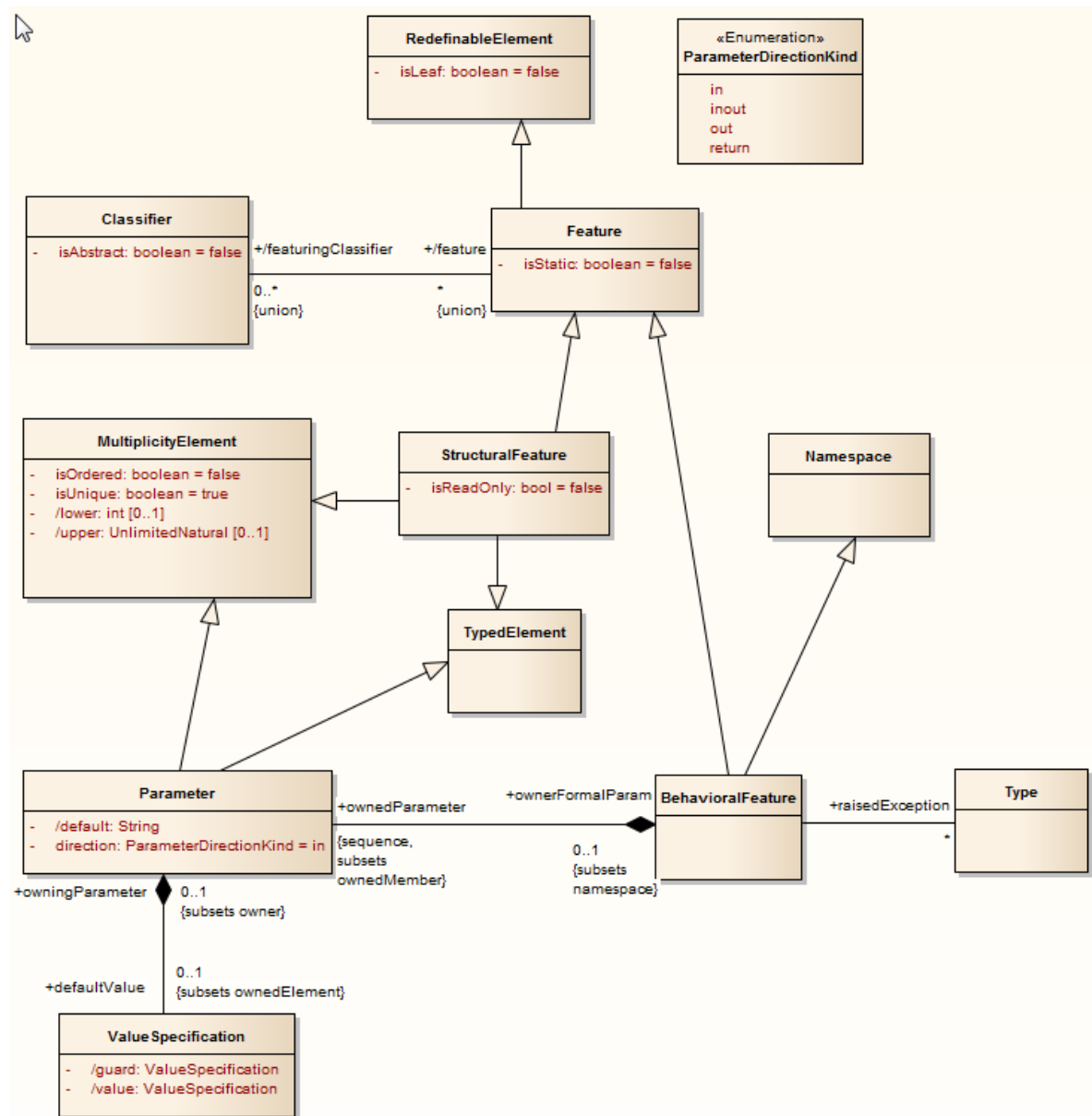
You can model the structure or abstract syntax of a language or of data using a **Meta Object Facility (MOF)** diagram. This is a **Class** diagram that is contained in a Package to which you assign a «*metamodel*» stereotype.

Create a MOF diagram

Step	Action	See also
1	Create or select the appropriate root Model Package and View for the MOF package.	<a href="#">Model Wizard</a> <sup>[753]</sup> <a href="#">Views</a> <sup>[769]</sup>
2	Create a package to contain your MOF elements. As part of this process, create a child <b>Class</b> diagram for this package.	<a href="#">Add a Package</a> <sup>[772]</sup>
3	In the Project Browser, double-click on the package name to display the Properties dialog.	<a href="#">Properties Dialog</a> <sup>[956]</sup>
4	In the <b>Stereotype</b> field type the value <b>metamodel</b> . Click on the <b>OK</b> button.	<a href="#">General Settings</a> <sup>[958]</sup>
5	Open the child Class diagram and begin to construct your model. In the Diagram Toolbox you can use either the: <ul style="list-style-type: none"> <li>• Class pages (<b>More tools   UML   Class</b>) or the</li> <li>• Metamodel pages (<b>More tools   UML   Metamodel</b>)</li> </ul> <p>A MOF diagram typically contains Class, Enumeration and Primitive elements (the Primitive elements from the Class Toolbox pages), and Generalization, Association, Compose and Aggregate relationships.</p>	<a href="#">Class Toolbox</a> <sup>[803]</sup> <a href="#">Metamodel Toolbox</a> <sup>[812]</sup>

Step	Action	See also

### Example MOF diagram



### Learn more

- [Class Diagram](#) <sup>1184</sup>

### Learning Center topics

- (Alt+F1) | [Enterprise Architect](#) | [SOA and XML Engineering](#) | [MOF](#) | [Create a MOF Model](#)

## 14.5.2 Export MOF Model to XMI

When you have created a MOF model, you can export it to an XMI file in either MOF 1.3 or MOF 1.4 XMI file specification format, to use in other applications or to transfer into other projects.

Access [Project Browser](#) | [Right-click MOF Root Package](#) | [Import/Export](#) | [Export Package to XMI file](#)  
(Ctrl+Alt+E): Publish

### Export MOF Model to XMI

Field/Option/ Button	Action	See also
<b>Root Package</b>	Displays the name of the selected model root package.	
<b>Filename</b>	Type in or browse for (click on ( ... )) the file path and name of the target XML file.	
<b>XML Type</b>	Scroll down to the end of the list and click on the format you want to use, either <b>MOF 1.3</b> or <b>MOF 1.4</b> .	
<b>Format XML Output</b>	(Optional) Select this checkbox to format the output into readable XML (this takes a few more seconds at the end of the run).	
<b>Write Log File</b>	(Optional, but recommended) Select this checkbox to write a log of the export activity.  The log file is saved to the directory into which the XML file is exported.	
<b>Exclude EA Extensions</b>	(Optional) Select the checkbox to exclude tool-specific information from the export.	
<b>Generate Diagram Images</b>	Pre-set to selected, to generate the exported diagrams into a package called <i>Images</i> in the directory into which the XML file is exported.  You specify the graphics format for these diagram images in the <b>Format</b> field, below.	
<b>Format</b>	Click on the drop-down arrow and select the graphics format you want to use for the exported diagram images.	
<b>Stylesheet</b>	(Optional) Click on the drop-down arrow and select an XSL Stylesheet to post-process the XML content before saving the package to file, using an XSLT to convert the output to HTML, XSL, code or other versions of XML.  The XSL style sheet should previously have been imported into the	<a href="#">Resources</a> 1173

Field/Option/ Button	Action	See also
	project through the Resources window.	
<b>Export</b>	Click on this button to initiate the export.	
<b>Progress</b>	Monitor the progress of the export - the indicator on the right of this field is completely filled with green when the export is complete.	
<b>View XML</b>	Click on this button to view the file you have created.	

### Notes

- MOF models exported to XML can be imported into an Enterprise Architect project using the standard import XML features of the system

### Learn more

- [MOF](#)<sup>[2456]</sup>
- [Publish Model Package](#)<sup>[476]</sup>
- [Import from XML](#)<sup>[478]</sup>

### Learning Center topics

- (Alt+F1) | **Enterprise Architect | SOA and XML Engineering | MOF | Export MOF to XML**

**Part**

---

**XV**

## 15 Model Simulation










Model Simulation brings your behavioral models to life with instant, real-time behavioral model execution. Coupled with tools to manage triggers, events, guards, effects, breakpoints and simulation variables, plus the ability to visually track execution at run-time, the Simulator is a powerful means of "watching the wheels turn" and verifying the correctness of your behavioral models. With Simulation you can explore and test the dynamic behavior of models. In the Corporate and above editions, you can also use Javascript as a run-time execution language for evaluating guards, effects and other script-able pieces of behavior.







Extensive support for triggers, trigger sets, nested states, concurrency, dynamic effects and other advanced simulation capabilities, provides a remarkable environment in which to build interactive and working models which help explore, test and visually trace complex business, software and system behavior. With Javascript enabled, it is also possible to create embedded COM objects which will do the work of evaluating guards and executing effects - allowing the simulation to be tied into a much larger set of dependent processes. For example, a COM object evaluating a guard condition on a State Transition might query a locally running process, read and use a set of test data - or even connect to an SOA web service to obtain some current information.

As Enterprise Architect uses a dynamic, script driven Simulation mechanism there is no need to generate code or compile your model before running a simulation. It is even possible to update simulation variables in real time using the Simulation console window. This is useful for testing alternate branches and conditions "on the fly", either at a set Simulation break point, or when the Simulation reaches a point of stability (for example, when the Simulation is 'blocked').

In the Professional version of Enterprise Architect, you can manually walk through simulations - although no Javascript will execute - so all choices are manual decisions. This is useful for testing the flow of a behavioral model and highlighting possible choices and processing paths.

### Simulation Overview

Topic	Link
<b>Overview of the Model Simulator</b>	<a href="#">How It Works</a>  <sup>2466</sup> <a href="#">How it Looks</a>  <sup>2468</sup>
<b>Use of the Simulation window and related windows, and running a simulation</b>	<a href="#">Simulation Windows</a>  <sup>2469</sup> <a href="#">Run Model Simulation</a>  <sup>2475</sup>
<b>Set up a simulation and activate a simulation script</b>	<a href="#">Set Up Simulation Script</a>  <sup>2472</sup> <a href="#">Activate Simulation Script</a>  <sup>2474</sup>
<b>Set up and use simulation breakpoints</b>	<a href="#">Simulation Breakpoints</a>  <sup>2477</sup>
<b>Simulate the use of Objects</b>	<a href="#">Objects and Instances in Simulation</a>  <sup>2479</sup>
<b>The use of different types of Action in simulation</b>	<a href="#">Action Behavior By Type</a>  <sup>2497</sup>

	<a href="#">Action Types</a>  <a href="#">Variable Actions</a> 
Perform dynamic simulation with Javascript	<a href="#">Dynamic Simulation with Javascript</a>  <a href="#">Interaction Operand Condition and Message Behavior</a> 
The use of Guards and Effects in simulations	<a href="#">Guards and Effects</a> 
The use of Triggers in simulations	<a href="#">Triggers</a>  <a href="#">Simulation Events Window</a>  <a href="#">Waiting Triggers</a>  <a href="#">Re-Signal Triggers</a>  <a href="#">Trigger Parameters</a>  <a href="#">Trigger Sets and Auto-Firing</a>  <a href="#">Using Trigger Sets to Simulate an Event Sequence</a> 
Call Behaviors and Variables	<a href="#">Call Behaviors</a> 
Simulating Activity Returns	<a href="#">Activity Return Value Simulation</a> 
Simulating Structured Activity Behavior	<a href="#">Structured Activity Simulation</a> 
Simulating multi-threaded processes	<a href="#">Multi-threading - Forks and Joins</a>  <a href="#">Multi-threading - Concurrent State Regions</a> 
Simulating subprocesses in separate diagrams	<a href="#">Using Composite Diagrams</a> 
Performing BPMN Simulations	<a href="#">BPMN Simulation</a> 
Simulate Win32 Dialog Behavior	<a href="#">Win32 Dialog Simulation</a> 



### Notes

- Currently Javascript is enabled for State Machines and Activity graphs; it is not enabled for Interaction diagrams

### Learning Center

- (Alt+F1) | **Enterprise Architect | Simulation | Simulation**
- (Alt+F1) | **Enterprise Architect | Simulation | Simulation | Define a Simulation Model**

## 15.1 How It Works

The Model Simulator enables you to simulate the execution of conceptual model designs containing behavior. When you start a simulation, the current model package is analyzed and a dynamic simulation process spawned to execute the model. As the simulator analyzes and works with UML constructs directly, there is no requirement to generate intermediary code or compile simulation 'executables'. This results in a very rapid and dynamic simulation environment in which changes can be made and tested very rapidly.

To get up and running with simulation the only steps required are:

- Build a behavioral diagram (state or activity for manual or dynamic execution, sequence for manual interaction only)
- Optional: load the Simulation Workspace layout ... a fast way of bringing up all the frequently used Simulation windows
- Press the simulator **Play** button!

The Professional edition provides a quick and simple way to verify your design's behavior for logical correctness by manually stepping through a diagram. In the Corporate edition and above it is possible to:

- Dynamically execute your behavioral models
- Assess guards and effects written in standard Javascript
- Define and fire triggers into running simulations
- Define and use sets of triggers to simulate different event sequences
- Auto-fire trigger sets to simulate complex event histories without user intervention
- Update simulation variables "on the fly" to change how simulations proceed
- Create and call COM objects during Simulation to extend the Simulation's reach and input/output possibilities
- Inspect Simulation variables at run time
- Set a script 'prologue' for defining variables, constants and functions prior to execution
- Use multiple Analyzer Scripts with differing 'prologues' for running the Simulation under widely differing conditions
- In the Business and Software Engineering and Ultimate Editions it is also possible to simulate BPMN models

### Platforms and Editions Available

Topic	Detail	See also
<b>Models and Platforms Supported</b>	<p>The Model Simulator currently supports the execution of UML Activity, Interaction, State Machine models and BPMN Business Processes on the simulation platforms:</p> <ul style="list-style-type: none"> <li>• UML Basic</li> <li>• BPMN</li> </ul>	<a href="#">Set Up Simulation</a> <small>[2472]</small>
<b>Edition Support</b>	<p>Model Simulation is available at different levels across the range of editions of Enterprise Architect:</p> <ul style="list-style-type: none"> <li>• Desktop - Simulation not available</li> </ul>	

Topic	Detail	See also
	<ul style="list-style-type: none"><li>• Professional - Manual Simulation only</li><li>• Corporate and above - Adds dynamic Javascript evaluation</li><li>• Business &amp; Software Engineering and Ultimate - Adds BPMN simulation</li></ul>	

#### Learn more

- [Set Up Simulation Script](#)<sup>[2472]</sup>
- [Run Model Simulation](#)<sup>[2475]</sup>
- [Diagram Context Menu](#)<sup>[778]</sup>

#### Learning Center

- (Alt+F1) | **Enterprise Architect | Simulation | Simulation | Example Models**

## 15.2 How it Looks

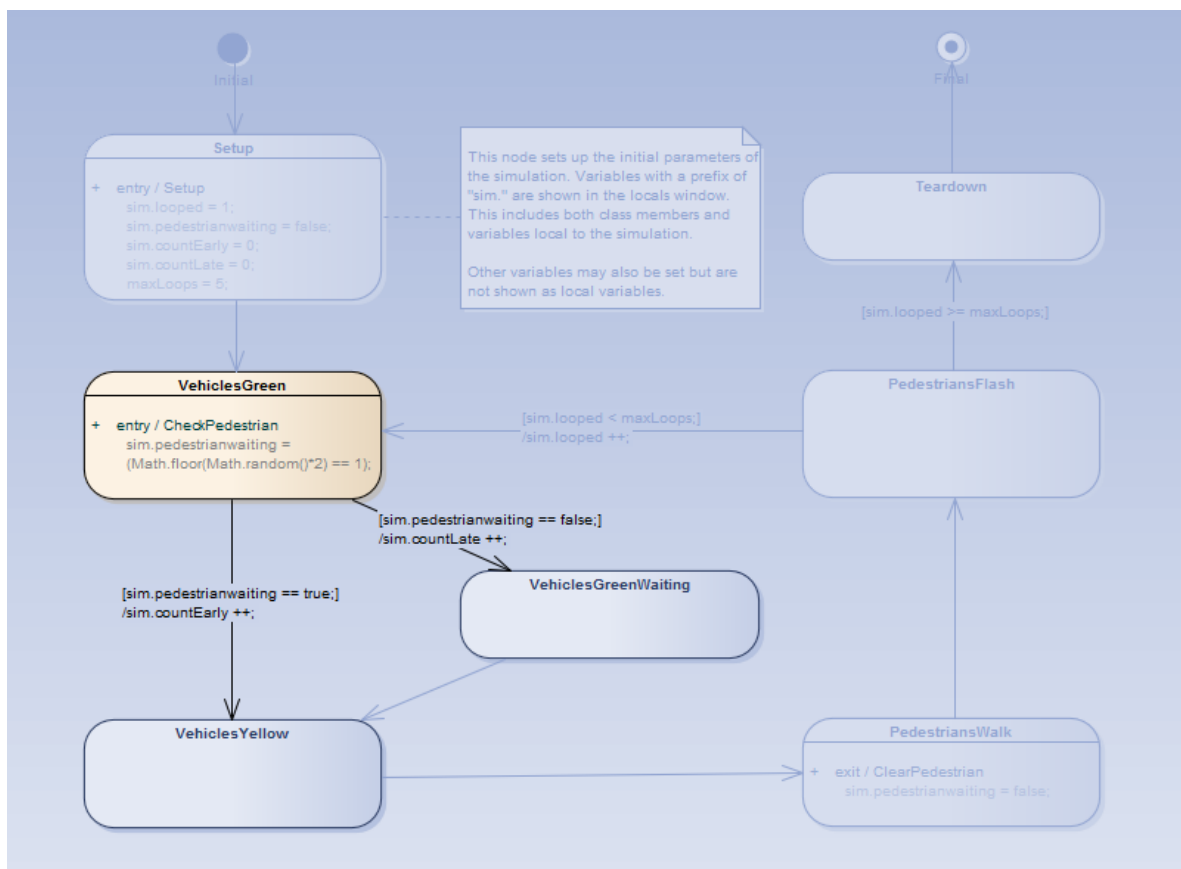
Enterprise Architect has a **special way of displaying model information during simulation**. This helps focus attention on the executing or active nodes.

During a Simulation, Enterprise Architect will **dynamically track and highlight the active nodes** within your model. If a node in another diagram is activated, that diagram will be automatically loaded and the current node highlighted. It is possible to modify the diagram while the simulation is running, however, the changes made are not recognized until the current simulation is ended and a new one begun.

### Highlighting of the active node (s) during simulation

- In the example below the currently active node ("VehiclesGreen") is highlighted in normal Enterprise Architect colors
- All possible **transitions out of the current node are rendered at full strength**
- The elements that are **possible targets** of the current active node's outgoing transitions are **rendered in a semi-faded style** so they are readable and clearly different to the other elements within the diagram
- All other elements are rendered in a **"fully faded"** style to show they are not targets of the next Simulation step.

As the Simulation progresses (especially if automatically run), this **highlighting helps focus the attention on the current item and its visual context**.

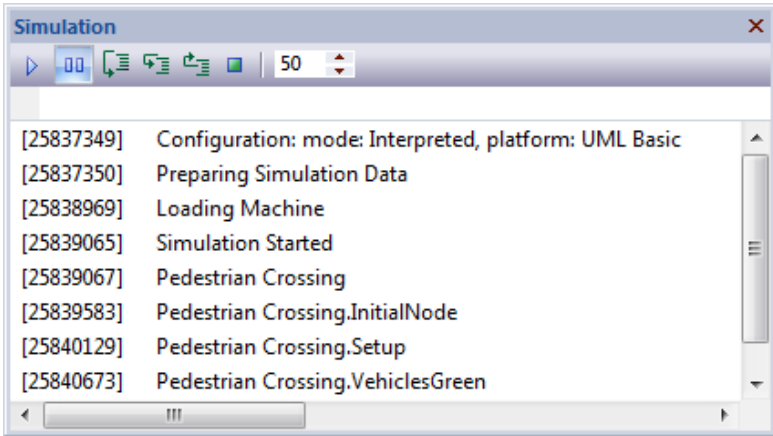


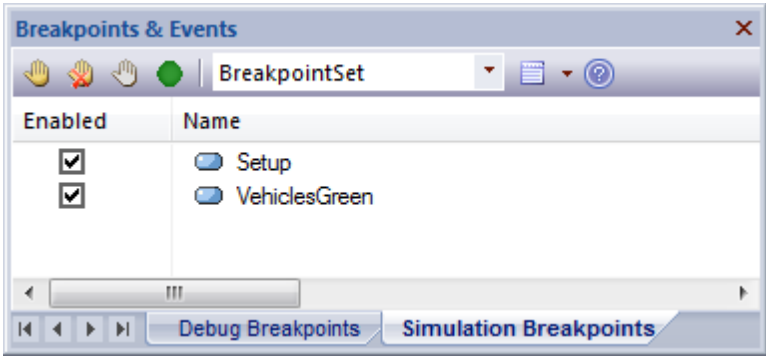
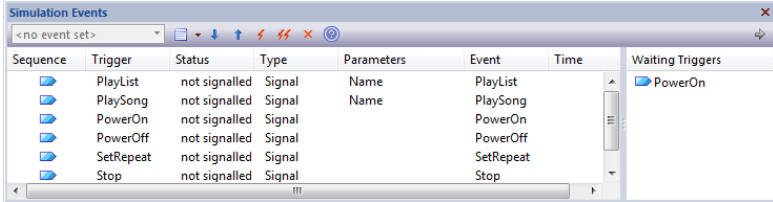
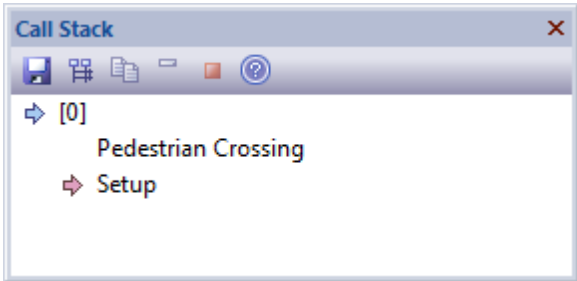
## 15.3 Simulation Windows

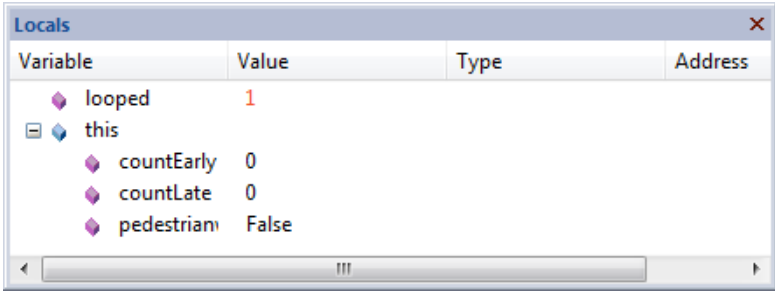
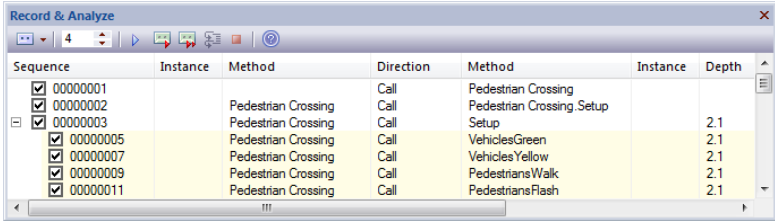
When executing a Simulation in Enterprise Architect it is possible to set **break-points**, fire **triggers**, examine **variables**, record a **trace** of execution, set simulation **speed**, view the **call stack** and **visually** trace the active nodes as the simulation proceeds.

When a Simulation runs, some aspects such as the output and console input are found in the Simulator window itself, while others such as the local variables and call-stack use the standard Execution Analyzer windows. The table below provides an overview of the main Windows used during Simulation.

**Access** [Analyzer](#) | [Simulator](#)

Window	Purpose	See also
<b>Execution and Console</b>	<p>The <b>Simulation window</b> provides the main interface for starting, stopping and stepping your Simulation. During execution it displays output relating to the currently executing step and other important information. See the <i>Run Model Simulation</i> topic for more information on the toolbar commands.</p> <p>Note the text entry box just underneath the toolbar. This is the <b>Console input area</b> - in here you can type simple <b>javascript</b> commands such as: <code>this.count = 4</code>; to dynamically change a Simulation variable named "count" to 4. In this way you can dynamically influence simulation at run-time.</p> 	<a href="#">Run Model Simulation</a> <sup>[2475]</sup>
<b>Breakpoints &amp; Events Window</b>	<p>The Simulation process also makes use of the <b>Simulation Breakpoints tab</b> of the <b>Breakpoints &amp; Events</b> window (<b>Analyzer   Breakpoints &amp; Events</b>). In here you set execution breakpoints on specific elements and messages in a Simulation. See the <i>Simulation Breakpoints</i> topic for more details.</p>	<a href="#">Simulation Breakpoints</a> <sup>[2477]</sup>

		
<b>Simulation Events Window</b>	<p>The <b>Simulation Events</b> window (<b>Analyzer   Simulation Events</b>) provides tools to manage and execute triggers. Triggers are used to <b>control the execution of State machines</b> transitions.</p> 	<a href="#">Simulation Events Window</a> <sup>[2504]</sup> <a href="#">Triggers</a> <sup>[2494]</sup> <a href="#">Waiting Triggers</a> <sup>[2508]</sup>
<b>Call Stack Window</b>	<p>During the Simulation the <b>Call Stack</b> window (<b>Analyzer   Call Stack</b>) displays information about the <b>Threads</b> and current execution context of the Simulation.</p> <p>The Simulator supports <b>multi-threaded Simulations</b> and will include a Thread entry for every active and paused thread of execution. For each thread, the Call Stack window will show the start or entry context (eg. a State Machine element) plus the current active element within that thread. If the current active element is the entry point of a composite activity or sub-machine state, the stack will also include the current active element within that sub-context (and all further nested, active composite, sub-states as well).</p> 	<a href="#">View the Call Stack</a> <sup>[2238]</sup>
<b>Simulation Local Variable Window</b>	<p>The Simulator uses the standard <b>Locals</b> window (<b>Analyzer   Locals</b>) to show all current Simulation <b>variables</b> when the simulation is <b>paused</b> at a <b>break point</b> or <b>single stepping</b>. Note that it is possible to dynamically update these variables using the Simulator Console described above.</p>	<a href="#">View the Local Variables</a> <sup>[2234]</sup>

		
<b>Recording</b>	<p>During execution of your Simulation, a <b>recording</b> is kept of all activity and displayed in the <b>Record &amp; Analyze</b> window (<b>Analyzer   Recorder</b>). This is similar to how the normal call recording works in the Visual Execution analyzer - although at this time, no ability to generate Sequence diagrams from recordings is possible.</p> 	<p><a href="#">The Recording History</a><sup>[2533]</sup></p>

## 15.4 Set Up Simulation Script

You **can** use **Simulation scripts** to provide fine control over how a simulation starts. In general, you do **not** need to set up a Simulation script unless:

- You want to run an **interpreted** simulation that requires variables to be **initialized** before the simulation commences; this is useful for setting up global variables and defining functions
- (In the Corporate Edition and above) You do not want to apply the default behavior of interpreting the Guards (that is, you prefer to use a **Manual** execution), or
- You want to have **multiple** ways of running the same diagram

For most diagrams it is possible to initialize a script for a simulation simply by setting variables in the first element or connector after the Start element. For **State Charts**, this is the **Transit** connector exiting the initial element, and for **Activity** models this is the first **Action** element.

As an alternative, you can use Simulation scripts to initialize settings before a simulation starts. This is useful for setting up different sets of initial values using multiple Analyzer Scripts, so that you can run your simulation under a range of pre-set conditions.




To **configure** a simulation execution script, first select the **Package** in the Project Browser, Package Browser, Diagram List or Model Search. You can then use the Execution Analyzer window to add a new Script for that selected package. You will use the Simulation page of the Execution Analyzer dialog to configure the relevant properties.

**Access**    **Analyzer | Execution Analyzer (Shift+F12) : New** (Toolbar icon)  
**Analyzer | Execution Analyzer (Shift+F12) : locate and double-click on required script > Simulation**

### Configure a Simulation Script

Field/Option	Action	See also
<b>Entry Point</b>	Click on the ( ... ) button and select the: <ul style="list-style-type: none"> <li>• <b>Entry point</b> for the simulation, and</li> <li>• Activity, Interaction or State Machine to simulate</li> </ul> <p>If you do not specify an entry point, the simulator attempts to work through the entire Package.</p>	
<b>Platform</b>	For UML Activity, Interaction or State Machine simulation, click on the drop-down arrow and select <b>UML Basic</b> .  For BPMN diagrams, click on the drop-down arrow and select <b>BPMN</b> .	
<b>Evaluate Guards and Effects using JavaScript</b>	(In Corporate and higher editions) <b>Leave</b> the checkbox unselected to perform a manual simulation, where you select the next State to transition to and the point where a decision must be made.  <b>Select</b> the checkbox to execute the code for Effect behavior in the simulation. The simulation executes <b>JavaScript</b> code in the following places: <ul style="list-style-type: none"> <li>• State <b>entry/exit/do</b> operations</li> </ul>	<a href="#">State Machine Table Conventions</a>







	<ul style="list-style-type: none"> <li>• <b>Transition guard/effect</b></li> <li>• BPMN Activity <b>Loop Conditions</b> and <b>Sequence Flow Condition Expressions</b></li> </ul> <p>With the exception of the guard, all of these should be one or more valid <b>JavaScript</b> statements, <b>including the semi-colon</b>.</p> <p>The guard must be a valid <b>boolean</b> expression, also terminated with a semi-colon.</p> <p>Variables that are members of of <b>sim</b> or <b>this</b> are listed in the Locals window when a simulation breakpoint is reached.</p> <pre>sim.count = 0;</pre>	 <a href="#">Transition</a>   <a href="#">View the Local Variables</a> 
<b>Input</b>	When Javascript is enabled, you can type script commands in this field that will <b>execute prior to the simulation being run</b> .	
<b>OK</b>	Click on this button to save your changes.	

**Notes**

- All simulation elements and relationships must reside within the Package configured for simulation

**Learn more**

- [How Simulation Works](#) 
- [Activate Simulation Script](#) 
- [Run Model Simulation](#) 
- [Using Simulation Breakpoints](#) 

**Learning Center topics**

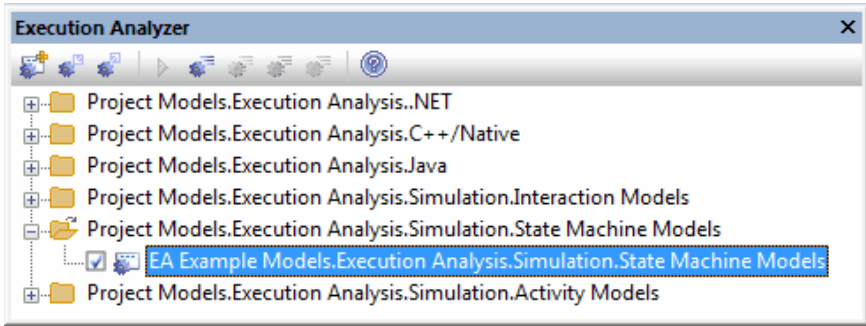

- (Alt+F1) | **Enterprise Architect | Simulation | Simulation | Configure a Simulation**

## 15.5 Activate Simulation Script

An Execution Script is configured for a model package defining the simulation parameters. The most common reason for activating an execution script is when multiple simulation scripts are configured against a package and you want to run a specific one.

**Access** [Project Browser package context menu](#) | [Execution Analyzer](#)

### Activate a Simulation script for execution

Step	Action	See also
1	<p>In the Execution Analyzer window, <b>select the required execution script</b>. This makes it the current default for your open model, so that clicking on the Simulation run button will automatically invoke this Simulation script.</p> 	
2	<p><b>Click on the checkbox</b> to the left of the script to activate it.</p>	
3	<p>Select the <b>Analyzer   Simulator</b> menu option to <b>execute the simulation</b>.</p>	<a href="#">Run Model Simulation</a>  <sup>2475</sup>

### Learn more

- [Set Up Simulation Script](#)  <sup>2472</sup>

### Learning Center topics

- (Alt+F1) | [Enterprise Architect](#) | [Simulation](#) | [Simulation](#) | [Run a Simulation](#)

## 15.6 Run Model Simulation

The simulation executes the model step-by-step, enabling you to validate the logic of your behavioral model. The current execution step is automatically highlighted in the model's diagram to make it easy to understand the various processes and state changes as they occur during the simulation.

There are several ways to **start a model simulation**



- When the active diagram can be simulated the **Run** button on the main Simulation window will run the current diagram, either by running an existing script or defining a new temporary one.
- When the active diagram can not be simulated the **Run** button on the main Simulation window will run the simulation for the active Execution Analyzer script.
- By right clicking on a Simulation script in the Execution Analyzer window and selecting the "Start Simulation" option;
- By right clicking on a suitable diagram and selecting one of the Execute Simulation options from the context menu.




**Visual cues during execution.** When the simulation is running, Enterprise Architect will **actively highlight each active node for each executed step**. In addition, all outgoing transitions, control flows & etc. will be highlighted, showing the possible paths forward. Elements at the end of possible paths forward will be de-emphasized to half-strength and any other remaining elements will be 90% "grayed out". This provides a very dynamic and easy to follow execution that continually refocuses attention on the execution 'context'.

Access **Analyzer | Simulator**

### Running a Simulation

Topic	Detail	See also
<b>Edition Specific Details</b>	<p>In the <b>Professional edition</b>, if a branch is encountered in the execution, the simulator prompts you to choose the appropriate path to take in your execution.</p> <p>In the <b>Corporate and above editions</b>, in which <b>Javascript</b> is enabled, the Simulation will automatically evaluate all guards, effects &amp; etc. and dynamically execute the Simulation without user intervention. If the Simulation becomes <b>blocked</b> due to no possibly paths forward evaluating to true (or multiple paths evaluating to true) you can <b>modify Simulation variables</b> on the fly using the console input of the Simulation Execution window.</p>	<a href="#">Activate Simulation Script</a> <sup>[2474]</sup>

Icon	Action	See also
	<b>Start the simulator</b> for the current simulatable diagram, or when the current diagram can't be simulated run simulation using the activated simulation script.	<a href="#">Activate Simulation Script</a> <sup>[2474]</sup>
	<b>Pause</b> the simulation.	

Icon	Action	See also
	When the simulation is paused, <b>step over</b> , <b>step in</b> and <b>step out</b> to control the simulator's execution at the required step in the model simulation.	
	<b>Stop</b> the simulation.	
	Varies the execution rate of the simulation, between 0% and 100%; at: <ul style="list-style-type: none"> <li>• <b>100%</b>, the simulation executes at the fastest possible rate</li> <li>• <b>0%</b> the simulator breaks execution at every statement</li> </ul>	

### Notes

- The Simulation tool only becomes active when a valid simulation Execution Script is activated
- You can set a Simulation script as the current default by setting its check box in the Execution Analyzer window.

### Learn more

- [Set Up Simulation Script](#)<sup>[2472]</sup>
- [Simulation Breakpoints](#)<sup>[2477]</sup>

### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Simulation | Simulation | Run a Manual Simulation**
- (Alt+F1) | **Enterprise Architect | Simulation | Simulation | Run an Interpreted Simulation**
- (Alt+F1) | **Enterprise Architect | Simulation | Dynamic Simulation | Debugging Simulations**

## 15.7 Simulation Breakpoints

The **Simulation Breakpoints** tab of the **Breakpoints & Events** window enables you to **interrupt** and **inspect** the simulation process.



When **dynamically** executing a simulation, the process will proceed **automatically** (in the Corporate and above editions) - if you want to stop execution at some point to examine variables, inspect call stacks or otherwise interact with the simulator, you can set a **breakpoint** on a model element in much the same way as you would with a line of source code. When the simulator reaches the breakpoint execution is **halted** and control returned to Enterprise Architect.



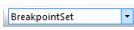

**Access** [Analyzer | Breakpoints & Events > Simulation Breakpoints](#)

### Breakpoints

Topic	Detail	See also
<b>Breakpoints</b>	<p>The simulation executes the model <b>step-by-step</b>, enabling you to validate the logic of your behavior model; the simulation halts when it reaches an element defined as a <b>breakpoint</b>.</p> <p>The UML elements that can be defined as breakpoints include: <i>Actions, Activities, States and most other behavioral nodes (decision, initial, final etc.)</i>.</p> <p>The UML relationships that can be defined as breakpoints include: <i>Interaction Messages</i>.</p> <p>The breakpoints are stored as <b>Breakpoint Sets</b> for a given Enterprise Architect project.</p> <p>Elements that are included in a simulation and that have breakpoints are marked by a <b>green circle</b> off the <b>top left corner</b> of the element, whilst the simulation is in progress. If the simulation is not running, the green circles are not displayed.</p> <p>When <b>Javascript</b> is enabled, all Simulation variables will be displayed in the <b>Locals</b> window - and it is possible to modify Simulation variables using the Simulation window's console input field (underneath the Toolbar).</p>	<a href="#">Run Model Simulation</a> <sup>[2475]</sup> <a href="#">Simulation Windows</a> <sup>[2469]</sup>

### Toolbar Options

Icon	Action	See also
	<b>Enables all breakpoints</b> defined in the current Breakpoint Set for the simulation session.	
	<b>Deletes all breakpoints</b> defined in the current Breakpoint Set for the simulation session.	

Icon	Action	See also
	<b>Disables all breakpoints</b> defined in the current Breakpoint Set for the simulation session.	
	<b>Adds a breakpoint</b> for the selected element or Sequence message to the current Breakpoint Set.	
	<b>Changes</b> the selected <b>Breakpoint Set</b> for use in the simulation session.	
	Performs Breakpoint Set commands: <ul style="list-style-type: none"> <li>• <b>New Set:</b> Create a new Breakpoint Set</li> <li>• <b>Save As Set:</b> Saves the current Breakpoint Set under a new name</li> <li>• <b>Delete Selected Set:</b> Deletes the current Breakpoint Set</li> <li>• <b>Delete All Sets:</b> Deletes all Breakpoint Sets saved for the diagram</li> </ul>	

#### Learn more

- [Simulation Windows](#) <sup>[2469]</sup>
- [Run Model Simulation](#) <sup>[2475]</sup>

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Simulation | Simulation | Using Breakpoints 1**
- (Alt+F1) | **Enterprise Architect | Simulation | Simulation | Using Breakpoints 2**
- (Alt+F1) | **Enterprise Architect | Simulation | Dynamic Simulation | Debugging Simulations**

## 15.8 Objects and Instances in Simulation

As a given business, system or mechanical process executes, the Activities and Actions within it might generate **objects** of a specific type and perform operations **on those objects**, perhaps even consuming or destroying them. You can simulate the creation, use and consumption of such objects using a simulation model that represents the objects and actions with model elements such as Classes, Instance Objects, attributes, operations and Ports (ActionPins and ObjectNodes). The model can also create, act on and destroy several different objects at different stages as part of the same process. Representing model data or objects in simulation makes the simulation more accurately reflect the real process.

### Object Concepts

Term	Description	See Also
<b>SimType</b>	The <b>type</b> of simulation element, such as Class, Enumeration or Interface. These can be classifiers of objects in a simulation.	
<b>SimObject</b>	An object that is an <b>instance</b> of (is classified by) a SimType element.	
<b>Attribute</b>	A <b>property</b> of a SimType element, or of a specified node such as an ActivityNode.	
<b>Operation</b>	A <b>behavior</b> of a SimType element, or of a specified node such as an ActivityNode.	
<b>Port</b>	A Port of a Class or Object, an ActionPin of an Action, or an ObjectNode of an Activity. Ports of classifiers are a <b>type</b> , whilst a Port of an object is a <b>realization</b> of the type.	
<b>Parameter / Activity Parameter</b>	<b>Parameters</b> of Operations; <b>Activity Parameters</b> are, specifically, parameters of ActivityNodes.	
<b>Slot</b>	A <b>realization</b> of an <b>attribute</b> in an object. A Slot has a <b>run time</b> value that can be initialized by the <b>run state</b> value of the Slot. If these values don't exist, the system uses the initial values of the attributes.	
<b>Runtime Environment</b>	All objects exist in the JavaScript runtime environment, so you can use JavaScript to create or change simulation objects and simulation variables.	
<b>Display Variables</b>	All simulation objects, simulation variables or events are identified on the Locals window while they are in effect. In some cases, to show the variables you might need to add break points to the model to pause processing while the variable exists.  As <b>all</b> objects and variables are shown, global variables that exist outside the	

Term	Description	See Also
	simulation but that are significant to it - such as the parent Class and Activity elements within which a process is defined - are automatically also represented as <b>default object variables</b> . So too is the anticipated output of the Activity, as a <b>return</b> variable.	

#### Learn more

- [Create Objects in a Simulation](#) <sup>[2480]</sup>
- [Destroy Objects in a Simulation](#) <sup>[2483]</sup>
- [Call Behaviors](#) <sup>[2489]</sup>

### 15.8.1 Create Objects in a Simulation

In a simulation model, you can create Classes and either create instances of them (Global Objects) to represent objects that exist in the process, or define Actions to generate one or more Objects at any point during the process.

You have three options for creating Objects in a simulation model:

- Manually create the Object
- Dynamically create an Object through a CreateObject Action element
- Use the JavaScript function `sim.CreateObject("name")` as the Effect of an Action element, to again create an Object dynamically

Having created an Object dynamically you can also instantiate any inner objects of that Object, such as an Activity on a Class, and act on the properties of that inner object.

#### Manually Create an Object

Simply create an Object element on a diagram in the model, either by:

- Dragging an Object element from the Object pages of the Diagram Toolbox and setting its classifier, or
- Dragging a classifier element from the Project Browser and pasting it into the diagram as an instance

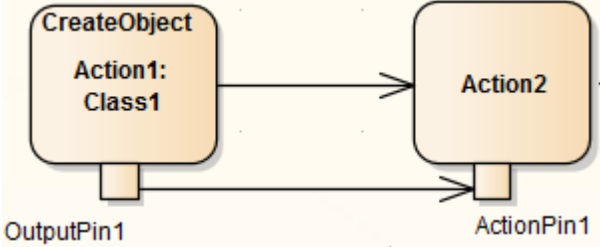
In the simulation model you can then set up the Object properties themselves (such as setting run-states to re-set the initial value of an attribute) or the behaviors of Actions to act on the Object (such as passing it along a process flow) and observe what happens to the Object in a simulation.

#### Create an Object through a CreateObject Action

If your process generates objects in runtime, you can simulate this using a **CreateObject Action**.

Step	Action	See also
1	On your Activity diagram, drag an Action icon from the Diagram Toolbox, and select the <b>Other   CreateObject</b> context menu option to define it as a	<a href="#">Action</a> <sup>[1266]</sup>



Step	Action	See also
	CreateObject Action element.	
2	<p>Set the classifier of the CreateObject Action to the Class of which the Object will be an instance.  <b>(Advanced   Set Classifier).</b></p> <p>Create an Action Pin on the CreateObject Action, of <b>kind</b> output.</p>	<a href="#">Action Pin</a> <sup>[1277]</sup>
3	<p>Create or select the next Action in the processing sequence, and add an Action Pin of <b>kind</b> input.</p> <p>Connect the two Actions with a Control Flow connector, and the two Action Pins with an Object Flow connector.</p> 	<a href="#">Control Flow</a> <sup>[1403]</sup> <a href="#">Object Flow</a> <sup>[1435]</sup>
4	<p>Perform a Simulation on the diagram. When the CreateObject Action is executed, it <b>creates an Object</b> having the properties of the classifier, and stores it in its Output Pin. The Object itself is <b>passed through the Object Flow connection</b> to the Input Pin of Action 2, where its properties can be listed in the Locals window for the simulation.</p>	

### Create Object Using JavaScript

You also can create simulation objects dynamically using a Javascript command in the **Effect** field of the Action element. The command is:

```
sim.newObject = sim.CreateObject( "ClassName" );
```

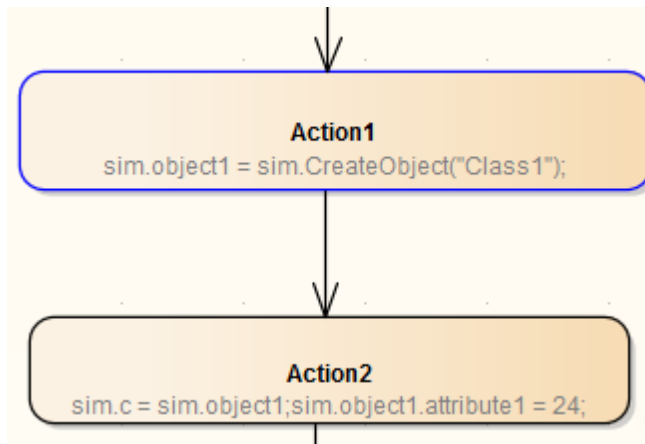
or

```
sim.newObject = new SimObject( "ClassName" ); (natural JavaScript)
```

That is: 'Simulate the creation of an Object based on Class <name>'. The classifying Class would exist in the same Package as the Action.

As for the CreateObject Action element, the Object is created during the simulation and can be passed down to and processed by 'downstream' elements. In this example, the created Object is identified as **sim.object1** and in **Action 2** it is accessed and one of its attributes given a different value (also by JavaScript as an Effect

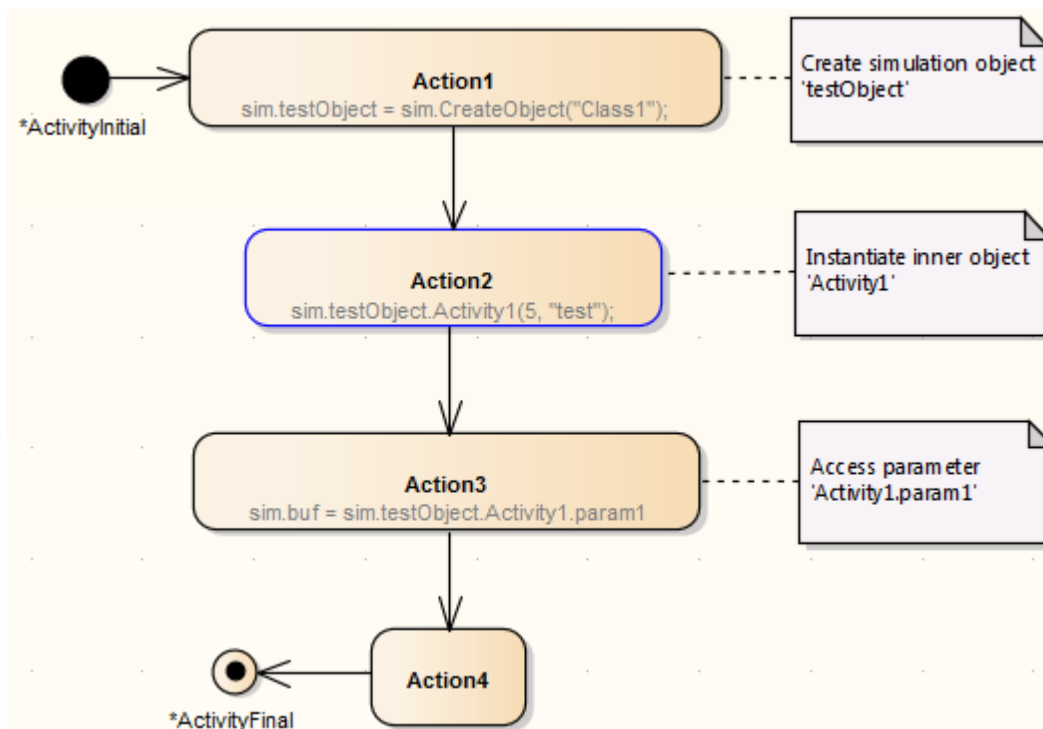
of the Action).



### Instantiate Inner Objects

As described above, you can create an Object using either JavaScript or a CreateObject Action. Similarly, you can **instantiate inner objects** using Javascript or a CallBehavior Action.

In this example, using **JavaScript**, the simulation first creates a test object based on Class1. Class 1 has an Activity element and diagram, with an Activity Parameter 1 set to the integer 5 and an Activity Parameter 2 set to the string "test". The value of Activity Parameter 1 is captured as a buffer value 'buf'



Learn more

- [Objects and Instances in Simulation](#)<sup>[2479]</sup>
- [Dynamic Simulation with Javascript](#)<sup>[2486]</sup>
- [Call Behaviors](#)<sup>[2489]</sup>
- [Action Types](#)<sup>[1271]</sup>

### 15.8.2 Destroy Objects in a Simulation

Having created or generated Objects in your simulation model, you can define Actions to **destroy** those objects at any point **during** the process. All simulation objects are destroyed automatically when the simulation **completes**.

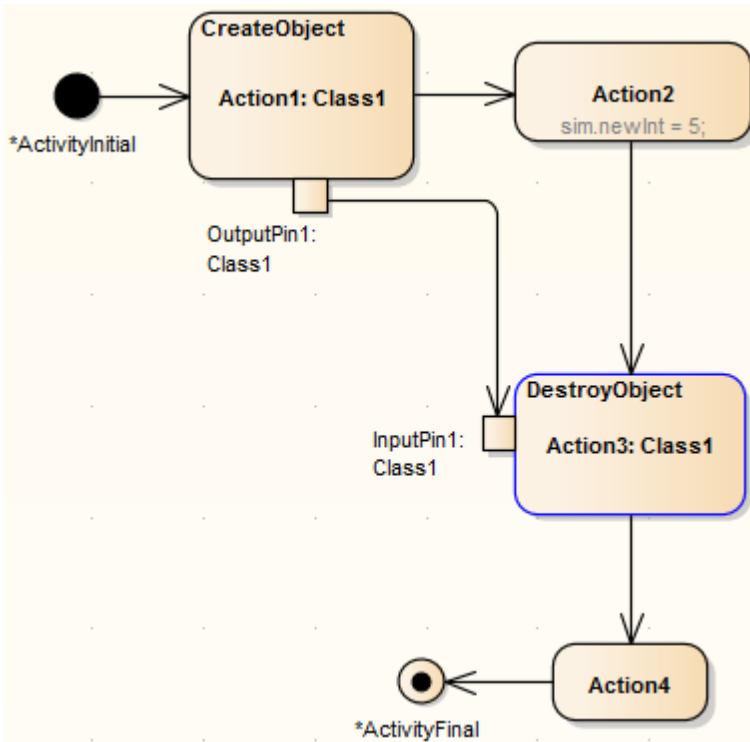
You have two options for destroying the Objects in your simulation model:

- Dynamically destroy the Objects through a **DestroyObject** Action element
- Dynamically destroy the Objects using **Javascript** in an Action element

The result of the deletion can be observed in the change of local variables, on the Local window.

Destroy an Object through a DestroyObject Action

Step	Action	See also
1	On your Activity diagram, drag an Action icon from the Diagram Toolbox, and select the <b>Other   DestroyObject</b> context menu option to define it as a DestroyObject Action element.	<a href="#">Action</a> <sup>[1266]</sup>
2	Set the classifier of the DestroyObject Action to the Class of which the Object is an instance. <b>(Advanced   Set Classifier)</b> . Create an Action Pin on the DestroyObject Action, of <b>kind</b> input.	<a href="#">Action Pin</a> <sup>[1277]</sup>
3	Connect the Input Action Pin to an Object Flow connector from the last Action that operated on the Object. In this example, the last Action <b>that operated on the Object</b> is the Action that created it.	<a href="#">Object Flow</a> <sup>[1435]</sup>

Step	Action	See also
	 <pre> graph TD     Start(( *ActivityInitial )) --&gt; Action1[Action1: Class1 CreateObject]     Action1 --&gt; Action2[Action2 sim.newInt = 5;]     Action1 -- OutputPin1: Class1 --&gt; Action3[Action3: Class1 DestroyObject]     Action2 --&gt; Action3     Action3 --&gt; Action4[Action4]     Action4 --&gt; End((( *ActivityFinal )))   </pre>	
4	<p>Perform a Simulation on the diagram. The process passes the Object name or value into the Input Action Pin as a parameter. When the DestroyObject Action is executed, it deletes the <b>Object</b> having that name or value from the model.</p> <p>In the example, the instance of Class1 is specifically destroyed before Action4 is processed, but the results of Action2 are unaffected.</p>	

#### Destroy an Object using Javascript

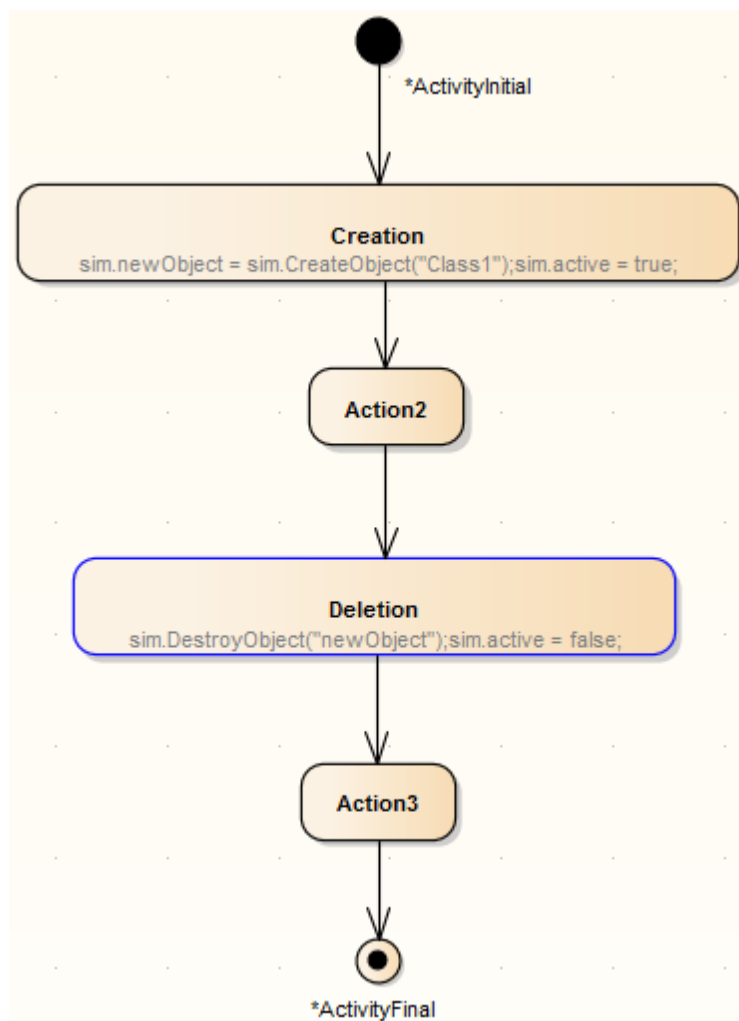
On the Properties Effect page of the Action element, in the **Effect** field, type either:

```
sim.DestroyObject ( "object name" )
```

or

```
delete sim.objectFullName
```

For example:



### Notes

- In either case, you can also destroy a **global** object (one that is created outside the process flow) by identifying the Object to the Action performing the destruction; in the case of the DestroyObject Action, by passing the Object name from a Port on the Object to the Input Pin on the Action through an Object Flow connector

### Learn more

- [View the Local Variables](#) 

## 15.9 Dynamic Simulation with Javascript

The Corporate and suite editions of Enterprise Architect have the capability of using **Javascript** to **evaluate guards, effects** and other aspects of **behavior** within the Simulation context. This provides for a **fully automated, intelligent execution** of your State or Activity model, with fine control over breakpoints, execution speed and simulation variables.

Javascript can be written that uses any variables. To enable you to display the variable values to the user interface two objects are defined that have their members shown in the **Local Variables** window. These are **sim** and **this**; for example:

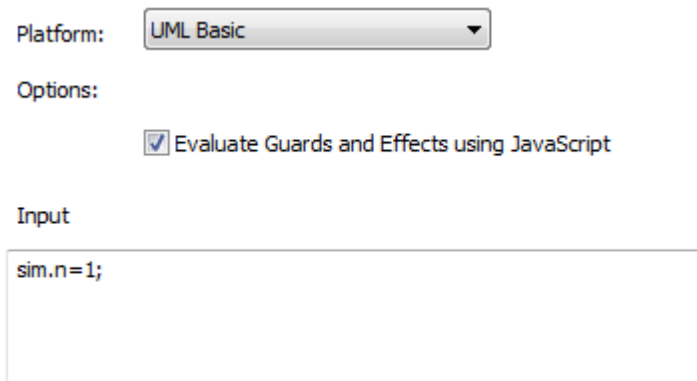
- `sim.logger`
- `sim.Customer.name`
- `this.count`
- `this.Account.amount`

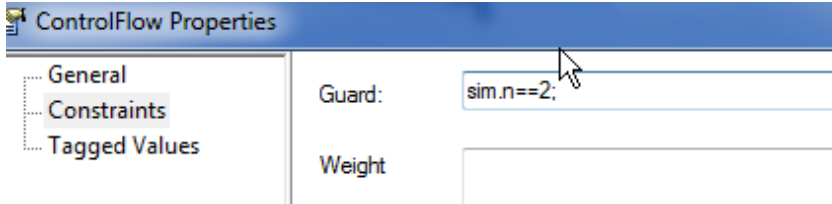
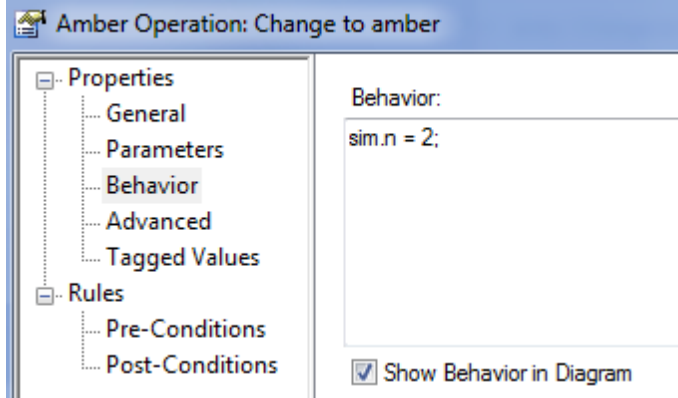
All the above variables will be shown in the **Locals** window.

The recommended convention is to add any global or control variables not declared in the owning Class to the **sim** object. In contrast it would be normal to add attributes of the owning classifier to the **this** object.

Some examples of where and how you can set Simulation behavior using Javascript are shown below. See the **EAExample.eap** model that comes with Enterprise Architect for some more examples. Also see the **Learning Center** for further information on setting up and working with Javascript in Simulations.

### Using Javascript

Setting	Action	See also
<b>Analyzer Script Input</b>	<p>If you enter <b>Javascript</b> code into the Execution Analyzer window <b>Input</b> field, this code will be <b>injected</b> into the Simulation and executed before the Simulation starts. This is useful for establishing COM variables, global counters, functions and other initialization.</p> 	
<b>Transition and Control Flow Guards</b>	<p>This is the <b>workhorse</b> of the Simulation functionality. As Enterprise Architect <b>evaluates</b> possible paths forward at each node in a Simulation, it tests the <b>Guards</b> on outgoing transitions and control flows and will only move forward if there is a <b>single true path</b> to follow - otherwise the Simulation is considered "blocked" and manual intervention is required. You must use the <b>"=="</b> operator to test for equality.</p>	

Setting	Action	See also
		
<b>Element Behavior</b>	<p><b>Entry and Exit behavior</b> may be defined for States. Such code will execute at the appropriate time and is useful for updating Simulation variables and making other assignments.</p>  <p>You can also simulate the behavior of Classes, via their Object Instances, and Activities in your model.</p>	<p><a href="#">Call Behaviors</a></p> <p>[2489]</p>
<b>Using COM</b>	<p>One very important feature of the implementation of Javascript in Enterprise Architect's simulator is that it supports the creation of <b>COM objects</b>. This provides the ability to <b>connect the running Simulation with almost any other local or remote process</b> and either influence the Simulation based on external data, or potentially change data or behavior in the external world based on the current Simulation state (eg. update a mechanical model or software simulation external to Enterprise Architect). <b>The syntax</b> for creating COM objects is shown below.</p> <pre> this.name=" Odd Even"; var logger = new COMObject ( " My Si m. Logger " ); logger.Show(); logger.Log( " Si mul a ti on s t a r t e d " ); </pre>	
<b>Signalled Actions</b>	<p>It is possible to raise a <b>signalled event (trigger)</b> directly using <b>Javascript</b>. The <b>BroadcastSignal()</b> command is used to raise a named trigger which may influence the current state of a simulation. For example the following fragment raises the signal (trigger) named "CancelPressed".</p> <pre> Broadcast Signal ( " Cancel P r e s s e d " ); </pre>	

Setting	Action	See also
	<p>Note that a trigger named <b>CancelPressed</b> must exist within the current simulation environment and must <b>uniquely have that name</b>.</p> <p>You can also call the signal using its GUID. For example:</p> <pre>Broadcast Signal ( "{ 996EAF52- 6843- 41f 7- 8966- BCAA0ABEC41F} " );</pre>	
<b>IS_IN()</b>	<p>The <b>IS_IN(state)</b> operator returns true if the current simulation has an <b>active state</b> in any thread <b>matching</b> the passed in <b>name</b>. For example it is possible to write code such as the following to conditionally control execution:</p> <pre>if ( IS_IN( " WaitingForInput " ) )     Broadcast Signal ( " Cancel Pressed " )</pre> <p>Note that the name must be <b>unique</b> within all contexts.</p>	
<b>Trace()</b>	<p>The Trace(statement) method allows you to print out trace statements at any arbitrary point in your simulation. This is an excellent means of <b>supplementing</b> the <b>Simulation log</b> with additional output information during execution.</p>	

#### Learn more

- [Interaction Operand Condition and Message Behavior](#) <sup>[2490]</sup>
- [Guards and Effects](#) <sup>[2492]</sup>
- [Triggers](#) <sup>[2494]</sup>

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Simulation | Dynamic Simulation |**
  - **Activity With JavaScript 1**
  - **Activity With JavaScript 2**
  - **State Machine With JavaScript 1**
  - **State Machine With JavaScript 2**
  - **Sequence With JavaScript 1**
  - **Sequence With JavaScript 2**
  - **Sequence With JavaScript 3**



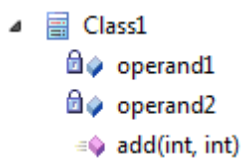
### 15.9.1 Call Behaviors

In the course of simulating a process, you can enact the behaviors defined in an operation of either a Class (through its simulation Object) or an Activity in the model. In each case, you use JavaScript to call the behaviour.

#### Invoke the Behavior of a Class

A Class in your model defines a behavior that you want to simulate. This behavior is defined in the Behavior page of an Operation of the Class.

For example, the Class is intended to add two integers, through the Operation **add**. The integers in this case are parameters of the operation, defined by attributes of the Class, **operand1** and **operand2**.



Step	Action	See also
1	<p>In the Properties dialog for the Operation, select the Behavior page and edit the <b>Behavior</b> field to apply the JavaScript simulation objects (<b>this</b> or <b>sim</b>) to the behavior definition.</p> <p>In the example:</p> <pre> this.operand1=operand1; this.operand2=operand2; return operand1+operand2           </pre>	<a href="#">Dynamic Simulation with Javascript<sup>[2486]</sup></a>
2	<p>Drag the Class onto your simulation Activity diagram and paste it as an <b>Instance</b>.</p> <p>In the example, the Object is called <b>calculator</b>. For clarity, the element shown here is set to display inherited attributes and operations, and the behavior code, on the diagram.</p> <pre> classDiagram     class calculator {         &lt;&lt;instance&gt;&gt;         ..Class1         - operand1: int         - operand2: int         ..Class1         + add(int, int): int         this.operand1= operand1;         this.operand2 = operand2;         return operand1+ operand2;     }   </pre>	
3	<p>On the simulation diagram, for the appropriate Action element, open the Properties dialog and on the Effect page type in the JavaScript to capture and</p>	

Step	Action	See also
	<p>simulate the Object's behavior.</p> <p>In the example, the JavaScript defines a value that will be provided by simulating the behavior of the operation from the Object, as performed on two provided integers. That is:</p> <pre>sim.result=sim.calculator.add(7,9)</pre>	
4	<p>Run the simulation, and observe its progress in the Locals window. Ultimately the Class behavior is reflected in the result of the simulation.</p> <p>In the example: <code>result = 16</code>.</p>	

#### Invoke the Behavior of an Activity

An Activity element can have a behavior, defined by an operation in that element. As a simple example, an Activity might have an operation called **Get Result**, with the behavior `return "ON"` ; .

You can simulate this behavior in the Activity's child diagram (that is, internal to the Activity), with a JavaScript statement in the appropriate Action element's **Effect** field. In the example, this might be:

```
sim.result=this.GetResult();
```

The statement invokes the parent Activity's operation **GetResult** and assigns the outcome of that operation's behavior to **sim.result**. You can observe the progress of the simulation and the result of simulating the behavior in the Locals window, where (in this example) the value **result "ON"** will ultimately display.

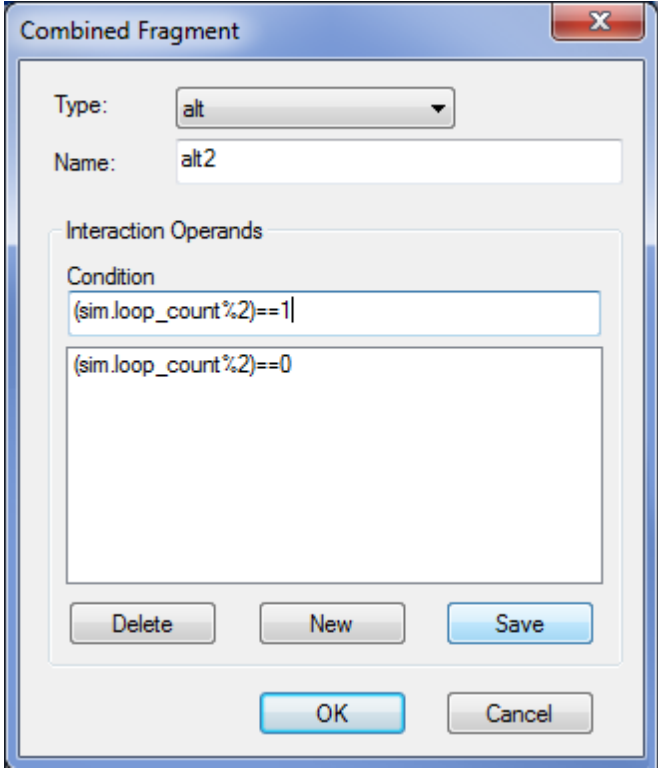
### 15.9.2 Interaction Operand Condition and Message Behavior

When you simulate the behavior of a Sequence diagram, you can use a Condition for the *CombinedFragment* Interaction Operand, to control the flow during the course of the simulation.

A Message in Sequence diagram can link to an Operation, so the behavior of the Operation can also be used during the course of the simulation.

#### Interaction Operand Conditions

Field/Column	Description	See also
<b>Operand Condition</b>	<p><b>Interaction Operand Conditions</b> are conditional statements that are evaluated whenever the simulator has to determine which path to take next. Operand Conditions typically have the following characteristics:</p> <ul style="list-style-type: none"> <li>• Defined in the Combined Fragment dialog</li> <li>• Written in <b>Javascript</b></li> <li>• Can refer to variables defined during simulation</li> </ul>	
<b>Adding</b>	To add an Operand <b>Condition</b> :	

Field/Column	Description	See also
<b>Operand Conditions</b>	<ol style="list-style-type: none"> <li>1. Double-click on the CombinedFragment element to open the Combined Fragment dialog.</li> <li>2. Click on the <b>New</b> button.</li> <li>3. In the <b>Condition</b> field, type the Javascript for the condition.</li> <li>4. Click on the <b>Save</b> button.</li> </ol> 	
<b>Evaluation Semantics</b>	During execution the Simulator evaluates any Operand Condition within the CombinedFragment; the CombinedFragment type and the outcome of the evaluation can determine the path that the simulation continues on.	

#### Learn more

- [Dynamic Simulation with Javascript](#)<sup>[2486]</sup>
- [Combined Fragment](#)<sup>[1287]</sup>
- [Interaction Operators](#)<sup>[1290]</sup>
- [Guards and Effects](#)<sup>[2492]</sup>
- [Triggers](#)<sup>[2494]</sup>

### 15.9.3 Guards and Effects

**Guards and Effects** are used to **control the flow** of the simulation and to **execute additional actions** or effects during the course of a simulation.

#### Guards and Effects

Concept	Detail	See also
<b>Guards</b>	<p>Guards are <b>conditional statements</b> that are evaluated anytime the simulator needs to decide which path to take next. Guards typically have the following characteristics:</p> <ul style="list-style-type: none"> <li>Defined on <b>transitions</b> and <b>control flows</b> to govern how simulation proceeds</li> <li>Written in <b>Javascript</b></li> <li>May refer to <b>variables</b> defined during simulation</li> </ul>	<a href="#">Dynamic Simulation with Javascript</a> <sup>[2486]</sup>
<b>Adding Guards</b>	<p>Guards are defined on the <b>Transition</b> or <b>Control Flow</b> in the Properties dialog for the selected connector. A Guard is typically a piece of Javascript that will evaluate to either <b>true</b> or <b>false</b>. For example, the following is a conditional statement that refers to a current variable (Balance) being greater than zero. Note the use of the prefix <i>this</i> to indicate that the variable is a member of the current Class context.</p> <div> <p>Guard: <input type="text" value="this.Balance &gt; 0;"/></p> <p>Effect: <input type="checkbox"/> Effect is a Behavior</p> </div>	
<b>Evaluation Semantics</b>	<p>During execution the Simulator will examine all possible paths forward and <b>evaluate</b> any <b>guard conditions</b>. Based on this evaluation the following can occur:</p> <ul style="list-style-type: none"> <li><b>A single valid path</b> forward evaluates to <b>true</b>; the simulator will follow that path</li> <li><b>Two valid paths</b> are found; the simulator will <b>block</b>, waiting for some manual input via the console window to resolve the deadlock</li> <li><b>No valid path exists</b>; the same response as when two paths are found - wait for the user to change the execution context using the console window</li> <li>No paths evaluate to true but a <b>default (unguarded path)</b> exists; the Simulator will take the <b>unguarded path</b></li> </ul>	
<b>Effects</b>	<p>Effects are defined behaviors that are executed at special times:</p> <ul style="list-style-type: none"> <li>On <b>entry</b> to a state</li> <li>On <b>exit</b> from a state</li> <li>When <b>transitioning</b> from one state to another (transition <b>effect</b>)</li> </ul>	

Concept	Detail	See also
	<p>Effects can either be a section of Javascript code or a call to another Behavior element in the current simulation.</p>	
<b>Javascript Effects</b>	<p>A Javascript effect might resemble this example, in which the Balance variable is decremented:</p> <p>Guard: <input type="text"/></p> <p><input type="checkbox"/> Effect is a Behavior</p> <p>Effect: <input type="text" value="this.Balance--;"/> <input type="button" value="..."/></p>	
<b>Call Behavior Effects</b>	<p>In this example the effect is a <b>call behavior</b> effect. In this case, it calls into a model the Activity named <i>Decrement Balance</i> that is defined elsewhere. The simulation will then <b>enter into that diagram/behavior</b> and continue to execute until returning to the point at which the effect was invoked.</p> <p>Guard: <input type="text"/></p> <p><input checked="" type="checkbox"/> Effect is a Behavior</p> <p>Effect: <input type="text" value="Decrement Balance"/> <input type="button" value="..."/></p>	
<b>Order of Execution of Effects</b>	<p>In <b>complex simulations</b> that might involve transitioning out of <b>deeply nested states</b> into other deeply nested states in a different parent context, it is important to consider the following rules concerning the order of execution:</p> <ul style="list-style-type: none"> <li>• <b>All exit actions (effects) encountered leaving a nested context</b> are executed in order of most deeply nested to least deeply nested</li> <li>• All <b>actions (effects)</b> defined on <b>transitions</b> are executed next</li> <li>• <b>Finally</b>, all <b>entry effects are executed</b> from the least deeply nested context to the most deeply nested</li> </ul> <p>So the basic rule is: all exit actions, followed by all transition actions, and finally all entry actions.</p>	
<b>Note on Javascript Variables</b>	<p><b>Javascript variables</b> to be accessed and referred to during Simulation execution belong to either:</p> <ul style="list-style-type: none"> <li>• <b>sim</b> e.g. <code>sim.pedestrianwaiting</code> - typically used for global simulation variables, or</li> <li>• <b>this</b> e.g. <code>this.CustomerNumber</code> - typically used to refer to owning Class attributes</li> </ul> <p>This is important to let the Javascript engine know you are referring to</p>	

Concept	Detail	See also
	a <b>Simulation variable</b> and not a simple local variable used during, for example, basic calculations. You can create Simulation variables of arbitrary scope and depth - for example, <i>this.customer.name</i> is a legitimate qualified name.	

#### Learn more

- [Interaction Operand Condition and Message Behavior](#)  <sup>[2490]</sup>
- [Triggers](#)  <sup>[2494]</sup>

### 15.9.4 Triggers




**Triggers represent signals** and events that can **activate transitions** leaving the current state(s). A trigger may represent a **real world signal** or event such as:

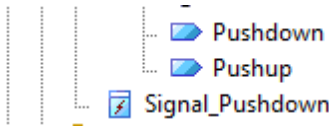
- A **button** being pressed
- A **message** being received
- A **pedal** being depressed
- A **switch** being thrown
- A **state** in a concurrent region being entered or exited









#### For a trigger to have an effect

- Transitions have to be defined which **will fire** when the simulation **receives** the **signal/event**
- The current simulation state(s) or its parent(s) must have an **outgoing transition** that **accepts** that **trigger**
- The transition activated must be **unguarded** or have a **guard** which will **evaluate** to **true**

#### Managing Triggers

Action	Detail	See also
<b>Creating Triggers</b>	<p>Triggers are <b>either</b> created as an <b>instance of a Signal</b> element or as an <b>anonymous event</b>. Triggers are connected to Transitions in the Transition Properties dialog as shown below. In this example a Trigger named 'Pushdown' has been defined based on the Signal 'Signal_Pushdown'.</p> <ul style="list-style-type: none"> <li>• <b>Omitting the Type and Specification</b> details results in a simple anonymous Trigger.</li> <li>• If <b>parameters</b> are needed, these are <b>defined on the Signal</b> and must be supplied at the time the event fires</li> </ul>	<a href="#">Trigger</a>  <sup>[1350]</sup> <a href="#">Transition</a>  <sup>[1446]</sup> <a href="#">Simulation Events Window</a>  <sup>[2504]</sup>

Action	Detail	See also						
	<p>Triggers</p> <p>Name: <input type="text" value="Pushdown"/> ...</p> <p>Type: <input type="text" value="Signal"/> Specification: <input type="text" value="Signal_Pushdown"/> ...</p> <p><input type="button" value="New"/> <input type="button" value="Save"/> <input type="button" value="Delete"/></p> <table border="1"> <thead> <tr> <th>Name</th><th>Type</th><th>Specification</th></tr> </thead> <tbody> <tr> <td>Pushdown</td><td>Signal</td><td>Signal_Pushdown</td></tr> </tbody> </table> <p>A trigger will appear in the <b>Project Browser</b> like the image below:</p> 	Name	Type	Specification	Pushdown	Signal	Signal_Pushdown	
Name	Type	Specification						
Pushdown	Signal	Signal_Pushdown						
<b>Using Triggers</b>	<p>Triggers are <b>deployed by connecting</b> them to transitions as in the example above and used during simulation by 'firing' them into the running simulation as required.</p> <p>When using triggers the following should be taken into account:</p> <ul style="list-style-type: none"> <li>• A 'triggered' transition can not take place until its <b>effective trigger is signalled</b> or fired</li> <li>• When a trigger is received it will <b>activate all current waiting transitions dependent on that trigger</b> (i.e. the trigger is broadcast)</li> <li>• Triggers are evaluated on <b>all transitions for all parents of a current child state</b>. This allows a parent state to exit all child states if necessary</li> <li>• Once used in a simulation, a trigger is <b>consumed</b> and must be re-fired if needed again</li> <li>• <b>Sets of triggers</b> may be saved and either manually or automatically fired to facilitate automated model simulation under different event models</li> </ul>	<p><a href="#">Re-Signal Triggers</a> <sup>[2509]</sup></p> <p><a href="#">Connector Type-Specific Options</a> <sup>[1103]</sup></p>						
<b>Firing Triggers</b>	<p><b>Firing triggers</b> means to <b>signal</b> or <b>activate</b> a trigger within the current simulation. This may activate zero, one or many waiting transitions depending on the state and concurrency of the current simulation.</p> <p>Firing triggers can be achieved in many ways. The most efficient is the <b>Waiting Triggers</b> list.</p> <p>During the course of model simulation, if the simulator reaches an <b>impasse</b> due to required triggers not being available (fired), the <b>list of all possible candidate triggers</b> is shown in the 'Waiting Triggers' list of the Simulation Events window.</p>	<p><a href="#">Simulation Events Window</a> <sup>[2504]</sup></p> <p><a href="#">Connector Type-Specific Options</a> <sup>[1103]</sup></p>						

Action	Detail	See also															
	<div><div>Waiting Triggers</div><div> Hold</div><div> Pushdown</div></div> <p><b>Double clicking a trigger</b> in this list will fire it into the simulation. Other ways to fire a trigger include:</p> <ol style="list-style-type: none"><li>1. <b>Double click an un-signalled trigger</b> in the Events window</li></ol> <table><thead><tr><th>Sequence</th><th>Trigger</th><th>Status</th><th>Type</th><th>Parameters</th></tr></thead><tbody><tr><td></td><td>Pushdown</td><td>not signalled</td><td>Signal</td><td></td></tr><tr><td></td><td>Pushup</td><td>not signalled</td><td>Simple</td><td></td></tr></tbody></table> <p>You can also use the context menu on the events above to either signal an un-signalled event, or to re-signal an event which has already been fired previously.</p> <ol style="list-style-type: none"><li>2. Use the <b>context menu</b> of the <b>Transition required</b> to fire and select the <b>Signal Trigger in Simulation</b> menu option.</li></ol>	Sequence	Trigger	Status	Type	Parameters		Pushdown	not signalled	Signal			Pushup	not signalled	Simple		
Sequence	Trigger	Status	Type	Parameters													
	Pushdown	not signalled	Signal														
	Pushup	not signalled	Simple														

#### Learn more

- [Dynamic Simulation with Javascript](#)<sup>[2486]</sup>
- [Guards and Effects](#)<sup>[2492]</sup>
- [Interaction Operand Condition and Message Behavior](#)<sup>[2490]</sup>

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Simulation | Simulation | State Machine Table**
- (Alt+F1) | **Enterprise Architect | Simulation | Triggers | Using Triggers in a State Machine**





## 15.10 Action Behavior By Type

You can vary the behavior initiated by an Action element by defining (or even redefining) its type. In simulation, you can apply and observe a number of different behaviors using the Actions in the types and groups described in this table.

Action Type	Description	See also
<b>Object Actions</b>	Object Actions operate on an object in a specific way, such as creating, destroying or reading the object. They include: <ul style="list-style-type: none"> <li>• <b>CreateObject</b></li> <li>• <b>DestroyObject</b> and</li> <li>• <b>Read Self</b></li> </ul>	<a href="#">Create Objects in a Simulation</a> <small>[2480]</small> <a href="#">Destroy Objects in a Simulation</a> <small>[2483]</small> <a href="#">Action Types</a> <small>[1272]</small>
<b>Variable Actions</b>	Variable Actions have an association variable in the form of the Tagged Value variable with the value of the name of an object in run-time. They provide the variable not only as an object but also as a property (such as an attribute or Port) of an object. They include: <ul style="list-style-type: none"> <li>• <b>ReadVariable</b></li> <li>• <b>WriteVariable</b></li> <li>• <b>ClearVariable</b></li> <li>• <b>AddVariableValue</b></li> <li>• <b>RemoveVariable</b></li> </ul>	<a href="#">Variable Actions</a> <small>[1273]</small>
<b>StructuralFeature Actions</b>	StructuralFeature Actions operate on a structural feature, namely an attribute of an Activity or of the classifier of an object. They include: <ul style="list-style-type: none"> <li>• <b>ReadStructuralFeature</b></li> <li>• <b>WriteStructuralFeature</b></li> <li>• <b>ClearStructuralFeature</b></li> <li>• <b>AddStructuralFeatureValue</b></li> <li>• <b>RemoveStructuralFeatureValue</b></li> </ul>	<a href="#">StructuralFeature Actions</a> <small>[1271]</small>
<b>Invocation and Accept Event Actions</b>	Invocation and Accept Event actions define the Triggers and Signals of an event. They include: <ul style="list-style-type: none"> <li>• <b>SendSignal</b></li> <li>• <b>BroadcastSignal</b></li> <li>• <b>AcceptEvent</b></li> <li>• <b>SendObject</b></li> <li>• <b>CallBehavior</b></li> <li>• <b>CallOperation</b></li> <li>• <b>AcceptCall</b></li> </ul>	<a href="#">SendSignal Action &amp; BroadcastSignal Action</a> <small>[1270]</small> <a href="#">AcceptEvent Actions</a> <small>[1269]</small> <a href="#">CallBehavior Action</a> <small>[1271]</small> <a href="#">SendObject Action</a> <small>[1271]</small>

Action Type	Description	See also
<b>Miscellaneous Actions</b>	The <b>ValueSpecificationAction</b> evaluates a value; it must have an input value and some evaluating code as its behavior or effect.	

**Learn more**

- [Model Simulation](#)  <sup>[2463]</sup>
- [Objects and Instances in Simulation](#)  <sup>[2479]</sup>

## 15.11 Structured Activity Simulation

One of the more complex structures in a behavioral model is a **Structured Activity**, which models a series of actions either in a nested structure or in a process of assessment and execution. The assessment types of Structured Activity are the **Conditional Node** and **Loop Node**, both of which you can simulate quite easily.

### Conditional Node

A Conditional Node essentially consists of one or more pairs of **Test/Body** partitions, each pair being referred to as a **Clause**. The Test partition is composed of Activity diagram elements that test a condition, and if that condition is met further Activity diagram elements in the Body partition are executed to produce a result.

If there is one Clause, the Conditional Node either outputs the result of the Body partition, or no result. If there is more than one Clause, control flows from one Test to the next until either a condition is met and a Body partition is executed to produce a result, or all Tests fail.

Simulation currently supports use of the **Is Assured** checkbox setting in the Condition page of the Properties dialog. The other two checkbox settings are ignored. If the **Is Assured** checkbox is:

- **Selected**, at least one Test must be satisfied, so its Body is executed and a result output; if no Test is satisfied and no result output, the Conditional Node is blocked and processing cannot continue beyond it
- **Not selected**, a Test can be satisfied and a result output, but if no Test is satisfied and no result output, processing can still continue beyond the Condition Node

You can simulate a range of possible paths and outcomes by typing JavaScript **sim.** statements that define or lead to specific Test results and Body results, in the **Effect** fields of the Action elements within each partition of each Clause. These **sim.** statements must identify the full path of the Conditional Node, Clause and Output Pin being set. For example, in a **test** to see if a person qualifies as a senior citizen:

```
if (sim.Person.age >=65)
sim.AgeCondition.Clause1.Decider1=true;
else
sim.AgeCondition.Clause1.Decider1=false;
```

The Condition Node is called **AgeCondition**, the test is in **Clause1** and the OutputPin for that test is **Decider1**.

### Loop Node

A Loop Structured Activity Node commonly represents the modeling equivalents of **While**, **Repeat** and **For** loop statements. Each Loop Node has three partitions:

- **Setup** - which initiates variables to be used in the loop's exit-condition; it is executed once on entry to the loop
- **Test** - which defines the loop exit-condition
- **Body** - which is executed **repeatedly** until the Test produces a false value

You define the Loop Nodes by dragging Activity diagram elements from the Toolbox pages into the Setup, Test and Body partitions. The Body partition can contain quite complex element structures, defining what the Loop Node actually produces in the process.

The Loop Node has a number of Action Pins:

- **Loop Variable (Input)** - the initial value to be processed through the Loop
- **Loop Variable (Output)** - the changing variable on which the Test is performed
- **Decider** - an Output Pin within the **Test** partition, the value of which is examined after each execution of the Test to determine whether to execute the loop Body
- **Body Output** - the output value of the processing in the Body partition, which updates the Loop Variable Output Pin for the next iteration of the loop, and
- **Result** - the value of the final execution of the Test partition (which is the value passed back from the last execution of the Body partition)

You can simulate the effects of different actions and outputs through the loop, by typing JavaScript **sim.** statements that define or lead to specific Test results and Body results, in the **Effect** fields of the Action elements within each partition. These sim. statements must identify the path of the Loop Node and Output Pin being set. For example, in an Action in the Test partition:

```
sim.LoopNode1.decider = (sim.LoopNode1.loopVariable>0);
```

#### Learn more

- [Conditional Node](#)<sup>[1345]</sup>
- [Loop Node](#)<sup>[1347]</sup>
- [Dynamic Simulation with Javascript](#)<sup>[2486]</sup>

## 15.12 Activity Return Value Simulation

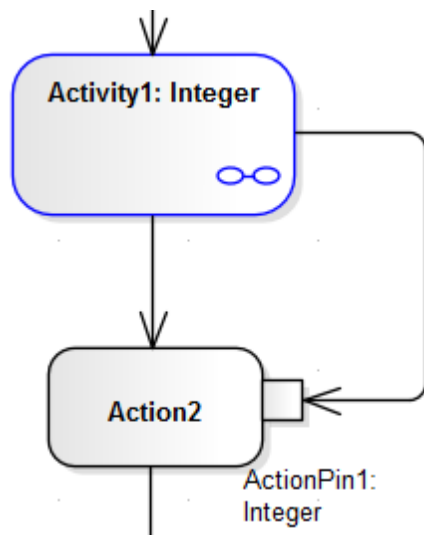
An **Activity** is likely to produce a **return value** as the output of the process it represents. You can simulate how that return value is passed on to the next stage in the process, under three scenarios:.

- The Activity simply produces a return value, which is passed directly to the next Action
- The Activity has one or more Activity Parameters - represented on a diagram by Activity Nodes - that accept input values to or hold output values from the child Actions of the Activity, and the output Activity Parameter collects and passes on the return value
- The Activity is instantiated by a CallBehavior Action that replicates the behavior of the Activity and passes the return value onwards

### Activity Return Value Pass Out

(This method is unique to Enterprise Architect simulation, mimicking the effect of an Activity Parameter without one having to exist.)

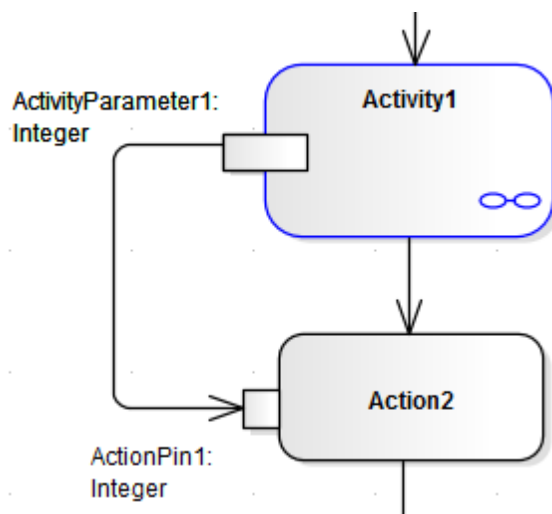
The Activity has a return value, which is transferred from the Activity element to an Action Pin on the next Action in the process via an **Object Flow** connector.



You can simulate this by setting a simple JavaScript statement to set the return value in the Activity's child element (such as `this.return=12;`) and, running the simulation, see the value transferred to the Action Pin in the Locals window.

### Activity Parameter Pass Out

If the Activity has an Activity Parameter, its value passes to the corresponding ActivityNode and then, via an Object Flow connector, to the Input ActionPin of the next Action in the process, as shown:

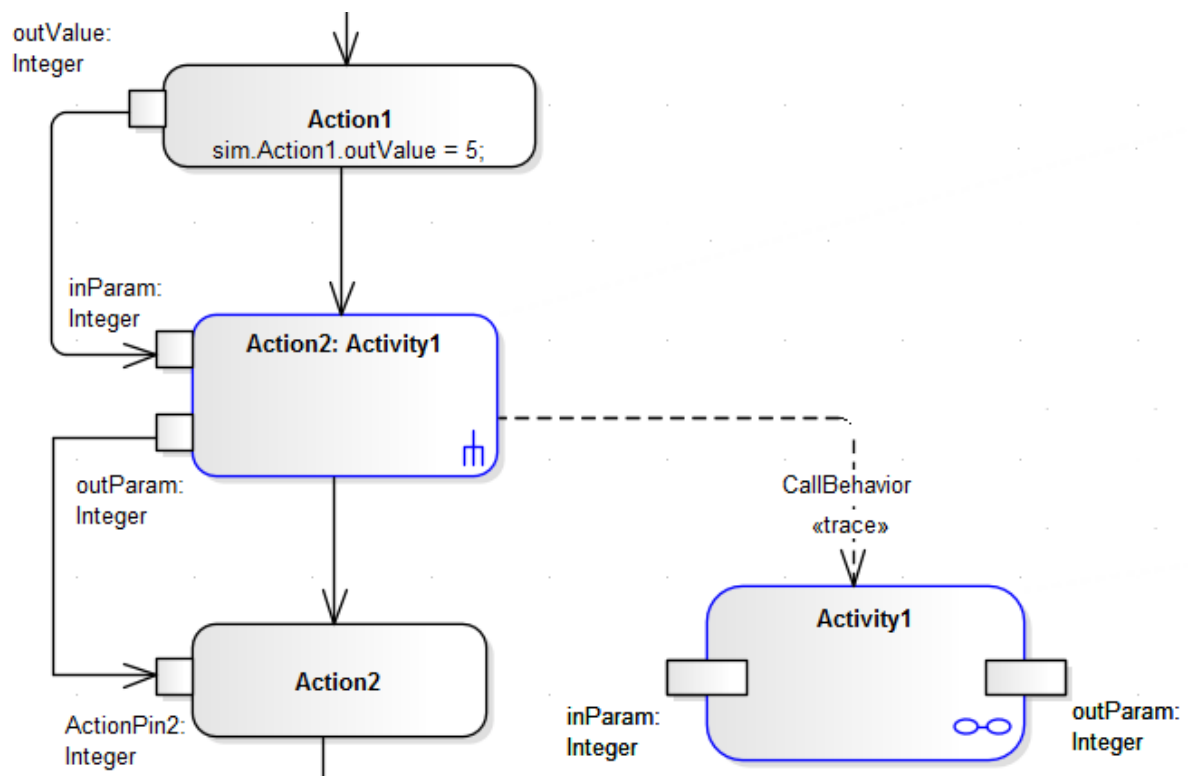


In the Locals window, you can either observe the Parameter's default value pass through to the ActionPin, or you can use JavaScript in the Activity's child Actions to simulate an update of the value within the Activity. For example:

```
this.ActivityParameter1=20;
```

#### CallBehavior Action

An Activity might be used a number of times in a process, in which case you might want to generate a separate instance of the Activity each time. You can do this using a **CallBehavior Action** to create an **object** of the Activity and execute its behavior. The input and output **Activity Parameters** are bound to corresponding input and output **Action Pins (arguments)** on the CallBehavior Action.



When you simulate the part of the process containing the Activity, you provide an input value (as in Action 1) that passes into the input Action Pin on the CallBehavior Action, which creates an Object of the Activity. The CallBehavior executes the behavior of the Activity, using the input Action Pin to act as the input Activity Parameter, and the Output Action Pin to receive the return as the output Activity Parameter. The Activity return value is then passed to an Action Pin on the next Action, as above, using an Object Flow connector. You can provide JavaScript statements in the Actions of the Activity to act on the input value and generate a return value, such as:

```

sim.buf=this.inParam;      and

this.outParam=sim.buf + 11;

```

### Learn more

- [Action Pin](#) <sup>[1277]</sup>
- [Object Flows in Activity Diagrams](#) <sup>[1436]</sup>
- [Activity Parameter Nodes](#) <sup>[1281]</sup>
- [CallBehavior Action](#) <sup>[1271]</sup>
- [Call Behaviors](#) <sup>[2489]</sup>

## 15.13 Simulation Events Window

The **Simulation Events** window is where you **manage triggers** and sets of events in a simulation. Its main functions are to:

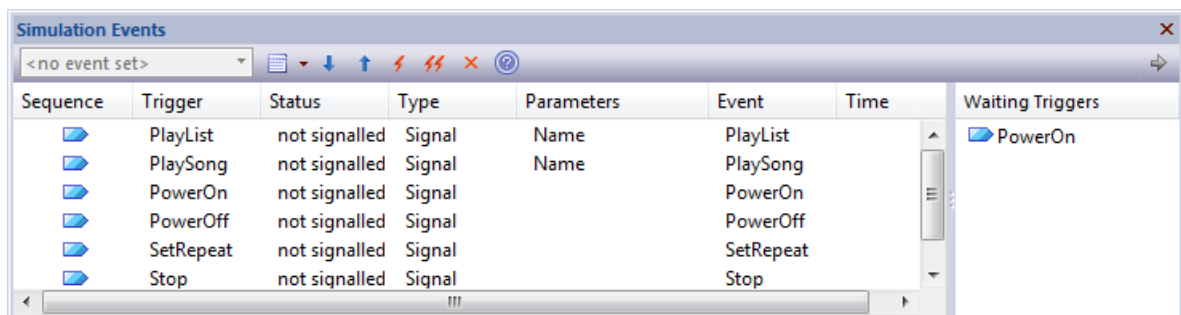
- **Add, delete and re-sequence** a set of triggers for a simulation
- **Display a list of fired, lost and waiting** events for the current running simulation
- Provide options to **fire any arbitrary trigger** into the current simulation
- Provides a convenient '**Waiting Triggers**' list of triggers that the simulation is waiting on
- Ability to **save trigger sets** for later use in both manual and automated simulations
- **Accept triggers dragged** from the Project Browser into the current list
- Enter **trigger parameters** for a waiting trigger prior to firing.

As **triggers are consumed** in the simulation, their status and position is logged in the main body of the Simulation Events window.

You can **save the log of fired triggers** as a trigger set or event set to reapply in another Simulation run, which you can execute manually or automatically. See the topic *Trigger Sets and Auto-Firing* for more information on building and using Trigger sets.

**Access** [Analyzer | Simulation Events](#)




The following image illustrates the Simulation Events window during execution.










### Column Details

Field/Column	Action	See also
<b>Sequence</b>	During and after the simulation, indicates the position in the sequence in which a trigger was fired or is expected to be fired. Note that if a trigger is fired out of sequence, it will be moved to the bottom of the signalled events section.	
<b>Trigger</b>	The name of the trigger - identifies the Trigger used to initiate the event.	<a href="#">Triggers</a> <sup>2494</sup>
<b>Status</b>	Indicates the status of the Trigger. Values can be:	



Field/Column	Action	See also
	<ul style="list-style-type: none"> <li><b>used</b> - the trigger has been fired and processing has passed on</li> <li><b>lost</b> - the trigger has been fired in the list, but it had no effect</li> <li><b>signalled</b> - a trigger was fired and consumed by one or more transitions</li> <li><b>not signalled</b> - the trigger has not yet been fired</li> </ul>	
<b>Type</b>	<p>Indicates the type of trigger. Currently only the following are supported:</p> <ul style="list-style-type: none"> <li><b>Signal</b></li> <li><b>(no type) an anonymous trigger</b></li> </ul>	<a href="#">Trigger</a> 
<b>Parameters</b>	<p>For a Signal Trigger, initially shows the parameters required for firing by the Signal specification. For example a "Login" signal might include username and password parameters - and each triggered invocation can use different parameters.</p> <p>Each time the simulation fires the trigger, the system will prompt you for values. You can also edit the values directly in the list when the trigger is set to <b>not signalled</b>.</p> <p>Parameters are very useful for testing the conditional logic in your simulation and to simulate a variety of inputs and data coming in from outside the simulation.</p>	<a href="#">Trigger Parameters</a> 
<b>Event</b>	<p>For a:</p> <ul style="list-style-type: none"> <li>Signal Trigger, identifies the Signal specification.</li> <li>For anonymous Triggers has no value</li> </ul>	
<b>Time</b>	<p>The simulation time at which the trigger was signalled. Note that this is an absolute (real world) time, and not a relative simulation event time.</p>	
<b>Waiting Triggers</b>	<p>Lists the Triggers available for selection from the current state(s), including those where more than one trigger is possible at a single transition. Double-click on a trigger to add and signal it as the next trigger in the current event sequence.</p> <p>You can show and hide this panel by clicking on the gray arrow just above the panel.</p>	<a href="#">Waiting Triggers</a> 



Toolbar Options

Option	Action	See also
	Use this drop list to select and work with previously defined trigger sets.  Before running a simulation, select a previously-defined trigger set to use for the next simulation run. You elect to not use a trigger set by selecting the <b>&lt;no event set&gt;</b> option.	
	Click to create and delete trigger sets: <ul style="list-style-type: none"> <li>• <b>Save Set</b> - Save the current trigger list as a new trigger set; the system prompts you for a name for the new set</li> <li>• <b>Save Set As</b> - Create a copy of the current set under a new set name</li> <li>• <b>Delete Selected Set</b> - Delete the current trigger set</li> <li>• <b>Delete All Sets for Diagram</b> - Delete all saved trigger sets for the current diagram</li> </ul>	<a href="#">Trigger Sets and Auto-Firing</a> [2512]
	Move the selected trigger one line down in the firing sequence of triggers.  This option is not available if there are no <b>not signalled</b> triggers below the selected line.	
	Move the selected trigger entry one line up in the firing sequence of triggers.  This option is not available if there are no <b>not signalled</b> triggers above the selected line.	
	Click to fire the selected trigger. You can also fire the trigger by double-clicking on it.	
	Click to toggle auto-firing on and off.  Auto-firing will fire the un-signalled triggers in your trigger set sequentially. If your set matches a valid execution path, then the simulation will run automatically. Out of sequence or unused triggers will be 'lost'.  A breakpoint pauses the auto-firing and you will need to click on the next trigger to resume auto-firing the simulation.	<a href="#">Trigger Sets and Auto-Firing</a> [2512]
	Delete the selected trigger(s) from the list.	

#### Context Menu Options

Options	Action	See also
<b>Signal Selected</b>	Signal, or fire, the selected <b>not signalled</b> trigger.	
<b>Remove Selected</b>	Remove a <b>not signalled</b> trigger from the sequence.	
<b>Re-Signal Selected</b>	Fire a used or signalled trigger again.	
<b>Set All to Unsignalled</b>	Set all used or signalled triggers to <b>not signalled</b> .	
<b>Clear Trigger List</b>	Clear all triggers from the window, regardless of their status.	

#### Learn more

- [Connector Type-Specific Options](#)  <sup>1103</sup>
- [Trigger Sets and Auto-Firing](#)  <sup>2512</sup>

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Simulation | Triggers | Using Trigger Sets 1**
- (Alt+F1) | **Enterprise Architect | Simulation | Triggers | Using Trigger Sets 2**
- (Alt+F1) | **Enterprise Architect | Simulation | Triggers | Using Trigger Sets 3**

## 15.14 Waiting Triggers

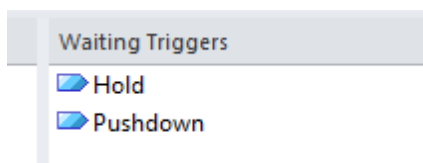
When a simulation reaches a point where any change of state (for any thread) **requires a Trigger to proceed**, the simulation is effectively **paused** and control returns to the system. The simulation is now effectively waiting for some form of event (a real world signal) to proceed. The **Waiting Triggers list** is useful in helping to determine which Trigger should be manually signalled.

Access **Analyzer | Simulation Events**

The **Waiting Triggers** list on the **Simulation Events** tab is:

- **Populated on each Simulation cycle** with any Triggers that would have an immediate effect if signalled
- Populated with a **discrete set** (any duplicates are not shown as a Trigger is effectively **broadcast** to all transitions at once)
- Activated by **double clicking** on the Trigger of interest
- Includes **all possible triggers** - including those activating transitions on **parents** of currently nested states

This example shows that the current simulation has hit a point where two possible Triggers can influence the flow of execution.



Due to the nature of Triggers and their effects, the list can refer to the following example situations equally validly:

- A single state has two outgoing transitions which are respectively waiting for Hold and Pushdown; firing one of these will activate the associated transition in the simulation
- A single state has two or more possible triggers for the same transition, such as a security camera being switched on by a motion detector, sound detector or heat detector
- Two (or more) threads (concurrent regions) each have a state waiting on either Hold or Pushdown; firing one of these triggers will result in the thread(s) waiting on that trigger to proceed while the other thread(s) will remain blocked
- A child state is waiting on one of the triggers while a parent state is waiting on the other; firing a trigger will result in the associated transition being fired and either the child or parent proceeding accordingly
- Any combination of the above

Learn more

- [Connector Type-Specific Options](#) 

## 15.15 Re-Signal Triggers

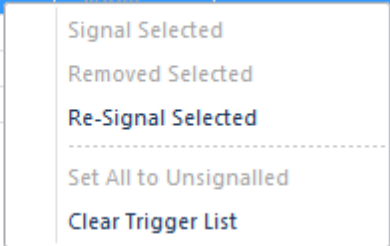
It is possible to **re-signal a Trigger** as a shortcut of dragging in additional Trigger instances for signalling.

**Access** [Analyzer](#) | [Simulation Events](#)

The Simulation Events window contains a list of Triggers which have **already fired**. By **right clicking on a Trigger** that you wish to signal again, you can use the context menu to cause the **re-signal** to happen.

The image below demonstrates this in action. When a signal is re-signalled, a new copy is made and placed at the end of the signalled triggers list, where it is automatically fired again.

Sequence	Trigger	Status	Type	Parameters
1	Pushdown	used	Signal	
2	Pushup	used		
3	Pushdown	used		
4	Pushup	used		

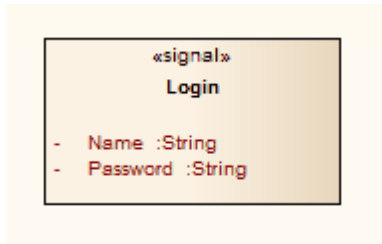
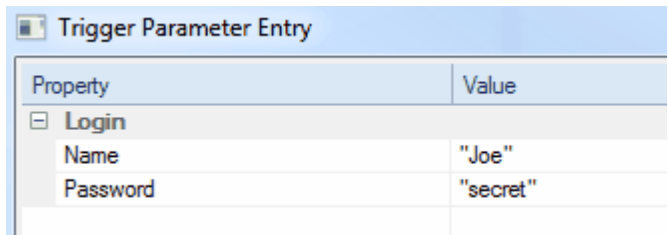


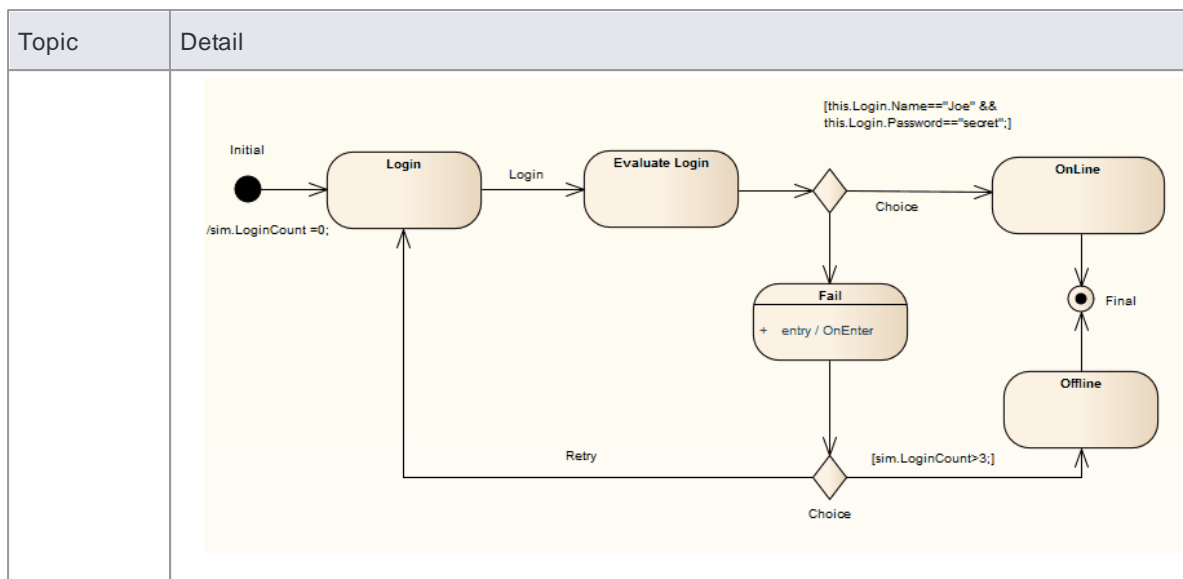
### Learning Center topics

- (Alt+F1) | [Enterprise Architect](#) | [Simulation](#) | [Triggers](#) | [Using Trigger Sets 1](#)
- (Alt+F1) | [Enterprise Architect](#) | [Simulation](#) | [Triggers](#) | [Using Trigger Sets 2](#)
- (Alt+F1) | [Enterprise Architect](#) | [Simulation](#) | [Triggers](#) | [Using Trigger Sets 3](#)

## 15.16 Trigger Parameters

Trigger parameters are **arguments passed into the simulation along with a trigger** when it is fired. They allow for **complex behavior** to be specified decision to be made based on variables and data passed into a simulation at run-time by a fired trigger (event).

Topic	Detail								
<b>Introduction</b>	<p>To use <b>trigger parameters</b> you</p> <ul style="list-style-type: none"> <li>First create a <b>Signal</b> element with the appropriate attributes that will become your parameters at run time</li> <li>On a suitable transition in your diagram, <b>create a trigger</b> that is based on the signal created earlier</li> <li>Now at run-time you will be prompted to <b>enter suitable parameters</b> - they are then passed in along with the trigger.</li> </ul>								
<b>Signals</b>	<p>A Signal element is a template or <b>specification</b> from which actual <b>triggers</b> can be built. The example below has two arguments, a Name and a Password. These will be filled in at execution time either manually or as part of a pre-defined trigger set.</p>  <pre> classDiagram     class LoginSignal["«signal» Login"] {         - Name :String         - Password :String     }   </pre>								
	<p>The Trigger parameters '<b>prompt</b>' that asks for suitable values for each parameter. <b>Note that you need to enclose strings in double quotes</b>, otherwise the interpreter will think you are referring to other <b>variables</b>.</p>  <table border="1"> <thead> <tr> <th>Property</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>[-] Login</td> <td></td> </tr> <tr> <td>    Name</td> <td>"Joe"</td> </tr> <tr> <td>    Password</td> <td>"secret"</td> </tr> </tbody> </table>	Property	Value	[-] Login		Name	"Joe"	Password	"secret"
Property	Value								
[-] Login									
Name	"Joe"								
Password	"secret"								
	<p>Below is an example diagram that makes use of trigger <b>parameters</b>. At the <b>Evaluate Login</b> state, the simulation examines the variables passed in as trigger <b>parameters</b> and makes a decision to either accept the credentials or deny them.</p>								



#### Learning Center topics

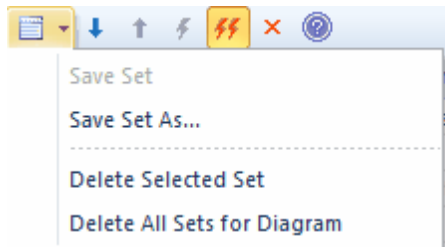
- (Alt+F1) | [Enterprise Architect](#) | [Simulation](#) | [Triggers](#) | [Triggers With Parameters](#)
- (Alt+F1) | [Enterprise Architect](#) | [Simulation](#) | [Triggers](#) | [Creating Triggers With Parameters](#)

## 15.17 Trigger Sets and Auto-Firing

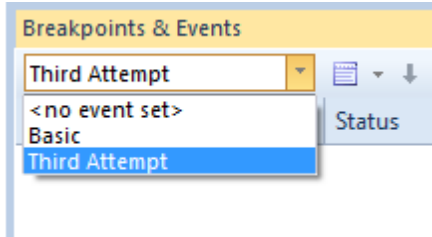
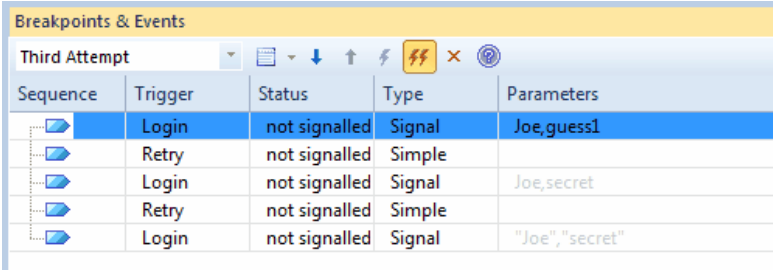

**Trigger Sets** are a powerful means of **automating and streamlining the execution**, testing and validation of simulation models. By **re-using sets of triggers** (with or without parameters) it is possible to quickly and efficiently walk through many simulation scenarios, either manually or automatically using the 'auto-firing' tool.

**Access** [Analyzer | Breakpoints & Events > Simulation Breakpoints](#)

### About Trigger Sets

Topic	Detail	See also
<b>Trigger Sets</b>	<ul style="list-style-type: none"> <li>Stored with an <b>associated diagram</b></li> <li>Made up of a <b>list of Triggers</b> in a set sequence</li> <li>May include Trigger <b>parameters</b> where necessary</li> <li>May be used <b>manually</b> by double clicking Triggers to fire as required</li> <li>May be used as part of the "<b>auto-fire</b>" behavior to automate execution</li> <li><b>Managed</b> from the Simulation Events Window</li> </ul>	
<b>Managing Sets</b>	<p>Trigger sets can be created by manually dragging triggers into the active triggers list and then using the Manage Trigger Sets drop menu to save a new set.</p> <p>It is also possible to save a set of triggers built up during a single simulation setting as a new set. This is convenient for creating multiple test paths through a simulation, based on saving the manually fired triggers for each test case.</p> <p><b><u>The Manage Triggers Menu</u></b></p>  <p>You can also delete a set and delete all sets for the current diagram.</p> <p>It is also possible to load a set, modify parameters and/or order of firing and save the set with a new name. This is a convenient method for rapidly creating a suite of simulation test scripts.</p>	
<b>Using Sets</b>	To use a trigger set you first select it by name from the trigger set drop	



Topic	Detail	See also
	<p>list as in the example image below. Once selected it loads the trigger list window with the defined trigger set.</p> <p>Note that the special item <b>&lt;no event set&gt;</b> means no set is currently selected. At the start of each simulation, if a set is selected, it will be loaded afresh for the next run. If <b>&lt;no event set&gt;</b> is selected, the trigger list will be cleared.</p>  <p>Once you have selected a trigger set and the list of triggers loaded you have two options:</p> <ul style="list-style-type: none"> <li>• Fire the triggers as required manually</li> <li>• Use the auto-fire feature to fully automate the simulation (see below)</li> </ul> 	
<b>Auto-Firing</b>	<p>Auto-firing is a convenient way of streamlining your simulations. Once you have loaded a trigger set, if you select the auto-fire button (below) then Enterprise Architect will automatically pick up waiting triggers when it reaches an impasse in the simulation. In practice, this means that trigger sets matching exactly a path through the simulation will automatically run without your intervention.</p> <p>As you can save any number of trigger sets with different pathways and trigger parameters, you can effectively and quickly test and work with many different scenarios.</p> 	
<b>Auto-Firing Rules</b>	<p>When a simulation runs with auto-firing enabled, Enterprise Architect will wait until a point is reached where the simulation is 'blocked' or stable, waiting on one or more triggers to advance the simulation. At</p>	

Topic	Detail	See also
	<p>that time, the first unfired trigger in the list will be picked up and fired into the simulation. The outcome depends on the relevance and perhaps on the parameters of the trigger.</p> <ul style="list-style-type: none"><li>• If the trigger matches a 'waiting' trigger it is immediately consumed and the simulation advances</li><li>• If the trigger matches no 'waiting' trigger or possible parent transition, then the trigger is 'lost' and the simulation remains in the current state. This corresponds to a scenario such as a user pressing an 'on' button several times in succession - there is no effect other than that occasioned by the first press.</li></ul>	

## 15.18 Using Trigger Sets to Simulate an Event Sequence

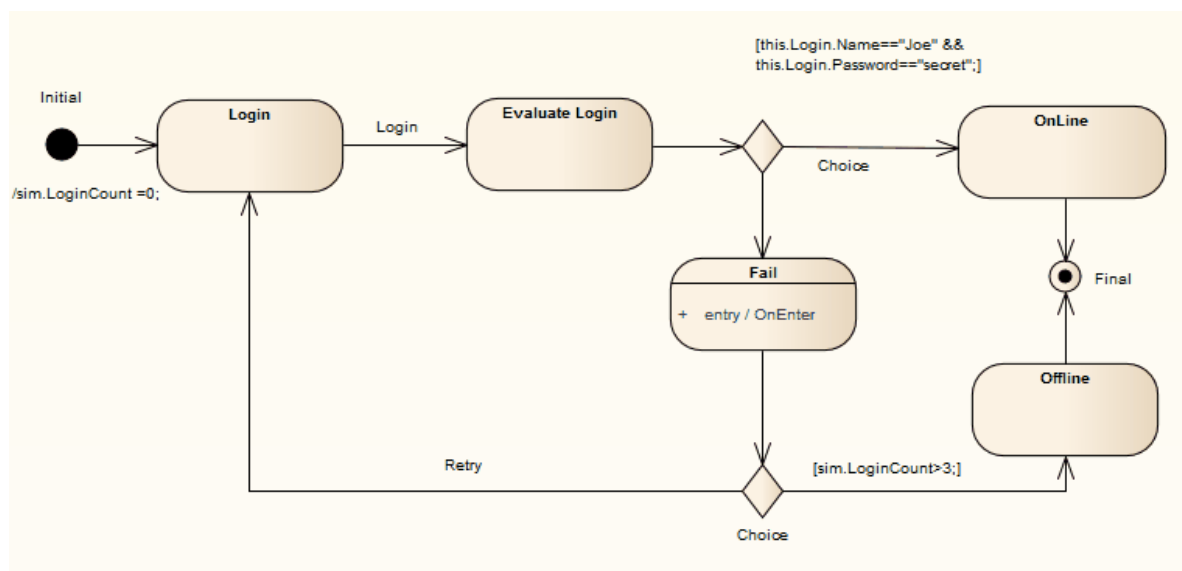
As a simple example of how useful **trigger sets** are, consider the following trigger set and associated diagram.

In this example we **simulate** a simple "three strikes and you are out" login process, taking a user name and password. The success path is waiting for the name "Joe" and the password "secret" (note - it is very important that parameters referencing strings are enclosed in quotes - otherwise the interpreter thinks the name refers to another variable within the simulation).

- Pass 1 tries Joe and guess1 - which fails
- Pass 2 tries Joe and secret, but as they are referring to variables, not strings - this fails as well
- Pass 3 shows the correct way of referencing trigger parameters and the simulation will succeed

Breakpoints & Events				
Third Attempt				
Sequence	Trigger	Status	Type	Parameters
1	Login	not signalled	Signal	Joe,guess1
2	Retry	not signalled	Simple	
3	Login	not signalled	Signal	Joe,secret
4	Retry	not signalled	Simple	
5	Login	not signalled	Signal	"Joe","secret"

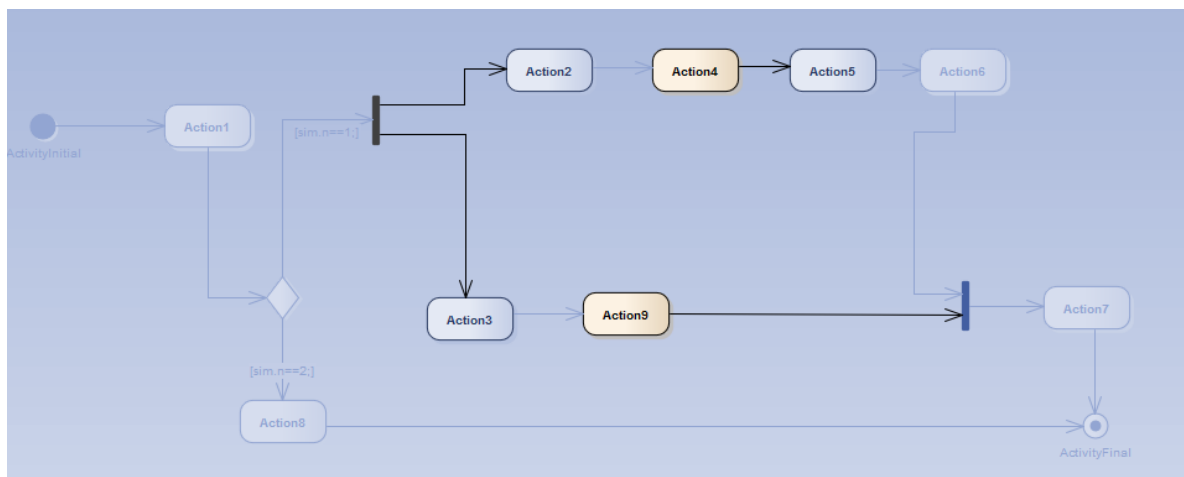
The simple diagram simulating a login process **requiring a username and password pair**.



## 15.19 Multi-threading - Forks and Joins

The Model Simulator provides the ability to handle **multi-threaded simulations** using **Fork** and **Join** nodes.

- In the example below, the current **execution point** has forked into **two threads**, each with its own active node.
- As this example progresses, the lower branch will **wait** at the **Join** node until the top branch has completed all its Actions.
- Once the two threads **merge back into one**, the Simulation will continue as a **single thread** until completion.
- When automatically stepping, each thread will be seen to execute a single step during one simulation "cycle" - although when single stepping or at a breakpoint, the behavior is to alternate step between threads as each thread receives processing time.
- Note that the **Call Stack** window will show **two active threads** and one "paused" thread in the example below. Once the threads merge there will be a return to single threaded execution.
- Also note that the **Local variables are shared (global) between all threads**. If you want to Simulate private variables on a thread you must create new Simulation variables at the start of each thread - pre-loading such variables with existing global data.



### Learn more

- [Fork/Join](#) 1307

### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Simulation | Multithreading | Multithreading Activity Diagram**

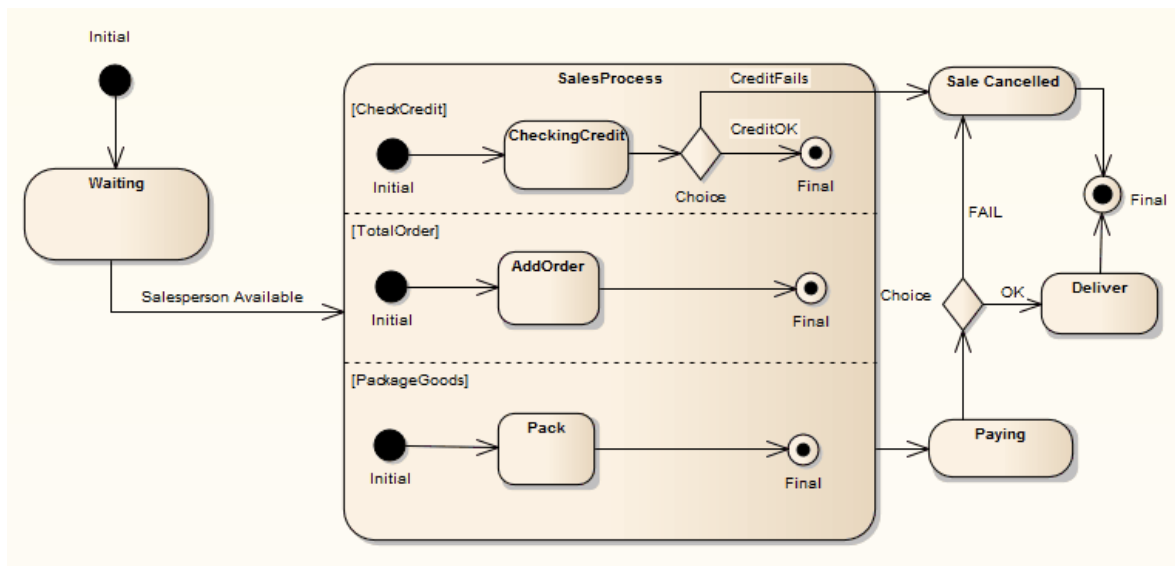
## 15.20 Multi-threading - Concurrent State Regions

**Concurrent regions** within a state represent state changes and processing that occurs in **parallel** inside one overall parent state. This is especially useful when **one region raises events** or modifies simulation variables that another region is dependent on. For example, one region could contain a simulated timer which raised events on set intervals that invoked state changes in the states within other regions.

**Concurrent regions are essentially the same as Forks and Joins** with similar logic and processing rules.

In the example below,

- When the transition to SalesProcess is taken, each region is **concurrently activated**.
- Credit is checked, the order totalled and the goods required packed up.
- However, in the event that the CreditCheck fails, this triggers the transition to the Sale Cancelled state. Note that when this occurs, the entire parent state and all owned regions are immediately exited, regardless of their processing state.
- If the Credit Check succeeds, the region moves to the final state and once the other regions have all reached their own final state, the parent state can then be exited.



### Learning Center topics

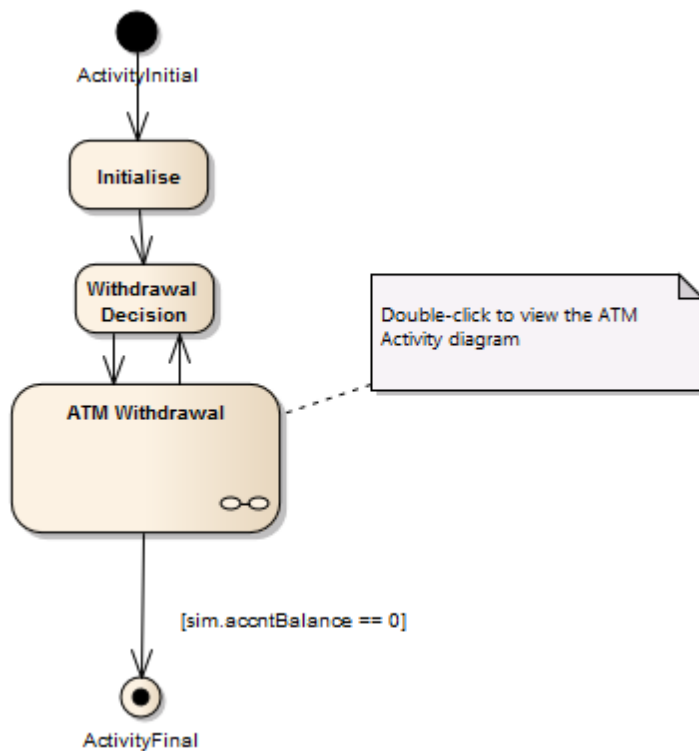
- (Alt+F1) | [Enterprise Architect | Simulation | Multithreading | Multithreading State Machines 1](#)
- (Alt+F1) | [Enterprise Architect | Simulation | Multithreading | Multithreading State Machines 2](#)

## 15.21 Using Composite Diagrams

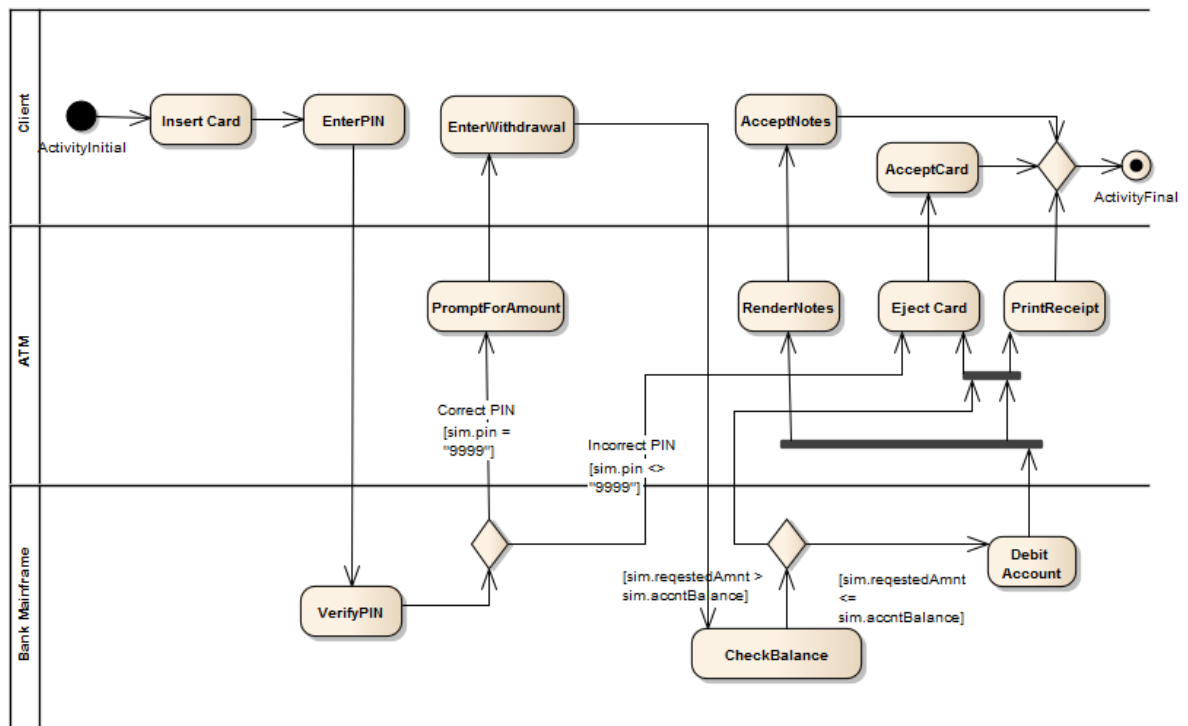
If you want to simulate processing that includes a branch represented on a different diagram (for example, to reduce complexity on the main diagram, or to hide areas of processing that are only actioned under an exception), you can use a **Composite element** to represent and access the branch on its child **Composite diagram**. When you run the simulation and it reaches the Composite element, it opens the child diagram and processes it before returning (if appropriate) to the main processing path. This is an excellent way of following the processing path in a complex process, representing sections of the process with Composite Activity elements that expand the actual processing in their respective child diagrams. You can have several Composite elements representing different stages or branches of the process.

One aspect to watch for (and that would be revealed by a failure in the simulation) is to have multiple threads that process simultaneously on separate diagrams. The simulation cannot pass to a new diagram if it is also following another thread on the current diagram.

This diagram provides an overview of an ATM cash withdrawal process:



The *ATM Withdrawal* Activity is a Composite element. If you double-click on it, you open and display the child diagram, which is a more detailed breakdown of the withdrawal process. Similarly, a simulation will open and **process** the child diagram.



### Learn more

- [Composite Elements](#) <sup>[936]</sup>
- [Multi-threading - Concurrent State Regions](#) <sup>[2517]</sup>

## 15.22 Win32 Dialog Simulation

Enterprise Architect supports the simulation of dialogs and screens created with the Win32 Dialog profile to integrate user interface design with system behavior. Dialogs can be programmatically accessed and invoked using Javascript, providing a fully customizable execution of your model.

To interact with the dialogs at runtime the 'dialog' simulation level keyword is used; this can be used for accessing or setting text variables on child controls or showing and hiding of the dialog; for example:

- dialog.Login.Show=true;
- dialog.Login.Username;
- dialog.Login.Username.Text="";

Win32 Dialog buttons can be used to trigger signals, firing off events when the button is clicked. Signal arguments can be filled from the dialog input fields, for example, to capture and send a username and password for evaluation.

### Learn more

- [Triggers](#)<sup>[2494]</sup>
- [Dynamic Simulation with Javascript](#)<sup>[2486]</sup>
- [Run Model Simulation](#)<sup>[2475]</sup>

### Learning Center topics

- **(Alt+F1) | Enterprise Architect | Simulation | User Interface |**
  - **Overview**
  - **Create WIN32 Screen**
  - **Enable a WIN32 Screen**
  - **Trigger with Buttons**
  - **Display Information to Screen**
  - **Capture UI Control Input**
  - **Run UI Simulation**





## 15.23 BPMN Simulation

**BPMN simulation** is a method for visualizing and validating the behavior of your **BPMN Business Process** diagrams. With visual indications of all **currently** executing activities **and** the possible activities that can be executed **next**, you will easily be able to identify and resolve potential issues with the process you have modeled.

Simulating BPMN models is similar to simulating standard UML Behavioral models, except that BPMN:

- Uses some different element types (such as **Gateway** instead of **Decision**) and
- Operates on scripting placed, generally, in the appropriate **Tagged Value** field associated with the connectors and elements, instead of in the **Properties** fields (and, if you prefer, rather than in the Execution Analyzer Build Scripts dialog); the scripting is written in JavaScript

### Working with BPMN Simulation

Activity	Detail	See also
<b>Create a BPMN Simulation Model</b>	When you create a BPMN model suitable for simulation, you take into consideration how you represent the start point, the flow and the conditions to be tested.	<a href="#">Create a BPMN Simulation Model</a>  <sup>[2521]</sup>
<b>Compare UML Activities to BPMN Processes</b>	The simulation of BPMN Business Process models has a number of differences to the simulation of UML Activity diagrams.	<a href="#">Compare UML Activities to BPMN Processes</a>  <sup>[2524]</sup>

### Notes

- BPMN simulation is available in the Business & Software Engineering edition and Ultimate edition of Enterprise Architect

### Learn more

- [BPMN Models](#)  <sup>[1845]</sup>
- [BPEL Models](#)  <sup>[1870]</sup>

### 15.23.1 Create a BPMN Simulation Model

As part of the process of developing a simulation model, consider which of the three options for performing the simulation you prefer to apply:

- Execute a simulation script to initialize the variables for the diagram - select **BPMN** as the **Platform**, execute the simulation as **As Script** and select the script; you would then define the conditions and decisions as JavaScript declarations within the Tagged Values of the elements and connectors on the diagram, either before you start the simulation or during the simulation
- Do not use a script, but initialize the variables within the first **Activity** and, again, modify the conditions and decisions within the Tagged Values of the elements and connectors, then execute the simulation as **Interpreted**; you can then re-initialize the variables during simulation, as well as the conditions
- Execute the simulation as **Manual** and manage the flow and conditions manually at each step

Create a BPMN diagram suitable for simulation

Step	Action	See also
1	Create a <b>Business Process</b> or <b>BPEL</b> diagram from the <b>BPMN 2.0</b> technology. If you create a BPEL diagram Enterprise Architect displays specialized dialogs to streamline the creation of compliant models.	<a href="#">Add New Diagrams</a> <sup>[822]</sup>
2	We recommend that you create a <b>Start Event</b> to clearly show where your simulation starts. You have several choices for the <b>Event Type</b> ; the choice does not influence the simulation of your model. If no Start Events are defined, the simulation will start from an <b>Activity</b> that has no incoming <b>Sequence Flows</b> .	<a href="#">Start Event</a> <sup>[1900]</sup>
3	Add all of the Activities that are involved in the Process being modeled. You have several choices for the <b>Task Type</b> ; the choice does not influence the simulation of your model. The behavior of Activities can be further decomposed by specifying an <b>Activity Type</b> of <b>Sub-Process</b> and selecting <b>Embedded</b> or <b>CallActivity</b> . Standard Loops are also supported.	<a href="#">Activity</a> <sup>[1904]</sup> <a href="#">Initialize Variables and Conditions</a> <sup>[2523]</sup>
4	Add <b>Sequence Flows</b> between your activities. In the BPEL properties dialog you can enter the condition that must be satisfied (true) before the Sequence Flow will be followed. You can also set the <b>conditionType</b> to <b>Default</b> to ensure that this flow will be taken if all other branches fail the condition specified.  If you are not working with a BPEL diagram, you use the <b>conditionExpression</b> and <b>conditionType</b> Tagged Values.	<a href="#">Sequence Flow</a> <sup>[1910]</sup> <a href="#">Initialize Variables and Conditions</a> <sup>[2523]</sup>
5	Add <b>End Events</b> for any conditions that will cause the process or active execution path to end. You have several choices for the <b>Event Type</b> ; of these only the <b>Terminate</b> type will influence the execution. In simulations with multiple active nodes, it causes the entire process to terminate instead of just the thread that reaches that node.	<a href="#">End Event</a> <sup>[1907]</sup>

Learn more

- [BPMN 2.0 Toolbox Pages](#) <sup>[1851]</sup>
- [Create BPEL 2.0 Model Structure](#) <sup>[1897]</sup>
- [Set Up Simulation Script](#) <sup>[2472]</sup>
- [How It Works](#) <sup>[2466]</sup>

Learning Center topics

- (Alt+F1) | **Enterprise Architect | Simulation | Simulation | BPMN Manual Simulation**
- (Alt+F1) | **Enterprise Architect | Simulation | Simulation | BPMN Interpreted Simulation**

### 15.23.1.1 Initialize Variables and Conditions

For a BPMN simulation model, you can **initialize your variables** in an Execution Analyzer script. You can **also** initialize these variables in the **Tagged Values** of the first **Activity** element of the process, which gives you greater flexibility in adding and changing variables as the simulation proceeds. Similarly, you can define the **conditions** and values to apply at the various decision points (Gateways) in the process, in the Tagged Values of the **Sequence Flow** connectors.

If you want to incorporate a **user-interface** into your simulation process, using **Win32**, you again use Tagged Values to identify the dialog or prompt to display, in the Activity element just prior to the point at which the value or decision is processed.

As for the simulation of UML diagrams, variables inside the **sim** object and **this** object are displayed in the **Local Variables** window.

**Access** **View | Tagged Values** (Ctrl+Shift+6)

#### Initialize Variables

1. On the diagram, click on the first Activity element in the process.
2. In the Tagged Values window, click on the drop-down arrow of the **taskType** value field, and select **Script**.
3. In the **script** value field, type in the appropriate JavaScript code, such as:

```
si m.loan=true; si m.status="undefined";
```

#### Define Conditions

1. On the diagram, click on a Sequence Flow connector that issues from a Gateway element.
2. In the Tagged Values window, click on the drop-down arrow of the **conditionType** value field, and select **Expression**.
3. In the **conditionExpression** value field (<memo>\*) click on the **Select** button ( ...) to display the Tagged Value Note window. Type in the appropriate JavaScript code, such as:

```
si m.status=="Hold"
```

4. Click on the **OK** button. The statement text displays as a label of the connector.

#### Incorporate Win32 User Interface

1. On the diagram, click on the Activity element that represents where the decision is made
2. In the Tagged Values window, click on the drop-down arrow of the **taskType** value field, and select **Script**.
3. In the **script** value field, type in the appropriate JavaScript code, such as:

```
di al og. Screen1. Show=true;
```

(This statement displays the dialog **Screen1**. You can temporarily hide the dialog by changing **Show** to **false**.)

#### Learn more

- [Create a BPMN Simulation Model](#) <sup>[2527]</sup>
- [Tagged Values](#) <sup>[1134]</sup>

- [Win32 Dialog Simulation](#)<sup>[2520]</sup>

### 15.23.2 Compare UML Activities to BPMN Processes

The execution and simulation of BPMN models have a number of differences from the execution and simulation of UML Activity diagrams. The mapping of similar concepts, and the differences between the two methods of expressing the behavior of a system, are presented here.

#### Comparison of UML Activities and BPMN Processes

UML Activity	BPMN Business Process	See also
The starting point is defined by an <b>Initial Node</b> . No method of specifying <b>why</b> the Activity was started is available.	The starting point is defined by a <b>Start Event</b> . This implies a specific cause for the Activity to start, although it may be unspecified.	<a href="#">Initial Node</a> <sup>[1313]</sup> <a href="#">Start Event</a> <sup>[1900]</sup>
The basic behavior unit in an Activity is the <b>Action</b> element. UML provides many different forms of Actions, although the simulation makes use of a small subset of these.	The basic behavior unit in an Activity is the <b>Activity</b> element. A number of different <b>Task Types</b> are available. These typically describe different methods of execution (for example <b>Manual</b> ) as opposed to what happens.	<a href="#">UML Action</a> <sup>[1266]</sup> <a href="#">BPMN Activity</a> <sup>[1904]</sup>
A <b>Control Flow</b> is used to connect the elements on an Activity diagram. A distinguishing feature is that only a single Control Flow can be followed from any node, except for an explicit Fork Node. To restrict flow on a Control Flow, add a <b>Guard</b> .	A <b>Sequence Flow</b> is used to connect the elements on a Business Process diagram. These differ from UML Activity diagrams in that all valid sequence flows are taken by default. To restrict flow on a Sequence Flow set the <b>conditionType</b> Tagged Value to <b>Expression</b> and create the script in the <b>conditionExpression</b> Tagged Value.	<a href="#">Control Flow</a> <sup>[1403]</sup> <a href="#">Sequence Flow</a> <sup>[1910]</sup>
A <b>Decision</b> node is used to explicitly model a decision being made. A <b>Merge</b> node, which uses the same syntax is used when the potential flows are combined back into one.	A <b>Gateway</b> node set to <b>Exclusive</b> is used when a single path must be selected. It is also used to combine the potential flows again. A direction may be specified as <b>Converging</b> or <b>Diverging</b> to explicitly select between the two modes.	<a href="#">Decision</a> <sup>[1294]</sup> <a href="#">Gateway</a> <sup>[1906]</sup>
A <b>Fork</b> node is used to concurrently execute multiple nodes, while a <b>Join</b> node, using the same syntax is used to wait for all incoming flows to become available and leave with a single flow.	A <b>Gateway</b> node set to <b>Parallel</b> is used to explicitly model concurrent execution of multiple nodes. It is also used to wait for all incoming flows to become available and leave with a single flow. A direction may be specified as <b>Converging</b> or <b>Diverging</b> to explicitly select between the two modes.	<a href="#">Fork/Join</a> <sup>[1307]</sup> <a href="#">Gateway</a> <sup>[1906]</sup>
There is no allowance for concurrently executing only some outputs from a node for UML Activities. If you needed this you add later Control Flows with the appropriate Guards.	A <b>Gateway</b> node set to <b>Inclusive</b> is used to explicitly model the situation where all outgoing flows with a true condition are executed concurrently.	<a href="#">Gateway</a> <sup>[1906]</sup>

UML Activity	BPMN Business Process	See also
A <b>Call Behavior Action</b> is used when behavior needs to be further decomposed by referring to an external activity.	<b>Activity</b> elements are set as an <b>CallActivity Sub-Process</b> when behavior needs to be further decomposed by referring to an external activity.	<a href="#">UML Action</a> [1266] <a href="#">BPMN Activity</a> [1904]
<b>Activity elements are sometimes directly on a diagram in place of an Action when behavior needs to be further decomposed and an external external activity is not desired. However, a strict UML model requires the use of a Call Behavior Action referencing an activity.</b>	<b>Activity</b> elements are set as an <b>Embedded Sub-Process</b> when behavior needs to be further decomposed without referring to an external activity.	<a href="#">UML Activity</a> [1279] <a href="#">BPMN Activity</a> [1904]

**Part**

---



## 16 Execution Analyzer



Whilst you are modeling the code of an application, you can perform test executions of the code and record and generate visual displays of the progress of execution for analysis, using the Visual Execution Analyzer. This builds on and processes the structures and operations of the Model Driven Development Environment (MDDE), which provides tools to design, build and debug an application, as a part of the Debugger facilities of Enterprise Architect.






One of the primary objectives of the Visual Execution Analysis feature is to provide the facility to generate Sequence, Test Domain Class and Collaborative Class diagrams from the stack traces you captured in debugging your application. This is a great way to document and understand what your program is doing during its execution phase, and to:

- Optimize existing system resources and understand resource allocation
- Verify that the system is following the rules as designed
- Produce high quality documentation that more accurately reflects system behavior
- Understand how and why systems work
- Train new employees in the structure and function of a system
- Provide a comprehensive understanding of how existing code works
- Identify costly or unnecessary function calls
- Illustrate interactions, data structures and important relationships within a system
- Trace problems to a specific line of code, system interaction or event
- Visualize why a sequence of events is important
- Establish the sequence of events that occur immediately prior to system failure
- Simulate the execution of behavior models including State Machines, Activities and Interactions

**Access** **Analyzer**  
**Project Browser | Right-click on package | Execution Analyzer**

### Operations

Operation	See
Create a sample model for your language compiler, on which to examine the facilities of the Visual Execution Analyzer.	<a href="#">Visual Execution Analyzer Samples</a> <sup>[2529]</sup>
Record executing programs and represent the behavior as a UML Sequence diagram; recording is supported for: <ul style="list-style-type: none"> <li>• Microsoft Windows Native C</li> <li>• Microsoft Windows Native C++</li> <li>• Microsoft Windows Visual Basic</li> </ul>	<a href="#">Recording Sequence Diagrams</a> <sup>[2531]</sup>

Operation	See
<ul style="list-style-type: none"> <li>• Microsoft .NET Family (C#, J#, VB)</li> <li>• Sun Microsystems Java</li> <li>• PHP</li> </ul>	
Record native Windows applications and profile their behavior.	<a href="#">Profiling</a>  <sup>[2555]</sup>
Create and work with objects created within the Enterprise Architect modeling environment using a dynamic Object Workbench.	<a href="#">Object Workbench</a>  <sup>[2567]</sup>
Run xUnit tests.	<a href="#">Unit Testing</a>  <sup>[2573]</sup>
Testpoint Management - provides the facility to pass or fail application tasks, viewing test results in real time as the program executes and results are saved.	<a href="#">Testpoint Management</a>  <sup>[2577]</sup>
Simulate UML behavior models to verify their logical and design correctness, for: <ul style="list-style-type: none"> <li>• Activities</li> <li>• Interactions and Sequences</li> <li>• State Machines</li> </ul>	<a href="#">Model Simulation</a>  <sup>[2463]</sup>

#### Learn more

- [Code, Build & Debug](#)  <sup>[2172]</sup> (for the Enterprise Architect MDDE and Debugger)

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Execution Analysis**





## 16.1 Visual Execution Analyzer Samples

Enterprise Architect enables you to easily import complete sample models (packages), including all necessary model information, code and build scripts. These sample patterns make it simple to explore and try out the Visual Execution Analyzer. You can generate an example model for:

- Java
- Microsoft.NET
- Microsoft C++
- PHP Apache

**Access** **Project | New Model (Ctrl+Shift+M) > VEA Examples**  
**Project Browser | Package context menu | Add a Model using Wizard > VEA Examples**

### Reference

Field	Action	See also
<b>Technology</b>	Select the appropriate technology.	
<b>Name</b>	Displays the samples available for the selected technology; select the required sample to import.	
description field	Displays a description of the selected sample.	
<b>Destination folder</b>	Browse for and select the directory in which to load the source code for the sample.	
<b>Use Local Path</b>	Enable the selection of an existing local path to place the source code under; changes the <b>Destination folder</b> field to a drop-down selection.	
<b>Compiler command</b>	Displays the default compiler command path for the selected technology; you must either: <ul style="list-style-type: none"> <li>• Confirm that the compiler can be found at this path, or</li> <li>• Edit the path to the compiler location</li> </ul>	<a href="#">Local Paths</a> 
<b>Edit Local Paths</b>	Many VEA examples specify their compiler using a local path.  The first time you use any sample you must click on this button to ensure the local path points to the correct location.  The Local Paths dialog displays.	<a href="#">Local Paths Dialog</a> 

### Notes

- If required, you can define custom samples by adding files to the *AppSamples* directory where Enterprise Architect is installed; top-level directories are listed as Technologies and can contain an icon file to customize the icon displayed for the technology
  - Directories below this are defined as groups in the patterns list; the patterns are defined by the presence of four files with a matching name: a zip file (.zip), XMI file (.xmi), config file (.cfg) and optional icon (.ico)
- The *config* file supports the following fields:
  - [provider], [language], [platform], [url], [description], [version] - all displayed in the description field
  - [xmrootpaths] - the root path of the source code in the exported xmi; this is replaced with the selected destination folder when the user applies the application pattern

#### Learn more

- [Model Wizard](#)<sup>[753]</sup>
- [Recording Sequence Diagrams](#)<sup>[2531]</sup>
- [Profiling](#)<sup>[2555]</sup> }
- [Object Workbench](#)<sup>[2567]</sup> } Use the Visual Execution Analyzer samples to explore these
- [Unit Testing](#)<sup>[2573]</sup> } areas of Visual Execution Analysis
- [Testpoint Management](#)<sup>[2577]</sup> }
- [Model Simulation](#)<sup>[2463]</sup> }

## 16.2 Recording Sequence Diagrams

This section explains how to use the Visual Execution Analyzer to record execution flow in the form of a Sequence diagram.

A Sequence diagram provides easy to understand visual information including:

- A representation of how information is passed through a system
- The sequence of various functions and their corresponding parameters
- A clear view of how different Classes interact to create behavior
- A visual overview of how data structures are used to produce results

A Sequence diagram extends traditional analysis to help identify errors in logic, explain unexpected system behavior and identify data flow inconsistencies; the Visual Execution Analyzer extends analysis through the use of a comprehensive array of reports that detail everything from state transitions through to the contents of the stack at a given time.

**Access**    **Analyzer | Recorder**



### Use to

- Set up and control the recording of code execution
- Interpret the results of recording
- Generate Sequence diagrams from the recording, and moderate the contents of the diagram

### Topics

Review the following topics in sequence:

Topic	Link
An overview of how the Visual Execution Analyzer creates a visual representation of the execution of an application, outlining the functions that are being called, the types of messages being sent, the key data structures used and the relationships between different Classes.  The diagram makes it much simpler to understand how information is moved throughout the system and what values are being passed by various functions.	<a href="#">How It works</a> <sup>[2532]</sup> <a href="#">The Recording History</a> <sup>[2533]</sup> <a href="#">Diagram Features</a> <sup>[2535]</sup>
Preparing to record execution of the application.	<a href="#">Setup for Recording</a> <sup>[2535]</sup>
Deploying recording markers, breakpoints and marker sets.	<a href="#">Place Recording Markers</a> <sup>[2536]</sup>
Controlling the recording session, using the Record & Analyze window.	<a href="#">Control the Recording Session</a> <sup>[2544]</sup>
Interpreting the results of a recording, saving and retrieving these results and generating a Sequence diagram from the recording.	<a href="#">Generating Sequence</a>

Topic	Link
	<a href="#">Diagrams</a>  <small>[2549]</small>
A description of how you can generate Sequence diagrams that show transitions in state as a program executes.	<a href="#">Reporting State Transitions</a>  <small>[2550]</small>

### Notes

- Recording is available to users of all editions of Enterprise Architect except the Desktop edition



### Learning Center topics

- (Alt+F1) | [Enterprise Architect | Execution Analysis | C++/Native Samples | C++ Recording 1](#)
- (Alt+F1) | [Enterprise Architect | Execution Analysis | C++/Native Samples | C++ Recording 2](#)
- (Alt+F1) | [Enterprise Architect | Execution Analysis | C++/Native Samples | C++ Recording 3](#)

## 16.2.1 How it Works

This topic explains how the Visual Execution Analyzer generates Sequence diagrams.

### Topics

Topic	Detail	See also
<b>Usage</b>	<p>The Visual Execution Analyzer enables you to generate a Sequence Diagram by recording the live execution of an application. As the application runs, a history of function calls is recorded. This history can be used to generate Sequence Diagrams.</p> <p>The diagram creates a visual representation of the execution of an application, outlining what functions are being called, the types of messages being sent, the key data structures used and the relationships between different Classes; that is, how information is moved throughout the system and what values are being passed by various functions. .</p> <p>The diagram below illustrates the Sequence Diagram output for a program that calculates the price of books.</p>	<p><a href="#">The Recording History</a>  <small>[2533]</small></p> <p><a href="#">Diagram Features</a>  <small>[2535]</small></p>

Topic	Detail	See also
	<div><pre>sequenceDiagram     participant Test     participant BookDB     participant vltitude Book     participant PriceTotaler      Test-&gt;&gt;Main()     activate Test     Test-&gt;&gt;BookDB: BookDB()     activate BookDB     BookDB-&gt;&gt;Test: AddBook(BookDB)     deactivate BookDB     loop 1 (4)         Test-&gt;&gt;BookDB: AddBook(string, decimal, bool)         activate BookDB         BookDB-&gt;&gt;vltitude Book: Book(string, decimal, bool)         activate vltitude Book         vltitude Book-&gt;&gt;Test:          deactivate vltitude Book     end     Test-&gt;&gt;PriceTotaler: ProcessPaperbackBooks(ProcessBookDelegate)     activate PriceTotaler     loop 2 (3)         Test-&gt;&gt;BookDB: PrintTitle(Book)         activate BookDB         BookDB-&gt;&gt;Test: Print(string)         deactivate BookDB     end     Test-&gt;&gt;PriceTotaler: PriceTotaler()     activate PriceTotaler     loop 3 (3)         Test-&gt;&gt;PriceTotaler: AddBookToTotal(Book)         activate PriceTotaler         PriceTotaler-&gt;&gt;Test: AveragePrice() decimal         deactivate PriceTotaler     end     deactivate PriceTotaler     deactivate Test</pre></div> <p>Enterprise Architect will record arguments to functions, and can optionally capture state transitions for a given State Machine. Calls to functions are aggregated on the class by default, but lifelines can also be created for each instance of an object.</p> <p>This information can be integrated with existing system knowledge and test data to optimize code execution, reduce errors and determine why application failure and system crashes occur.</p> <p>If an application crashes, data corruption such as a stack overflow can prevent you from diagnosing and rectifying the problem; however, the Visual Execution Analyzer enables you to record a given execution sequence to provide a reliable source of information that might further explain why a crash occurred.</p> <p>A Sequence diagram can convey more detail and provide greater understanding than reading unfamiliar code that might have been written by someone else; it also makes it easier to document existing code when the Sequence diagram illustrates functions that are being called and the specific sequence of events that occur to produce a particular type of system behavior.</p>	

**Learn more**

- [Setup for Recording](#)<sup>[2535]</sup>

**16.2.1.1 The Recording History**

All information recorded when the execution analysis of an application encounters user-defined recording markers is held in the Record & Analyze window.

**Access**    **Analyzer | Recorder**

Facilities

Facility	Information/Options	See also
<b>Information Display</b>	<p>The columns in the Record &amp; Analyze window are as follows:</p> <ul style="list-style-type: none"> <li>• <b>Sequence</b> - The unique sequence number</li> <li>• <b>Threads</b> - The operating system thread ID</li> <li>• <b>Delta</b> - The elapsed thread CPU time since the start of the sequence</li> <li>• <b>Method</b> - There are two <b>Method</b> columns: the first shows the caller for a call or for a current frame if a return; the second shows the function called or the function it is returning to</li> <li>• <b>Direction</b> - <i>Stack Frame Movement</i>, can be <i>Call</i>, <i>Return</i>, <i>State</i>, <i>Breakpoint</i> or <i>Escape</i> (<i>Escape</i> is used internally when producing a Sequence diagram, to mark the end of an iteration)</li> <li>• <b>Depth</b> - The stack depth at the time of a call; used in the generation of Sequence diagrams</li> <li>• <b>State</b> - The state between sequences</li> <li>• <b>Source</b> - There are two <b>Source</b> columns: the first shows the source filename and line number of the caller for a call or, if a return, for a current frame; the second shows the source filename and line number of the function called or function returning</li> <li>• <b>Instance</b> - There are two <b>Instance</b> columns, which only have values when the Sequence diagram produced contains State Transitions; the values consist of two items separated by a comma - the first item is a unique number for the instance of the Class that was captured, and the second is the actual instance of the Class</li> </ul> <p>For example: supposing a Class <i>CName</i> has an internal value of 4567 and the program created two instances of that Class; the values might be:</p> <ul style="list-style-type: none"> <li>• 4567,1</li> <li>• 4567,2</li> </ul> <p>The first entry shows the first instance of the Class and the second entry shows the second instance.</p>	
<b>Operations on Information</b>	<p>The Record &amp; Analyze window toolbar provides a range of facilities for controlling the recording of the execution of an Analyzer script.</p> <p>You can perform a number of operations on the results of a recording, using the Record &amp; Analyze window context menu, once the recording is complete.</p>	<p><a href="#">Recorder Toolbar</a> [2544]</p> <p><a href="#">Working With Recording History</a> [2546]</p>

Notes

- The checkbox against each operation is used to control whether or not this call can be used to create

a Sequence, Test Domain Class or Collaborative Class diagram from this history

- In addition to enabling or disabling the call using the checkbox, you can use context menu options to enable or disable an entire call, all calls to a given method, or all calls to a given Class

### 16.2.1.2 Diagram Features

When you generate a Sequence diagram, it includes the following features:

Feature	Detail	See also
<b>References</b>	When the Visual Execution Analyzer cannot match a function call to an operation within the model, it still creates the sequence but also creates a reference for any Class that it cannot locate.  It does this for all languages.	<a href="#">Generating Sequence Diagrams</a> <sup>[2549]</sup>
<b>Fragments</b>	Fragments displayed in the Sequence diagram represent loops or iterations of a section(s) of code.  The Visual Execution Analyzer attempts to match function scope with method calls to as accurately as possible represent the execution visually.	
<b>States</b>	If a State Machine has been used during the recording process, any transitions in State are presented after the method call that caused the transition to occur.  States are calculated on the return of every method to its caller.	<a href="#">Reporting State Transitions</a> <sup>[2550]</sup>

### 16.2.2 Setup for Recording

This section explains how you prepare to record execution of the application.

Topic	Link
Prerequisites - To set up the environment for recording Sequence diagrams you must: <ul style="list-style-type: none"> <li>• Have completed the basic setup for Build &amp; Debug and created Execution Analysis scripts for the package</li> <li>• Be able to successfully debug the application</li> </ul>	<a href="#">Prerequisites</a> <sup>[2173]</sup> <a href="#">Analyzer Scripts</a> <sup>[2175]</sup> <a href="#">Debugging</a> <sup>[2222]</sup>
Narrow the focus of a recording by applying filters.	<a href="#">Configuring Recording Detail</a> <sup>[2219]</sup>
Control the detail of a recording by adjusting the stack depth.	<a href="#">Control Stack Depth</a> <sup>[2536]</sup>

### 16.2.2.1 Control Stack Depth

When recording particularly high-level points in an application, the stack frame count can result in a lot of information being collected; to achieve a quicker and clearer picture, it is better to limit the stack depth on the toolbar of either:

- The Breakpoint and Markers window or
- The Record & Analyze window

Access **Analyzer | Recorder**

#### Set the recording stack depth

You set the recording stack depth in the numerical field on the toolbar of the Breakpoints & Markers window or the Record & Analyze window:



By default, the stack depth is set to three frames. The maximum depth that can be entered is **30** frames.

The depth is relative to the stack frame where a recording marker is encountered; so, when recording begins, if the stack frame is **6** and the stack depth is set to **3**, the Debugger records the frames **6** through **8**.

For situations where the stack is very large, it is recommended that you first use a low stack depth of 2 or 3. From there you can gradually increase the stack recording depth and insert additional recording markers to expand the picture so that all the necessary information is displayed.

#### Learn more

- [Recorder Toolbar](#)<sup>[2544]</sup>
- [Breakpoint & Markers](#)<sup>[2540]</sup>
- [Control the Recording Session](#)<sup>[2543]</sup>

### 16.2.3 Place Recording Markers

This section explains how to place record markers, which enable you to silently record code execution between two points. The recording can be used to generate a Sequence diagram.

As this process records the execution of multiple threads, it can be particularly useful in capturing event driven sequences (such as mouse and timer events).

Access **Analyzer | Breakpoints & Events**

#### Use to

- Set recording markers, to define what is to be recorded during code execution in order to generate a Sequence diagram

#### Topics



Topic	Link
Different recording markers can be used for recording the execution flow; see the related links for information on the properties and usage of these markers.	<a href="#">Marker Types</a> <sup>[2538]</sup> <a href="#">Set Record Markers</a> <sup>[2537]</sup>
How to manage breakpoints in the Breakpoint & Markers window.	<a href="#">The Breakpoint &amp; Events Window</a> <sup>[2540]</sup>
How to activate and deactivate markers.	<a href="#">Breakpoint and Marker Management</a> <sup>[2224]</sup>
Working with Marker Sets - when you create a breakpoint or marker, it is automatically added to a marker set, either the Default set or a set that you create for a specific purpose.	<a href="#">Working with Marker Sets</a> <sup>[2541]</sup> <a href="#">Recording Activity for a Class</a> <sup>[2542]</sup>

**Notes**

- The *Breakpoint and Marker Management* topic (Software Engineering) covers a different perspective

**Learning Center topics**

- (Alt+F1) | **Enterprise Architect | Execution Analysis | Recording Execution**

**16.2.3.1 Set Record Markers**

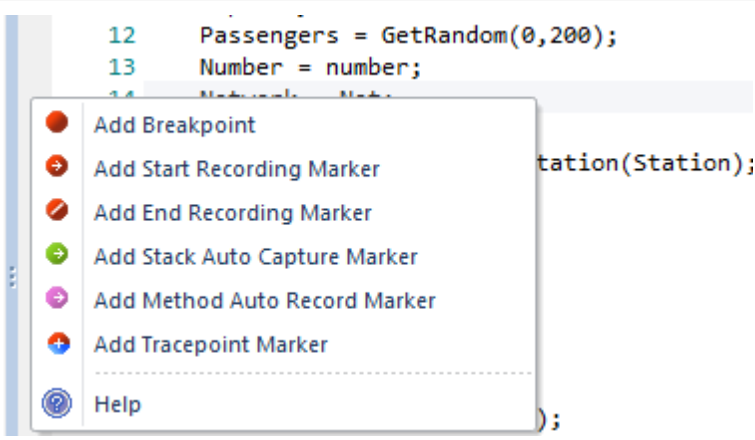
Markers are set in the source code editor. Like breakpoints, they are placed on a line of code; when that line of code executes, the Execution Analyzer performs the code execution recording action appropriate to the marker.

**Access** Click on the Class containing code: **F12**

**How to**

To set a recording marker

Step	Action	See also
1	Press <b>F12</b> to open the source code to debug, in the integrated source code editor.	<a href="#">Editing Source Code</a> <sup>[2146]</sup>
2	Find the appropriate code line and right click in the left (Breakpoint) margin to bring up the breakpoint/marker context menu; select the required marker type:	<a href="#">Marker Types</a> <sup>[2538]</sup>

		
3	If a <b>Start Recording Marker</b> has been set, you must also set an <b>End Recording Marker</b> .	

### 16.2.3.2 Marker Types

Recording markers are similar to breakpoints; however, instead of breaking execution as it does at a breakpoint, the debugger takes the action defined by the type of marker. If the marker is denoted as a recording *start point*, the debugger immediately begins to trace all executed calls from that point for the breaking thread. Recording is stopped again when either the thread that is being captured terminates or the thread encounters a recording *end point*.

Recording markers are set in the source code editor. If you right-click on the breakpoint margin at the point to begin recording, a context menu displays that enables you to select the type of a marker.

#### Use to

- Set Start and End points for recording
- Set Markers for a Single Method
- Set Stack Trace Markers

#### Reference

Marker	Detail	See also
<b>Start Recording marker</b>	Select the <b>Add Start Recording Marker</b> option, then right-click on the breakpoint margin at the point to stop recording and select the <b>Add End Recording Marker</b> context menu option; the markers are shown below:	<a href="#">The Breakpoints &amp; Events Window</a> <sup>[2540]</sup>
<b>End Recording marker</b>		<a href="#">Analyzer</a>

© 1998-2014 Sparx Systems Pty Ltd

Marker	Detail	See also
	<div><div><div>185 //</div><div>186 // CRecurrenceDlg message handlers</div><div>187</div><div>188 BOOL CRecurrenceDlg::OnInitDialog()</div><div>189 {</div><div>190 CBCGPDialg::OnInitDialog();</div><div>191</div><div>192 UINT nMask =</div><div>193 CBCGPDateTimeCtrl::DTM_SPIN  </div><div>194 CBCGPDateTimeCtrl::DTM_DATE  </div><div>195 CBCGPDateTimeCtrl::DTM_TIME  </div><div>196 CBCGPDateTimeCtrl::DTM_CHECKBOX  </div><div>197 CBCGPDateTimeCtrl::DTM_DROPCALENDAR  </div><div>198 CBCGPDateTimeCtrl::DTM_CHECKED;</div><div>199</div><div>200 UINT nFlags = CBCGPDateTimeCtrl::DTM_CHECKED   CBCGPDateTimeCtrl::DT</div><div>201 //-----</div><div>202 // Setup date fields:</div></div></div> <p>Recording markers can be nested. When a new Method Auto Record marker is hit while recording the stack depth to record to will be extended to cover the current method and the required depth from that function.</p>	

Learn more

- [Set Record Markers](#)<sup>[2537]</sup>
- [Recording Activity for a Class](#)<sup>[2542]</sup>

Learning Center topics

- (Alt+F1) | Enterprise Architect | Execution Analysis | Recording Execution | Method Record Marker

16.2.3.3 The Breakpoints & Markers Window

Using the Breakpoints & Markers window, you can apply control to Visual Execution Analysis when recording execution to generate Sequence diagrams; for example, you can:

- Enable, disable and delete markers
- Manage markers as sets
- Organize how markers are displayed, either in list view or grouped by file or Class

Access Analyzer | Breakpoints & Events

Learn more

- [Breakpoint and Marker Management](#)<sup>[2224]</sup>
- [Working with Marker Sets](#)<sup>[2541]</sup>

Learning Center topics

- (Alt+F1) | Enterprise Architect | Execution Analysis | Recording Execution


### 16.2.3.4 Working with Marker Sets

Marker sets enable you to create markers as a named group, which you can reapply to a code file for specific purposes.

You can perform certain operations from the Breakpoints & Events window alone, but to understand and use markers and marker sets you should also display the appropriate code file in the Source Code Viewer (click on the Class element and press **(F12)**).

**Access** **Analyzer | Breakpoints & Events: Set Options toolbar icon**

#### Topics

Topic	Detail	See also
<b>Example of Use</b>	<p>You might create a set of Method Auto Record markers to record the action of various functions in the code, and a set of Stack Capture markers to record the sequence of calls that cause those functions to be called.</p> <p>You could then create Sequence diagrams from the recordings under each set.</p>	<a href="#">Marker Types</a> <sup>[2538]</sup>
<b>Create a Marker Set</b>	<p>To create a marker set from the Breakpoints &amp; Events window, click on the drop-down arrow on the <b>Set Options</b> icon (  ) and select the <b>New Set</b> option.</p> <p>The New Breakpoint Marker Set dialog displays; in the <b>Enter New Set Name</b> field, type a name for the set, and click on the <b>Save</b> button.</p> <p>The set name displays in the text field to the left of the <i>Set Options</i> icon.</p> <p><i>Alternatively</i>, you can either:</p> <ul style="list-style-type: none"> <li>• Create a Class Activity marker set or</li> <li>• Select the <b>Save as Set</b> option from the <b>Set Options</b> drop-down, to make an exact copy of the currently-selected set, which you can then edit</li> </ul>	<a href="#">Recording Activity for a Class</a> <sup>[2542]</sup>
<b>Accessing Sets</b>	<p>To access a marker set, click on the drop-down arrow on the text field to the left of the <b>Set Options</b> icon, and select the required set from the list.</p> <p>The markers in the set are listed in the Breakpoints &amp; Events window.</p> <p>You would normally load a marker set prior to the point at which an action is to be captured.</p> <p>For example, to record a sequence involving a particular dialog, when you begin debugging you would load the set prior to invoking the dialog; once you bring up the dialog in the application, the operations you have marked are recorded.</p>	
<b>Add Markers to Set</b>	<p>To add markers to a marker set, add each required marker to the appropriate line of code in the Source Code Viewer.</p> <p>The marker is immediately added to whichever set is currently listed in</p>	<a href="#">Set Record Markers</a> <sup>[2537]</sup>

Topic	Detail	See also
	<p>the Breakpoints &amp; Events window.</p> <p>Each marker listed on the dialog has a checkbox in the <b>Enabled</b> column; newly-added markers are automatically enabled, but you can disable and re-enable the markers quickly as you check the code.</p>	
<b>Storage of Sets</b>	<p>When you create a marker set it is immediately saved within the model; any user using the model has access to that set.</p> <p>However, the <b>Default</b> set, which always exists for a model, is a personal workspace, is not shared and is stored external to the model.</p>	
<b>Delete a Marker from a Set</b>	Right-click on the marker and select the <b>Delete Breakpoint</b> context menu option.	
<b>Delete a Set</b>	<p>If you no longer require a marker set, access it on the Breakpoints &amp; Events window and select the <b>Delete Selected Set</b> option from the <b>Set Options</b> drop-down list.</p> <p>You can also clear <i>all</i> user-defined marker sets by selecting the <b>Delete all sets</b> option; a prompt displays to confirm the deletion.</p>	

### Notes

- Marker Sets are very simple and flexible but, as they are available for use by any user of the model, they can be easily corrupted; consider the following guidelines:
  - When naming a set, use your initials in the name and try to indicate its use, so that other model users can recognize its owner and purpose
  - When using a set other than **Default**, avoid excessive experimentation so that you don't add lots of ad-hoc markers to the set
  - Make sure you are aware of which marker set is exposed in the Breakpoints & Events window, as you can easily inadvertently add markers to the set that are not relevant to the code file the set was created for
  - In any set, if you have added markers that don't have to be kept, delete them to maintain the purpose of the set; this is especially true of the Default set, which can quickly accumulate redundant ad-hoc markers

### Learning Center topics

- (Alt+F1) | [Enterprise Architect](#) | [Execution Analysis](#) | [Recording Execution](#) | [Creating a Marker Set](#)
- (Alt+F1) | [Enterprise Architect](#) | [Execution Analysis](#) | [Recording Execution](#) | [Enable a Marker Set](#)

#### 16.2.3.5 Recording Activity for a Class

In addition to setting breakpoints and markers in the code editor, or creating a *marker set* through the Breakpoints & Events window, you can record all the operations of a Class or a subset using the Class Markup Selection dialog to create a marker set of record markers for those operations. These marker sets are also available to all users of the model.

**Access** [Project Browser Class context menu](#) | [Execution Analyzer](#) | [Markup Class For Recording](#)

### Reference

Field	Usage	See also
<b>Existing marker set</b>	Select to change the operations, marker types or stack depth of an existing marker set.	
<b>New marker set</b>	Select to create a new marker set.	
<b>Name</b>	Type the name of the marker set to create or edit.	
<b>Operation check boxes</b>	All checkboxes default to selected; click on the checkbox against each operation that you <i>do not</i> want to record.  Click again on the checkbox against any excluded operation that you do now want to record, to re-select it.	
<b>Include disabled operations</b>	Select this option to set a recording marker against all operations, but disable all those that you have not specifically selected above.  In the Breakpoints & Events window, you can then enable these markers individually or as a whole, rather than enabling them by editing the marker set.	<a href="#">Working with Marker Sets</a> <sup>[2541]</sup>
<b>Marker Type</b>	Click on the drop-down arrow and select the type of marker to set against each of the selected operations.  The marker type specifies the action to take when the process encounters that marker on each operation: <ul style="list-style-type: none"> <li>• Record function</li> <li>• Record stack trace</li> <li>• Break execution</li> </ul>	<a href="#">Marker Types</a> <sup>[2538]</sup>
<b>Limit recording frame depth</b>	Set the stack depth to limit the recording, to avoid ultimately producing Sequence diagrams that are too complicated to read.	<a href="#">Control Stack Depth</a> <sup>[2536]</sup>
<b>OK</b>	Click to store the marker set under the name you have specified; the set can then be loaded either before or during a session, from the Breakpoint & Markers window toolbar.	<a href="#">Breakpoint and Marker Management</a> <sup>[2224]</sup>

### Learning Center topics

- [\(Alt+F1\) | Enterprise Architect | Execution Analysis | Recording Execution | Automated Set Creation](#)

### 16.2.4 Control the Recording Session

The Record & Analyze window enables you to control a recording session. The control has a toolbar, and a history window that displays the recording history as it is captured. Each entry in this window represents a call sequence made up of one or more function calls.

**Access** [Analyzer | Recorder](#)

You must also open the Execution Analyzer window (**Analyzer | Execution Analyzer**), which lists all the scripts in the model; you must select and activate the appropriate script for the recording.

#### Learn more


- [Recorder Toolbar](#)<sup>[2544]</sup>
- [Working With Recording History](#)<sup>[2546]</sup>
- [Start Recording](#)<sup>[2547]</sup>
- [Step Through Function Calls](#)<sup>[2548]</sup>
- [Nested Recording Markers](#)<sup>[2548]</sup>
- [The Recording History](#)<sup>[2533]</sup>

#### 16.2.4.1 Recorder Toolbar







You can access facilities for starting, stopping and moderating an execution analysis recording session through the Record & Analyze window toolbar.

**Access** [Analyzer | Recorder > Toolbar](#)

#### Options

Icon	Description	See also
	<p>Display a menu of options for defining what the recording session operates on.</p> <ul style="list-style-type: none"> <li>• <b>Attach to Process</b> - enabled even if no Analyzer Script exists, this option displays a dialog through which you select a process to record and a debugging platform to use; you can also optionally select a record marker set and / or a State machine to use during the recording</li> <li>• <b>Generate Sequence Diagram from Recording</b> - generate a Sequence/State diagram from the Execution Analyzer trace</li> <li>• <b>Generate Testpoint Diagram from History</b> - generate a Test Domain diagram from the Execution Analyzer trace, that can be used with the Testpoint facility</li> <li>• <b>Generate Class Diagram from History</b> - generate a Collaboration Class diagram from the Execution Analyzer trace, depicting only those Classes and operations involved in the recorded action (use case)</li> <li>• <b>Generate All</b> - generate all three of the above diagrams from the Execution Analyzer trace</li> <li>• <b>Save as Artifact</b> - create an Artifact element that contains the current</li> </ul>	<p><a href="#">Start Recording</a><sup>[2547]</sup></p> <p><a href="#">Recording Activity for a Class</a><sup>[2542]</sup></p> <p><a href="#">Generating Sequence Diagrams</a><sup>[2549]</sup></p> <p><a href="#">Testpoint Management</a><sup>[2577]</sup></p> <p><a href="#">Combine Testpoints</a><sup>[2586]</sup></p> <p><a href="#">Artifact</a><sup>[1358]</sup></p>






Icon	Description	See also
	<p>recording history, under the currently-selected package in the the Project Browser; if you subsequently drag this Artifact element onto a Class diagram and double-click on it, the history recorded in the Artifact is copied back into the Record &amp; Analyze window</p> <ul style="list-style-type: none"> <li>• <b>Load Sequence History from file</b> - select an XML file from which to <b>restore</b> a previously-saved recording history</li> <li>• <b>Save Sequence History to file</b> - <b>save</b> the recording history to an XML file</li> </ul>	
	Select the recording stack depth for the marker set; that is, the number of frames from the point at which recording began.	<a href="#">Control Stack Depth</a> <sup>[2536]</sup> <a href="#">Marker Types</a> <sup>[2538]</sup>
	<p>Launch and record the application described in the script; you can optionally select a record marker set and / or a State machine to use during the recording.</p> <p>The icon is enabled when the active Analyzer Script is configured for debugging.</p>	<a href="#">Start Recording</a> <sup>[2547]</sup> <a href="#">Recording Activity for a Class</a> <sup>[2542]</sup>
	<p>Perform ad-hoc <b>manual</b> recording of the current thread during a debug session.</p> <p>Use this function with the 'step' buttons of the debugger; each function that is called due to a step command is logged to the history window.</p> <p>The icon is enabled if no recording is taking place and you are currently at a breakpoint (that is, debugging).</p>	
	<p>Perform ad-hoc <b>auto-recording</b> during a debug session.</p> <p>When you click on this icon, the Analyzer begins recording and does not stop until either the program ends, you stop the debugger or you click on the <b>Stop</b> icon.</p> <p>This icon is enabled if no recording is taking place and you are currently at a breakpoint (that is, debugging).</p>	
	<p>Step into a function, record the function call in the History window, and step back out.</p> <p>Enabled for <b>manual</b> recording only.</p>	<a href="#">Step Through Function Calls</a> <sup>[2548]</sup>
	Stop recording the execution trace.	

### 16.2.4.2 Working With Recording History

You can perform a number of operations on or from the results of a recording session, using the Record & Analyze window context menu.

#### Options

Option	Action	See also
<b>Show Source for Caller</b>	Display the source code, in the Source Code Viewer, for the method calling the sequence.	<a href="#">Editing Source Code</a>  [2148]
<b>Show Source for Callee</b>	Display the source code, in the Source Code Viewer, for the method being called by the sequence.	
<b>Generate Diagram for Selected Sequence</b>	Generate a Sequence diagram for a single sequence selected in the recording history.	<a href="#">Generating Sequence Diagrams</a>  [2549]
<b>Generate Sequence Diagram</b>	Generate a Sequence diagram including all sequences in the recording history.	
<b>Clear</b>	Clear the recording history currently displayed in the Record & Analyze window.	<a href="#">The Recording History</a>  [2533]
<b>Save Recording History to File</b>	Save the recording history to an XML file. A browser window displays, on which you specify the file path and name for the XML file.	
<b>Load Recording History From File</b>	Load a previously saved recording history from an XML file. A browser window displays, on which you specify the file path and name for the XML file to load.	
<b>Disable All Calls</b>	Disable every call listed in the Record & Analyze window.	
<b>Disable This Call</b>	Disable the selected call.	
<b>Disable This Method</b>	Disable the selected method.	
<b>Disable This Class</b>	Disable the selected Class.	

Option	Action	See also
<b>Disable All Calls Outside This Call</b>	Disable every call listed in the Record & Analyze window <i>except for</i> the selected call.	
<b>Enable All Calls</b>	Enable every call listed in the Record & Analyze window.	
<b>Enable This Call</b>	Enable the selected call.	
<b>Enable This Method</b>	Enable the selected method.	
<b>Enable This Class</b>	Enable the selected Class.	
<b>Help</b>	Display the Help topic for the Record & Analyze window.	<a href="#">The Recording History</a> <sup>[2533]</sup>

### 16.2.4.3 Start Recording

When you are recording execution flow as a Sequence diagram, you start the recording by selecting the **Recording** icon on the Record & Analyse window toolbar. The Record dialog displays with the recording options set to the defaults; that is, the current Breakpoint and Markers Set, the filters defined in the current Analyzer Script and the recording mode as basic.

**Access** **Analyzer | Recorder: Recording**

#### Record Dialog Options

Field/Button	Detail	See also
<b>Recording Set</b>	Recording markers determine what is recorded. If you have a recording set to use, click on the drop-down arrow and select it.	<a href="#">Working with Marker Sets</a> <sup>[2541]</sup>
<b>Additional Filters</b>	Filters are used by the debugger to exclude matching function calls from the recording history. Recording filters are defined in the Analyzer Script.  In the <b>Additional Filters</b> field you can add other filters for this specific run. if you specify more than one filter, separate them with a semi-colon.	<a href="#">Configure Recording Detail</a> <sup>[2219]</sup>
<b>Basic Recording Mode</b>	In basic mode the debugger records a history of the function calls made by the program whenever it encounters an appropriate recording marker.	

Field/Button	Detail	See also
<b>Track Instances of Named Classes</b>	In Track Instances mode the debugger also captures the creation of instances of the Classes you specify. It then includes that information in the history. The resulting Sequence diagram can then show lifelines for each instance of that Class with, where appropriate, function calls linked to the lifeline.	
<b>Track State Transitions</b>	The recording can also capture changes in State using a specified State Machine diagram. The State Machine diagram must exist as a child of a Class.  The Execution Analyzer captures instances of that Class and calculates the State of each instance whenever a function in the current recording sequence returns.	<a href="#">Reporting State Transitions</a> <sup>[2550]</sup>
<b>OK</b>	Click on this button to start the debugger.	<a href="#">Generating Sequence Diagrams</a> <sup>[2549]</sup>

#### 16.2.4.4 Step Through Function Calls

##### Topics

Topic	Detail	See also
<b>Stepping Through</b>	<p>The Step Through function can be executed by clicking on the <b>Step Through</b> button on the Record &amp; Analyze window toolbar.</p> <p>Alternatively, press ( <b>Shift+F6</b> ) or select the <b>Analyzer   Debug   Step Into</b> menu option.</p> <p>The <i>Step Through</i> command causes a <i>Step Into</i> command to be executed; if any function is detected, then that function call is recorded in the History window.</p> <p>The debugger then steps out, and the process can be repeated.</p> <p>This button enables you to record a call without having to actually step into a function; the button is only enabled when at a breakpoint and in manual recording mode.</p>	

#### 16.2.4.5 Nested Recording Markers

When a recording marker is first encountered, recording starts at the current stack frame and continues until the frame pops, recording additional frames up to the depth defined on the Recording toolbar. Consider this call sequence:

A -> B -> C -> D -> E -> F -> G -> H -> I -> J -> K -> L -> M -> N -> O -> P -> Q -> R -> S ->

If you set a recording marker at **K** and set the recording depth to **3**, this would record the call sequence:

**K -> L -> M**

If you also wanted to record the calls **X**, **Y** and **Z** as part of the Sequence diagram, you would place another recording marker at **X** and the analyzer would record:

**K -> L -> M -> X -> Y -> Z**

However, when recording ends for the **X-Y-Z** component (frame **X** is popped), recording will **resume** when frame **M** of the **K-L-M** sequence is re-entered. Using this technique can help where information from the recorded diagram would be excluded due to the stack depth, and it lets you focus on the particular areas to be captured.

#### Learn more

- [Control Stack Depth](#) <sup>[2536]</sup>
- [Recorder Toolbar](#) <sup>[2544]</sup>

## 16.2.5 Generating Sequence Diagrams


This topic describes what you might do with the recording of an execution analysis session.




**Access** **Analyzer | Recorder**

#### Use to

- Generate a Sequence diagram from a recorded execution analysis session, for:
  - all recorded sequences or
  - a single sequence in the session
- Save the recorded sequence to file
- Retrieve the saved recording and load it into the Record & Analyze window

#### Reference

Action	Detail	See also
<b>Generate a diagram</b>	<p>Select the appropriate package in the Project Browser, in which to store the Sequence diagram.</p> <p>To create the diagram from all recorded sequences, either:</p> <ul style="list-style-type: none"> <li>• Click on the <b>Recorder Menu</b> icon (  ) in the Record &amp; Analyze window toolbar, and select the <b>Generate Sequence Diagram from Recording</b> option, or</li> <li>• Right-click on the body of the window and select the <b>Generate Sequence Diagram</b> context menu option</li> </ul> <p>To create the diagram from a single sequence, either:</p>	<p><a href="#">The Recording History</a> <sup>[2533]</sup></p> <p><a href="#">Recorder Toolbar</a> <sup>[2544]</sup></p> <p><a href="#">Working With Recording History</a> <sup>[2546]</sup></p>

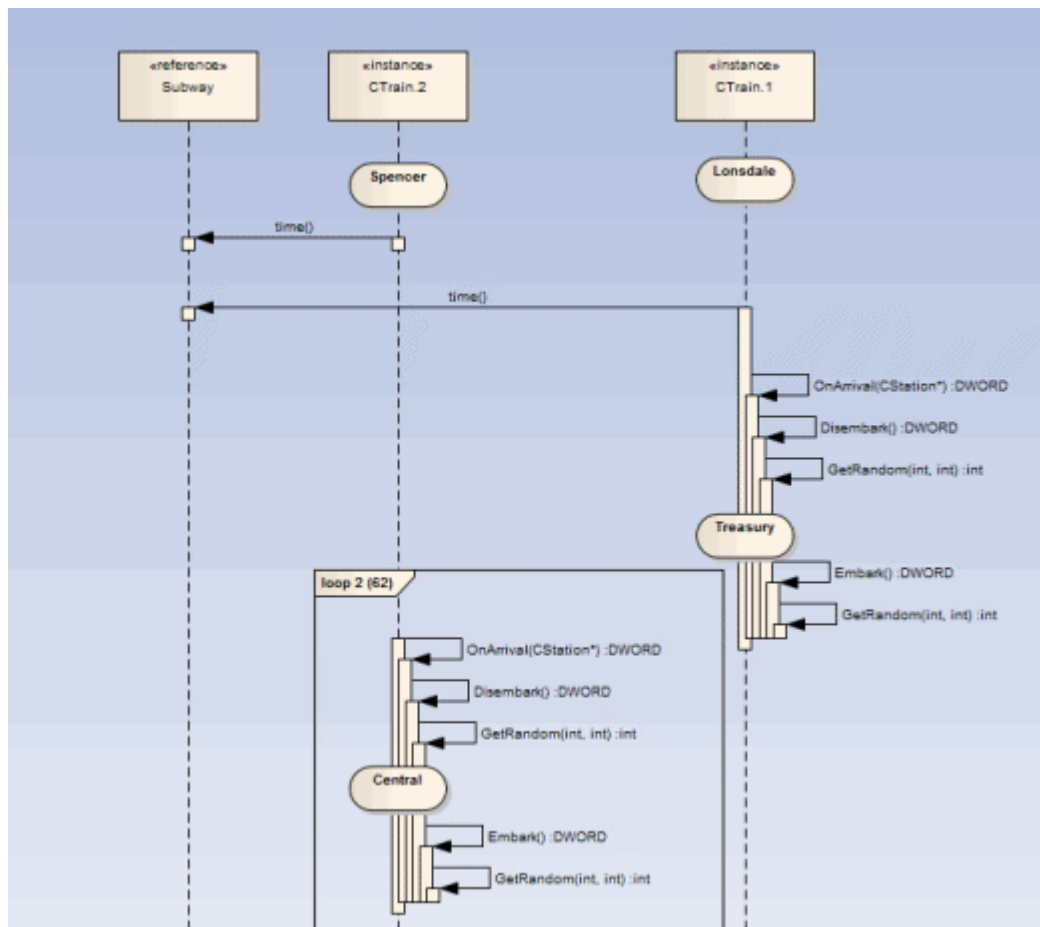
Action	Detail	See also
	<ul style="list-style-type: none"> <li>Click on the <b>Recorder Menu</b> icon (  ) in the Record &amp; Analyze window toolbar, and select the <b>Generate Sequence Diagram from Recording</b> option, or</li> <li>Right-click on the sequence and select the <b>Generate Diagram from Selected Sequence</b> context menu option</li> </ul>	
<b>Save a recorded sequence to an XML file</b>	Click on the sequence, click on the <b>Recorder Menu</b> icon (  ) in the Record & Analyze window toolbar, and select the <b>Save Sequence History to File</b> option.	
<b>Access an existing sequence XML file</b>	<p>Either:</p> <ul style="list-style-type: none"> <li>Click on the <b>Recorder Menu</b> icon (  ) in the Record &amp; Analyze window toolbar, and select the <b>Load Sequence History from File</b> option, or</li> <li>Right-click on a blank area of the screen and click on the <b>Load Sequence From File</b> context menu option</li> </ul> <p>The Windows Open dialog displays, from which you select the file to open.</p>	<a href="#">Recorder Toolbar</a> <sup>[2544]</sup>

### 16.2.6 Reporting State Transitions

This section describes how you can generate Sequence diagrams that show transitions in state as a program executes.

#### Use to

- Generate Sequence diagrams that report user-defined transitions in state as a program executes (as shown in the example generated diagram below)

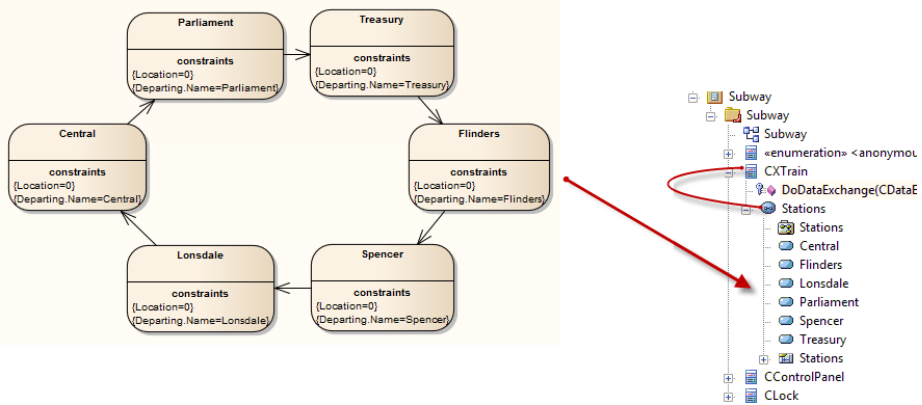


Topic	Link
Create a State Machine under the Class to be reported.	<a href="#">Reporting a State Machine</a> <sup>[2551]</sup>
Set the constraints against each State to define the change in state to be reported.	<a href="#">Recording and Mapping State Changes</a> <sup>[2553]</sup>

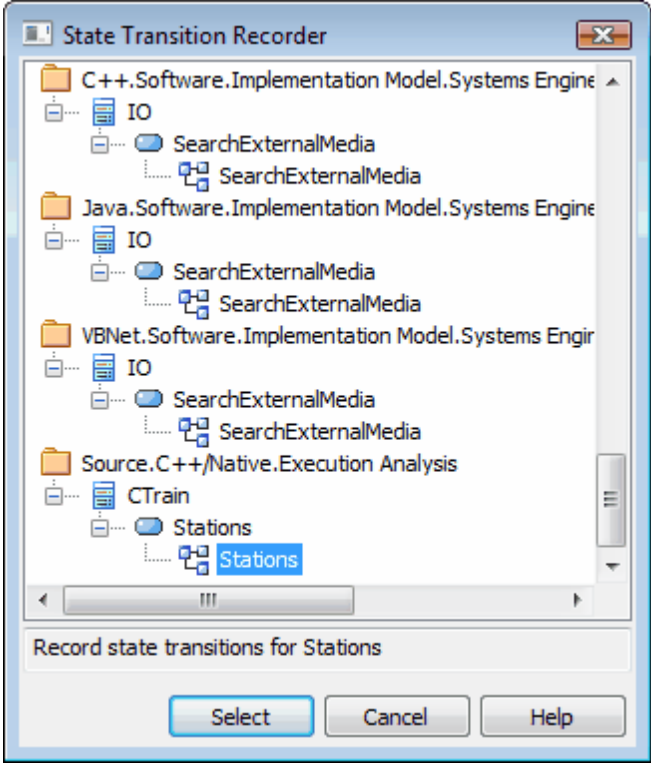
### 16.2.6.1 Reporting a State Machine

A State Machine can be used to illustrate how States change during the execution of an application, explicitly describing the transitions from State to State. You can show these State transitions on the Sequence diagrams you generate in Execution Analysis.

Topic	Detail	See also
<b>Process</b>	<p>Firstly you create a State Machine to model all the valid system states, as a child of the appropriate Class element, and on the State Machine you create the State elements that correspond to any states to be captured for your Class.</p> <p>You then set constraints in the States to record and map the State changes.</p> <p>When you record code execution for the Class, you set an option in the Record dialog and browse for the State Machine using the State Transition dialog.</p> <p>The State Transition dialog presents a list of State Machines for the entire model, in</p>	<a href="#">State Machines</a> <sup>[1203]</sup> <a href="#">Recording and Mapping State Changes</a> <sup>[2553]</sup> <a href="#">Start</a>

Topic	Detail	See also
	<p>which you locate and select the appropriate diagram (see the Example below).</p> <p>When you generate the Sequence diagram, it depicts not only the sequence but changes in State at the various points in the sequence; each Class instance participating in the detection process is displayed with its own lifeline.</p>	<p><a href="#">Recording</a> 2547</p> <p><a href="#">Reporting State Transitions</a> 2550</p>
<p><b>Example</b></p>	<p>The State Machine below, called <i>Stations</i>, shows the different States within the Melbourne Underground Loop subway system.</p> <p>A train traveling on the subway network can be stopped at any of the stations represented on the State Machine.</p> <p>The <i>Stations</i> State Machine is a child of the <i>CTrain</i> Class.</p>  <p>When you browse for the diagram in the State Transition Recorder dialog, the hierarchy shows only the root package, parent Class and child Sub Machine and diagram; no other model components are listed.</p>	





Topic	Detail	See also
		

16.2.6.2 Recording and Mapping State Changes

This topic discusses how to set constraints against each State in the State Machine under a Class, to define the change in state to be recorded.

Topics

Topic	Detail	See also									
Example	<p>The State Properties dialog shown below is for the State called <i>Parliament</i>; the Constraints tab is open to show how the State is linked to the Class <i>CXTrain</i>.</p> <p>A State can be defined by a single constraint or by many; in the example below the State <i>Parliament</i> has two constraints:</p> <div data-bbox="469 1700 1232 1870"><p>Defined Constraints   <span>New Save Delete</span></p><table><thead><tr><th>Constraint</th><th>Type</th><th>Status</th></tr></thead><tbody><tr><td>Location=0</td><td>Invariant</td><td>Approved</td></tr><tr><td>Departing.Name=Parliament</td><td>Invariant</td><td>Approved</td></tr></tbody></table></div> <p>The values of constraints can only be compared for <i>elemental</i>, <i>enum</i> and <i>string</i> types</p> <p>The <i>CXTrain</i> Class has a member called <i>Location</i> of type <i>int</i>, and a</p>	Constraint	Type	Status	Location=0	Invariant	Approved	Departing.Name=Parliament	Invariant	Approved	<p><a href="#">Reporting a State Machine</a></p>
Constraint	Type	Status									
Location=0	Invariant	Approved									
Departing.Name=Parliament	Invariant	Approved									

Topic	Detail	See also
	<p>member called <i>Departing.Name</i> of type <i>CString</i>; what this constraint means is that this State is evaluated to <b>true</b> when:</p> <ul style="list-style-type: none"> <li>• an instance of the <i>CXTrain</i> Class exists and</li> <li>• its member variable <i>Location</i> has the value <b>0</b> and</li> <li>• the member variable <i>Departing.Name</i> has the value <b>Parliament</b></li> </ul>	
<b>Operators in Constraints</b>	<p>There are two types of operators you can use on constraints to define a State:</p> <ul style="list-style-type: none"> <li>• Logical operators AND and OR can be used to combine constraints</li> <li>• Equivalence operators {= and !=} can be used to define the conditions of a constraint</li> </ul> <p>All the constraints for a State are subject to an AND operation unless otherwise specified; you can use the OR operation on them instead, so you could rewrite the constraints in the above example as:</p> <p style="padding-left: 40px;">Locat i on=0 OR</p> <p style="padding-left: 40px;">Locat i on=1 AND</p> <p style="padding-left: 40px;">Depart i ng. Name! =Cent r al</p> <p>Below are some examples of using the equivalence operators:</p> <p style="padding-left: 40px;">Depart i ng. Name! =Cent r al AND</p> <p style="padding-left: 40px;">Locat i on! =1</p>	

**Notes**

- Quotes around strings are optional; the comparison for strings is always case-sensitive in determining the truth of a constraint

**Learn more**

- [Reporting a State Machine](#) 

## 16.3 Profiling

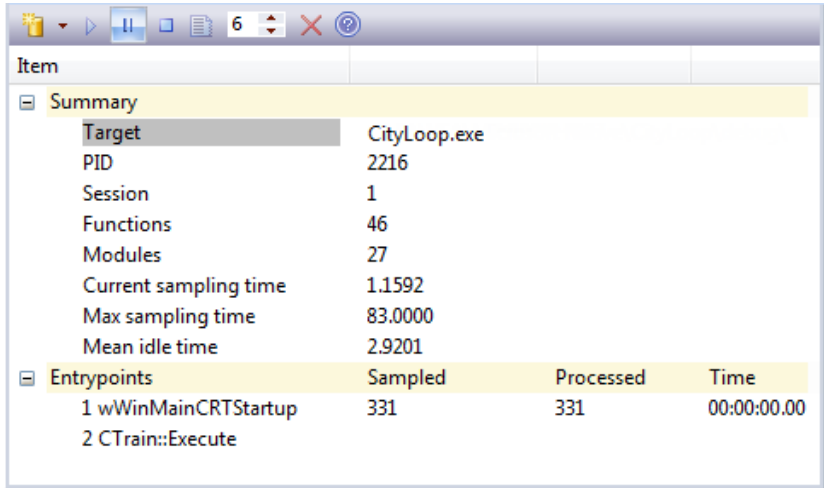
During the development of applications, you would commonly want to investigate which parts of code are taking the longest to execute. You can perform such an investigation by running the Profiler on an executing application to quickly generate a report on the most frequently called functions in the running application, tasks in the application that are taking more time than expected and which functions are taking the most time in the application.

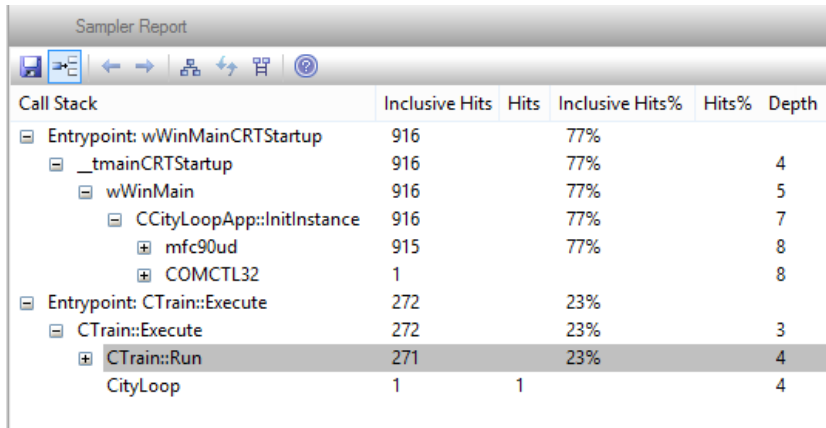
In Profiling the behavior of an application that is configured to be debugged in the Execution Analyzer, you can monitor calls in the application. Profiles can be recorded by executing an existing Analysis script that typically defines the application to build, test and debug, and any sequence recording options, or attaching to a running process. The second option is useful if the application being analyzed spawns from a call from another application (such as Winword.exe), Is running as a Service, or Is best monitored at a known point of processing (such as after a user interaction).

The Profiler reports can be stored as external binary or .xml files, which you can re-load at a later time, or internal documents attached to Artifact elements or Team Review posts, to distribute to team members.

**Access** [Analyzer | Profiler](#)

### The Profiling Process

Feature	Detail	See also
<b>Preparation</b>	The Profiler supports analysis of applications built on C, C++, Visual Basic, Microsoft .NET and Java platforms, each of which has certain pre-requisites.	<a href="#">System Requirements</a> [2556]
<b>Operation</b>	<p>When the Profiler is launched during the execution of a script, it creates a Profiler process that collects samples at regular intervals from the stacks of each thread in the target application process.</p> <p>The Profiler window displays the status details of the Profiler process as it executes. You have a number of toolbar options for starting the Profiler process, modifying its action, and pausing or stopping the process.</p>  <p>The screenshot shows the Profiler window with a toolbar at the top containing icons for file operations, execution control, and help. The main area is divided into two sections: 'Summary' and 'Entrypoints'. The 'Summary' section lists various metrics for the target application 'CityLoop.exe', including PID (2216), Session (1), Functions (46), Modules (27), Current sampling time (1.1592), Max sampling time (83.0000), and Mean idle time (2.9201). The 'Entrypoints' section is a table showing sampled and processed counts for specific functions, with '1 wWinMainCRTStartup' and '2 CTrain::Execute' listed.</p>	<a href="#">Profiler Operation</a> [2557]  <a href="#">Getting Started</a> [2558]

Feature	Detail	See also
Output	<p>The Profiler generates a report in the Call Stack, which shows how the application functions are called in relation to the application; for example, how one function might be taking most of the overall time recorded, and how that is caused by some of the sub-functions that it calls.</p> 	<a href="#">Generate, Save and Load Profile Reports</a> <sup>[2560]</sup>

### Notes

- The Profiler, or sampler, is available in the Professional, Corporate, Business and Software Engineering, System Engineering and Ultimate editions
- The Profiler can also be used under WINE (Linux and Mac) to debug standard Windows applications deployed in a WINE environment

### Learning Center topics

- [Alt+F1 | Enterprise Architect | Execution Analysis | Profiling Native Code | Introducing the Profiler](#)

## 16.3.1 System Requirements

Using the Profiler, you can analyze applications built for the following platforms:

- Microsoft™ Native (C++, C, Visual basic)
- Microsoft .NET (supporting a mix of managed and unmanaged code)
- Java

### Microsoft Native applications

For C, C++ or Visual Basic applications, the Profiler requires that the applications are compiled with the Microsoft™ Native compiler and, for each application or module to be analyzed, a PDB file is available. The Profiler can sample both debug and release configurations of an application, provided that the PDB file for each executable exists and is up to date.

### Microsoft .NET applications

For Microsoft .NET applications, the Profiler requires that the appropriate Microsoft .NET framework is installed and, for each application or module to be analyzed, a PDB file is available.

### Java

For Java, the Profiler requires that the appropriate JDK from Oracle is installed. If you have configured an analyzer script for debugging, you can select the **Run** button. Alternatively, you can attach the Profiler to a running virtual machine; in this case, the Profiler requires that the JVM be hosting the Enterprise Architect profiling agent. For example:

```
java.exe -cp "%classpath%" -agent path: "C:\Program Files (x86)\Sparx Systems\EA\vea\x86\ssamplerlib32" myapp
```

or

```
java.exe -cp "%classpath%" -agent path: "C:\Program Files (x86)\Sparx Systems\EA\vea\x64\ssamplerlib64" myapp
```

(Refer to the JDK documentation for details of the -agent path VM startup option.)

### Learn more

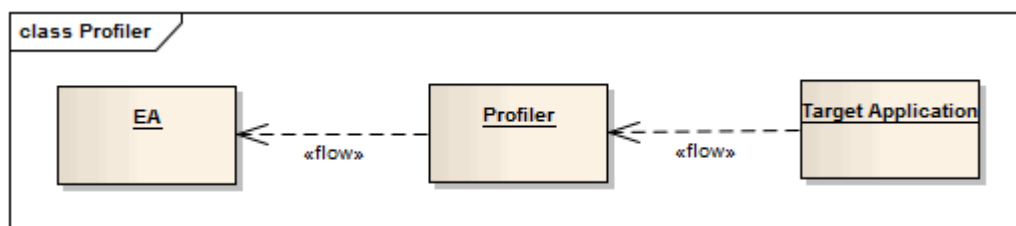
- [Debugging](#) <sup>[2222]</sup>
- [Microsoft C++ and Native \(C, VB\)](#) <sup>[2189]</sup>
- [Java](#) <sup>[2191]</sup>
- [.NET](#) <sup>[2200]</sup>
- [General Setup](#) <sup>[2189]</sup>
- [Profiling](#) <sup>[2555]</sup>
- [Getting Started](#) <sup>[2558]</sup>

### Learning Center topics

- **Alt+F1 | Enterprise Architect | Execution Analysis | Profiling Native Code**




## 16.3.2 Profiler Operation

Whenever you launch the Profiler or attach it to a running application, a Profiler process is started. The Profiler operates by taking samples from the target process at intervals of up to 250 milliseconds. At these intervals the Profiler interrupts the process and collects stack information for all threads running at that time; this information is sent back to Enterprise Architect where it is collected, sorted and stored.



**Access** **Analyzer | Profiler**

Actions









Action	Detail	See also
<b>Start a Profiler Process</b>	Click on the <b>Run Profiler</b> button on the Profiler window toolbar.	
<b>End a Profiler Process</b>	<p>The sampler process exits if:</p> <ul style="list-style-type: none"> <li>You click on the <b>Stop</b> button</li> <li>The target application terminates, or</li> <li>You close the current model</li> </ul> <p>If you stop the Profiler and the process is still running, you can quickly attach to it again.</p>	
<b>Pause and resume a session</b>	<p>You can pause and resume profile sample collection at any time during a session.</p> <p>When sampling is turned on or resumed, the Profiler process becomes active and samples are collected from the target; resuming sampling collects <b>completely new</b> samples.</p> <p>The Profiler process idles if sampling is turned off or paused during a session. The <b>Report</b> and <b>Discard Data</b> buttons then become enabled.</p>	<a href="#">Getting Started</a>  <sup>2558</sup>
<b>Generate report</b>	<p>Click on the <b>Report</b> button to display the Profiler report.</p> <p>Having reviewed this report, you can also generate from it a report on the execution of an individual function.</p>	<a href="#">Generate, Save and Load Profile Reports</a>  <sup>2560</sup> <a href="#">Function Line Reports</a>  <sup>2564</sup>
<b>Clear results</b>	You can clear any sample data collected for the target application and begin again - click on the <b>Discard Data</b> button to discard the samples currently collected in the Profiler window.	

**16.3.3 Getting Started**

When you run a Profiling session, almost every option you might need is available from the Profiler window toolbar. You can, for example, initiate the profiling session, attach to an already-running process, pause and resume profiling, stop the session, generate and view the Profile report or load a previously-generated report. You can also set Profiler options to modify the operation of the Profiler.

Access   **Analyzer | Profiler**

Toolbar Options

Icon	Action	See also
	<p>Set Profiler options, using a drop-down menu; the options are:</p> <ul style="list-style-type: none"> <li>• <b>Attach to Running Process</b> - attach to and profile a process that is already running</li> <li>• <b>Switch to debugger</b> - (enabled when you are running the Profiler) end the profiling session and attach the <b>debugger</b> to the running process; available on Microsoft Native and Microsoft .NET platforms</li> <li>• <b>Load Report</b> - load and display a previously-generated report from an XML disk file</li> <li>• <b>Analyzer Scripts ( Shift+F12 )</b> - display the Execution Analyzer window to create or edit scripts and configure the debugger</li> <li>• <b>Start Sampling Immediately</b> - begin sample collection immediately upon either process start (main thread entry point executed) or attachment of process by the Profiler</li> <li>• <b>Capture Debug Output</b> - capture any appropriate debug output and redirect it to the System Output window</li> <li>• <b>Stop Process on Exit</b> - select to terminate the target process when the Profiler is stopped</li> </ul>	<p><a href="#">Debugger Facilities</a> <small>[2231]</small></p> <p><a href="#">Analyzer Scripts</a> <small>[2175]</small></p>
	(When an application is configured for the package) create the Profiler process, which launches the configured application.	<a href="#">Managing Scripts</a> <small>[2175]</small>
	<p>When the application is running, pause and resume sample capture.</p> <p>Pausing sampling activates the <b>Report</b> and <b>Discard Data</b> buttons.</p>	
	Stop the Profiler process; if any samples have been collected, the <b>Report</b> button is enabled.	
	Generate a report on the current number of samples collected.	<a href="#">Generate, Save and Load Profile Reports</a> <small>[2560]</small>
	Set the interval, in milliseconds, at which samples are taken of the target process; the range of possible values is <b>1 - 250</b> .	<a href="#">Setting Options</a> <small>[2563]</small>
	Discard the collected data. You are prompted to confirm the discard.	
	Display the Help topic for this window.	

### Learning Center topics

- [Alt+F1 | Enterprise Architect | Execution Analysis | Profiling Native Code | Profile Application Startup](#)
- [Alt+F1 | Enterprise Architect | Execution Analysis | Profiling Native Code | Profile Running Application](#)
- [Alt+F1 | Enterprise Architect | Execution Analysis | Profiling Native Code | Introducing the Profiler | Set Capture Options](#)


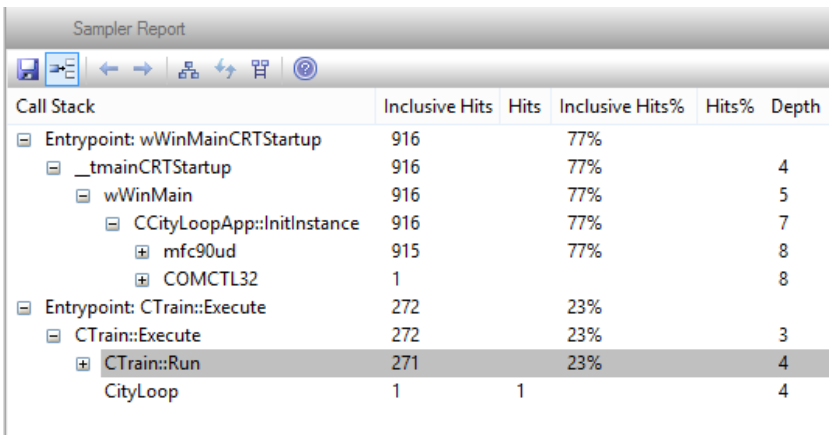

## 16.3.4 Generate, Save and Load Profile Reports

When you profile the execution of your MS Native Windows application, the Profiler generates a report in the Call Stack that shows how functions are called in relation to the profiled application; for example, how one function might be taking most of the overall time recorded, and what contributions are made by the subfunctions that function calls. You have a number of options for reviewing and sharing the resulting report. You might:






- View the report immediately, in detailed or summary form
- File the report in binary or xmi format
- Reload the report at a later time
- Refine the information displayed in the report
- Distribute the report as a Team Review resource
- Attach the report as a document to an Artifact element




**Access** [Analyzer | Profiler \(execute\): View Report](#)

### Options

Action	Detail	See also
<b>Display Report</b>	<p>When you have run the Profiler, click on the  icon in the <b>Profiler</b> toolbar. The generated report displays on the Call Tree Report tab of the Call Stack .</p>  <p>This report can be very detailed, especially if every level is expanded. You</p>	<a href="#">Getting Started</a>  <sup>2558</sup>



Action	Detail	See also																																										
	<p>can display a summary of the report by clicking on the Summary Report tab of the Call Stack.</p> <table> <tr> <th>Name</th><th>Inclusive Hits</th><th>Occurrences</th></tr> <tr> <td>wWinMainCRTStartup</td><td>1258</td><td>1</td></tr> <tr> <td>wWinMain</td><td>1258</td><td>1</td></tr> <tr> <td>__tmainCRTStartup</td><td>1258</td><td>1</td></tr> <tr> <td>CNetwork::WindowProc</td><td>24</td><td>3</td></tr> <tr> <td>CNetwork::IsRunning</td><td>10</td><td>1</td></tr> <tr> <td>std::_Vector_const_iterator&lt;CTrain *,std::allocator&lt;CTr...</td><td>4</td><td>2</td></tr> <tr> <td>std::_Vector_iterator&lt;CTrain *,std::allocator&lt;CTrain *&gt; ...</td><td>4</td><td>2</td></tr> <tr> <td>std::vector&lt;CTrain *,std::allocator&lt;CTrain *&gt; &gt;::begin</td><td>3</td><td>1</td></tr> <tr> <td>CControlPanel::UpdateTime Table</td><td>2</td><td>1</td></tr> <tr> <td>CCityLoopDlg::WindowProc</td><td>2</td><td>1</td></tr> <tr> <td>CTrain::Run</td><td>2</td><td>1</td></tr> <tr> <td>CTrain::Execute</td><td>2</td><td>1</td></tr> <tr> <td>std::_RanIt&lt;CTrain *,int,CTrain * const *,CTrain * const ...</td><td>2</td><td>1</td></tr> </table> <p>You can filter and reorganize the information in the report, in the same way as you do for the results of a Model Search.</p>	Name	Inclusive Hits	Occurrences	wWinMainCRTStartup	1258	1	wWinMain	1258	1	__tmainCRTStartup	1258	1	CNetwork::WindowProc	24	3	CNetwork::IsRunning	10	1	std::_Vector_const_iterator<CTrain *,std::allocator<CTr...	4	2	std::_Vector_iterator<CTrain *,std::allocator<CTrain *> ...	4	2	std::vector<CTrain *,std::allocator<CTrain *> >::begin	3	1	CControlPanel::UpdateTime Table	2	1	CCityLoopDlg::WindowProc	2	1	CTrain::Run	2	1	CTrain::Execute	2	1	std::_RanIt<CTrain *,int,CTrain * const *,CTrain * const ...	2	1	<p><a href="#">Customizing the Search View</a> <small>[708]</small></p>
Name	Inclusive Hits	Occurrences																																										
wWinMainCRTStartup	1258	1																																										
wWinMain	1258	1																																										
__tmainCRTStartup	1258	1																																										
CNetwork::WindowProc	24	3																																										
CNetwork::IsRunning	10	1																																										
std::_Vector_const_iterator<CTrain *,std::allocator<CTr...	4	2																																										
std::_Vector_iterator<CTrain *,std::allocator<CTrain *> ...	4	2																																										
std::vector<CTrain *,std::allocator<CTrain *> >::begin	3	1																																										
CControlPanel::UpdateTime Table	2	1																																										
CCityLoopDlg::WindowProc	2	1																																										
CTrain::Run	2	1																																										
CTrain::Execute	2	1																																										
std::_RanIt<CTrain *,int,CTrain * const *,CTrain * const ...	2	1																																										
<b>Show or hide unknown frames</b>	<p>By default the report excludes unknown function calls and frames in the call tree. Click on the  icon in the Call Stack toolbar to show and hide these frames in the report.</p>																																											
<b>Display the previous or next incidence of a function</b>	<p>When you select a node representing a function, and that function call is present in more than one stack, you can click on the  and  icons in the Call Stack toolbar to navigate between all stacks in which the function appears. Navigation is from highest use to lowest.</p>																																											
<b>Set a Node as the root</b>	<p>If you want to examine a deeply-nested branch, you can make it the root node of the report. To do this, click on the node and click on the  icon in the Call Stack toolbar. Only function calls from the selected branch will be displayed.</p>																																											
<b>Reset the Root Node</b>	<p>To return the report to displaying the branch from the node on which the report was originally generated, click on the  icon in the Call Stack toolbar.</p>																																											

Action	Detail	See also
<b>Generate a Sequence Diagram</b>	<p>You can generate a Sequence diagram from any node in the Profiler report. To do this, either:</p> <ul style="list-style-type: none"> <li>Right-click on the node and select the <b>Create Sequence Diagram</b> context menu option, or</li> <li>Click on the node and click on the  icon in the Call Stack toolbar</li> </ul> <p>The generated Sequence diagram reflects all activity resulting from the selected node. It is created as a child diagram of the Interaction corresponding to the node, and is immediately displayed in the Diagram View.</p>	
<b>Save Report to File</b>	<p>Either:</p> <ul style="list-style-type: none"> <li>Click on the  icon in the Call Stack toolbar, or</li> <li>Right-click on the Call Stack and select the <b>Save Report to File</b> context menu option</li> </ul> <p>The Save As dialog displays. Browse to the appropriate file location and, in the <b>File name</b> field, type the file name.</p> <p>In the <b>Save as type</b> field, select the file type you want to use, either binary (.ssprf) or .xmi.</p> <p>Click on the <b>Save</b> button.</p>	
<b>Load a Saved Report</b>	<p>Click on the  icon in the <b>Profiler Toolbar</b> and select the <b>Load Report</b> context menu option.</p> <p>The Open dialog displays, on which you browse for and select the report file.</p> <p>Click on the <b>Open</b> button; the Call Stack opens or refreshes with the loaded report.</p>	
<b>Generate a Function Line report</b>	<p>In the Sampler report, right-click on the name of the function to analyze, and select the <b>Create Line report for function</b> menu option.</p> <p>Once the Profiler binds the method, the line report is opened on the Sampler Report window.</p>	<a href="#">Function Line Reports</a> <sup>[2564]</sup>
<b>Make Report a Team Review Resource</b>	<p>You can save any current report as a resource for a Category, Topic or Post in the Team Review to share and review at any time, as it is saved with the model. The report can also be compared with future runs.</p> <p>To begin this process, select the menu option <b>Team Review Context Menu   Share Resource   Add Active Profiler Report</b>.</p>	<a href="#">Work on Team Review Items</a> <sup>[346]</sup>
<b>Attach Report to an Artifact</b>	<p>In the Project Browser, select the Package or element under which to create the Artifact element.</p>	<a href="#">Artifact</a> <sup>[1358]</sup>

Action	Detail	See also
<b>Element</b>	<p>On the Call Stack window showing the report, right-click and select the <b>Save Report to Artifact</b> context menu option. You are prompted to provide a name for the report (and element); type this in and click on the <b>OK</b> button.</p> <p>The Artifact element is created in the Project Browser, underneath the selected package or element.</p> <p>If you add the Artifact to a diagram as a simple link, when you double-click on the element the Call Stack window displays, showing the saved report.</p>	

### Notes

- If you add the Profiler report to an Artifact element and also attach a linked document, the Profiler report takes precedence and is displayed when you double-click on the element; you can display the linked document using the **Edit Linked Document** context menu option

### Learn more



- [Profiling](#) 

### Learning Center topics

- **Alt+F1 | Enterprise Architect | Execution Analysis | Profiling Native Code | View Report**
- **Alt+F1 | Enterprise Architect | Execution Analysis | Profiling Native Code | Load Report from Disk**

## 16.3.5 Setting Options

### Topics

Topic	Icon	Detail	See also
<b>Interval</b>		Set the interval, in milliseconds, at which samples are taken of the target process; the range of possible values is <b>1 - 250</b> .	
<b>Profile Options</b>		<p>Set Profiler options, using a drop-down menu; the options include:</p> <ul style="list-style-type: none"> <li>• <b>Start Sampling Immediately</b> - begin sample collection immediately upon either process start (main thread entry point executed) or attachment of process by Profiler</li> <li>• <b>Capture Debug output</b> - capture any appropriate debug output and redirect it to the Enterprise Architect Output window</li> <li>• <b>Stop Process on Exit</b> - select to terminate the target process when the Profiler is stopped</li> </ul>	

### Learning Center topics



- [Alt+F1 | Enterprise Architect | Execution Analysis | Profiling Native Code | Introducing the Profiler | Set Capture Options](#)

### 16.3.6 Start & Stop the Profiler

For most debugging operations it is necessary to have first configured an Execution Analyzer Script that typically defines the application to build, test and debug, and any sequence recording options. It is possible to use the Profiler without doing any of this by using the **Attach to Process** option in the drop-down option list.

If the application to profile is the one defined in the current Package, use the **Run Profiler** button.



-  - (When an application is configured for the Package) create the Profiler process, which launches the configured application
-  - Stop the Profiler process

#### Learn more

- [Managing Scripts](#) 

#### Learning Center topics

- [Alt+F1 | Enterprise Architect | Execution Analysis | Profiling Native Code | Profile Running Application](#)

### 16.3.7 Function Line Reports

After you have run the Profiler on an executing application and generated a Sampler report, you can further analyze the activity of a specific **function** listed in the Sampler report by generating a **function line report** from that report item. A function line report shows the number of times each **line** of the function was executed. You produce one function line report at a time, on any method in the Sampler report that has a valid source file. The line report is particularly useful for functions that perform **loops** containing **conditional branching**; the coverage can provide a picture of the most frequently and least frequently executed portions of code within a single method.

The line report you generate is saved when you save the Sampler report. The **body** of the function is also saved with the line report to preserve the function state at that time.

**Platforms supported**    Java, Microsoft .NET and Microsoft native code

#### Create a Line Report

In the Sampler report, right-click on the name of the function to analyze, and select the **Create Line report for function** menu option.

Once the Profiler binds the method, the line report is opened on the Sampler Report window. The report shows the body of the function, including line numbers and text. As each line is executed a hit value will accumulate against that line. A timer will update the report approximately once every second.

Call Tree Report   Summary Report   ConsoleApplication::CQuickSort::Quicksort		
Module: ConsoleApplication Function: CQuickSort::Quicksort Date: 20/09/2013 2:53:21 PM Author: smeagher Iterations: 28679		
LineNo	Hits	Code
21	28645	{
22	28644	if (r <= l)
23	14460	return;
24	14184	int i = l-1, j = r, p = l-1, q = r;
25		for (,;)
26		{
27	439580	while (a[++i] < a[r]) ;
28	14185	while (a[--j] > a[r])
29		if (j == i)
30		break;
31		if (i >= j)
32	14185	break;
33		
34		Exchange(a, i, j);
35		if ( a[i] == a[r])
36		Exchange(a, ++p, i);
37		
38		if ( a[j] == a[r])
39		Exchange(a, j, --q);
40		
41		}
42	14185	Exchange(a, i, r);
43	14185	j = i-1; i = i+1;
44	14185	for (int k = l; k < p; k++, j--)

### End Line Report Capture

Once enough information is captured, or the function has ended, click on the Profiler toolbar **Stop** button to stop recording the capture.

### Save Reports

Use the **Save** button on the Call Stack toolbar to save the Sampler report and any function line reports to a file.

### Delete Line Reports

Closing the line report tab will close that report but the report data will only be deleted when the report is saved.

[Learn more](#)

- [Generate, Save and Load Profile Reports](#) 

### 16.3.8 *Save Report in Team Review*

You can save any current report as a resource for a Category, Topic or Document in the Team Review. The report can then be shared and reviewed at any time as it is saved with the model. This helps you to:

- Preserve a profiler report to compare against future runs
- Allow other people to investigate the profile

**[Access](#)**   **Team Review Context Menu | Share Resource | Add Active Profiler Report**

[Learn more](#)

- [Work on Team Review Items](#) 

## 16.4 Object Workbench

This section describes the Object Workbench, a tool in Enterprise Architect debugging that enables you to create your own Class workbench instances and invoke methods on them.

Topic	Link
Setup requirements for the workbench.	<a href="#">Workbench Setup</a> <sup>[2216]</sup>
Overview of the workbench as a debugger tool.	<a href="#">How it Works</a> <sup>[2567]</sup>
How to create and delete workbench instances.	<a href="#">Create &amp; Delete Workbench Instances</a> <sup>[2568]</sup>
How to invoke methods to record the Stack trace and produce Sequence diagrams.	<a href="#">Invoke Methods</a> <sup>[2570]</sup>

### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Execution Analysis | Object Workbench**

### 16.4.1 How it Works

The Workbench is a tool in Enterprise Architect Debugging, enabling you to create your own variables and invoke methods on them. Stack trace can be recorded and Sequence diagrams produced from the invocation of such methods. It provides a quick and simple way to debug your code.

### Topics

Topic	Detail	See also
<b>Mode</b>	<p>The Workbench operates in two modes:</p> <ul style="list-style-type: none"> <li>• Idle mode - When the Workbench is in idle mode, instances can be created and viewed and their members inspected</li> <li>• Active mode - When methods are invoked on an instance, the Workbench enters Active mode and the displayed instances change if the debugger encounters any breakpoints; if no breakpoints are set, then the instances do not change and the Workbench immediately returns to Idle mode</li> </ul>	
<b>Logging</b>	The results of creating instances and the results of calls on their methods are displayed in the Debug window.	<a href="#">Run the Debugger</a> <sup>[2237]</sup>

### Learn more

- [Workbench Setup](#) <sup>[2216]</sup>
- [Create & Delete Workbench Instances](#) <sup>[2568]</sup>
- [Invoke Methods](#) <sup>[2570]</sup>

## 16.4.2 Create & Delete Workbench Instances

This topic explains how to create and delete a workbench instance for any Class in your model, for which a script has been built and executed.

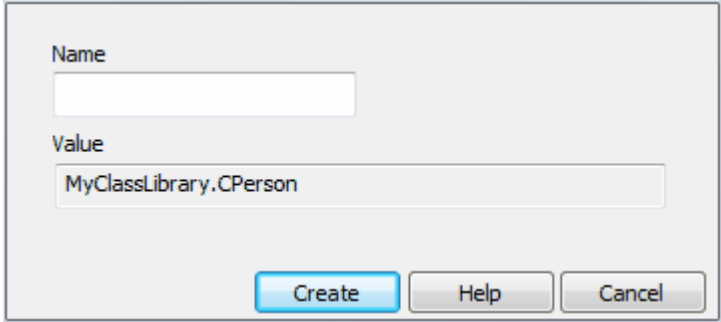
The Workbench supports the following workbench platforms:

- Microsoft .NET (version 2.0 or later) excluding native C++, C and VB
- Java (JDK 1.4 or later)

The Execution Analysis script for the package must have a debugger configured.

**Access** **Project Browser Class context menu | Execution Analyzer | Create Workbench Instance of Class (Ctrl+Shift+J)**  
**Diagram Class context menu | Execution Analyzer | Create Workbench Instance of Class**

### Topics

Topic	Detail	See also
<b>Naming the Workbench</b>	<p>Select the <b>Create Workbench Instance of Class</b> option</p> <p>The Workbench dialog displays</p>  <p>Type in a name for the new instance (for example, <b>Catherine</b>); each instance name must be unique for the workbench.</p>	
<b>Choosing a Constructor</b>	<p>Click on the <b>Create</b> button</p> <p>The following dialog displays:</p>	



Topic	Detail	See also				
	<div><div><div>Catherine(MyClassLibrary::CPerson)</div><div><div>CPerson()</div><div>CPerson()</div><div>CPerson(MyClassLibrary.CPerson)</div><div>CPerson(string,string,int)</div></div></div></div> <p>Select the constructor from the drop-down list.</p> <p>If you select a constructor with parameters, a set of fields displays in which you enter the value for each parameter (see <i>Enter Parameters</i>, below).</p> <p>If you do not define a constructor, or define a single constructor taking no parameters, go straight to the row entitled <i>Invoke Workbench Instance</i>.</p>					
Enter Parameters	<p>In the dialog above, type any parameters required by the constructor:</p> <ul style="list-style-type: none"><li><b>Literals as arguments</b><ul style="list-style-type: none"><li>Text: abc or "abc" or "a b c"</li><li>Numbers: 1 or 1.5</li></ul></li><li><b>Objects as arguments</b><p>If an argument is not a literal then you can supply it in the list only if you have already created an instance of that type in the workbench; you do this by typing the name of the instance as the argument.</p><p>The debugger checks any name entered in an argument against its list of workbench instances, and substitutes that instance in the actual call to the method.</p></li><li><b>Strings as arguments</b><p>Surrounding strings with quotes is unnecessary as anything you type for a string argument becomes the value of the string; for example, the only time you should surround strings with quotes is in supplying elements of a string array, or where the string is equal to the name of an existing workbench instance.</p><p>"A b c"</p><p>"a b \$ % 6 4"</p><p>A b c d</p><p>As 5 7 ) 2 === 4</p></li><li><b>Arrays as arguments</b><p>Enter the elements that compose the array, separated by commas:</p><table><tr><td>Type</td><td>Arguments</td></tr><tr><td>String( )</td><td>one,two,three,"a book","a bigger book"</td></tr></table></li></ul>	Type	Arguments	String( )	one,two,three,"a book","a bigger book"	
Type	Arguments					
String( )	one,two,three,"a book","a bigger book"					

Topic	Detail	See also
	<p>CPerson( )      Tom,Dick,Harry</p> <p>If you enter text that matches the name of an existing instance, surround it in quotes to avoid the debugger passing the instance rather than a string.</p> <pre>void SetName(string)</pre> <p>string      "Bill"</p> <p>Invoke</p>	
<b>Invoke Workbench Instance</b>	<p>Click on the <b>Invoke</b> button to commit the workbench instance.</p> <p>The instance then displays in the Workbench window, showing the variables of the instance in a hierarchy of the type and value of the instance and of any members.</p>	
<b>Delete Workbench Instances</b>	<p>You can delete instances using the <b>Delete</b> shortcut menu on any instance on the Workbench.</p> <p>If all instances are deleted, the debugger is shut down and the Workbench window is closed.</p>	

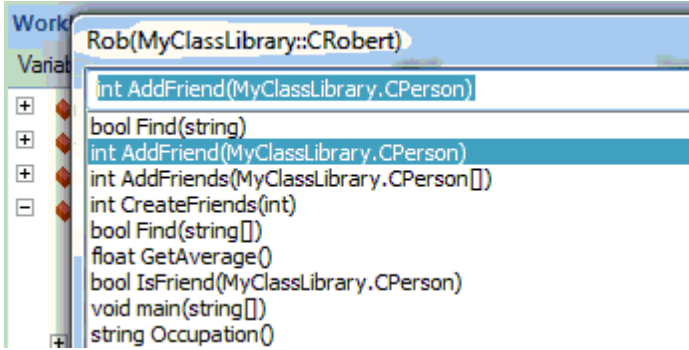
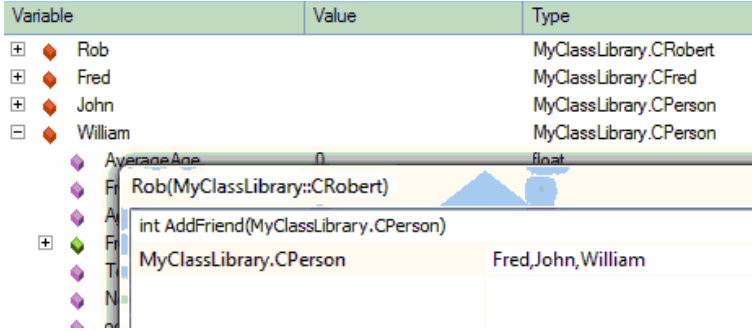
#### Learning Center topics


- (Alt+F1) | [Enterprise Architect](#) | [Execution Analysis](#) | [Object Workbench](#) | [Creating Variables](#)

### 16.4.3 Invoke Methods

#### Topics

Topic	Detail	See also
<b>Access</b>	On the Workbench window, right-click on the instance on which to execute a method, and select the <b>Invoke</b> context menu option.	
<b>Choose Method</b>	<p>A list of methods for the type are presented in a dialog; select a method from the list and click on the <b>Invoke</b> button.</p> <p>Note that all methods listed are public; private methods are not available.</p>	

Topic	Detail	See also
		
<b>Supply Arguments</b>	<p>In the example, the instance or variable <i>Rob</i> has been created, of type <i>MyClassLibrary.CRobert</i>, and the method <i>AddFriends</i> invoked that takes an array of <i>CPerson</i> objects as its only argument.</p> <p>You now supply to it three other Workbench instances: <i>Fred</i>, <i>John</i> and <i>William</i>.</p> 	
<b>Arguments</b>	<p>In the dialog above, type any parameters required by the constructor:</p> <ul style="list-style-type: none"> <li> <b>Literals as arguments</b> <ul style="list-style-type: none"> <li>Text: abc or "abc" or "a b c"</li> <li>Numbers: 1 or 1.5</li> </ul> </li> <li> <b>Objects as arguments</b> <p>If an argument is not a literal then you can supply it in the list only if you have already created an instance of that type in the workbench; you do this by typing the name of the instance as the argument.</p> <p>The debugger checks any name entered in an argument against its list of workbench instances, and substitutes that instance in the actual call to the method.</p> </li> <li> <b>Strings as arguments</b> </li> </ul>	

Topic	Detail	See also						
	<p>Surrounding strings with quotes is unnecessary as anything you type for a string argument becomes the value of the string; for example, the only time you should surround strings with quotes is in supplying elements of a string array, or where the string is equal to the name of an existing workbench instance.</p> <p>"A b c"</p> <p>"a b \$ % 6 4"</p> <p>A b c d</p> <p>As 5 7 ) 2 === 4</p> <ul style="list-style-type: none"><li>• <b>Arrays as arguments</b></li></ul> <p>Enter the elements that compose the array, separated by commas:</p> <table><tr><td>Type</td><td>Arguments</td></tr><tr><td>String( )</td><td>one,two,three,"a book","a bigger book"</td></tr><tr><td>CPerson( )</td><td>Tom,Dick,Harry</td></tr></table> <p>If you enter text that matches the name of an existing instance, surround it in quotes to avoid the debugger passing the instance rather than a string.</p> <div><div>void SetName(string)</div><div><div>string</div><div>"Bill"</div></div><div><div>Invoke</div></div></div>	Type	Arguments	String( )	one,two,three,"a book","a bigger book"	CPerson( )	Tom,Dick,Harry	
Type	Arguments							
String( )	one,two,three,"a book","a bigger book"							
CPerson( )	Tom,Dick,Harry							
Invoke	<p>Having chosen the constructor and supplied any arguments, click on the <b>Invoke</b> button to create the variable.</p> <p>Output confirming this action is displayed in the Debug window.</p>	<a href="#">Run the Debugger</a> 						

#### Learning Center topics

- (Alt+F1) | [Enterprise Architect](#) | [Execution Analysis](#) | [Object Workbench](#) | [Invoke Methods](#)

## 16.5 Unit Testing

Enterprise Architect supports integration with unit testing tools in order to make it easier to develop good quality software.

In sequence:

- You download and install the NUnit and JUnit applications (JUnit - <http://www.junit.org/> NUnit - <http://www.nunit.org/index.php?p=home>) - Enterprise Architect does not include these applications in the installer
- Enterprise Architect helps you to create test Class stubs with the **JUnit** and **NUnit** transformations
- You define your test code within the Class stubs
- You set up and run a **test script** against any package
- All test results are automatically recorded inside Enterprise Architect

### Learn more

- [JUnit Transformation](#) <sup>[2037]</sup>
- [NUnit Transformation](#) <sup>[2038]</sup>
- [Set Up Unit Testing](#) <sup>[2573]</sup>
- [Add Testing Command](#) <sup>[2573]</sup>
- [Run Unit Tests](#) <sup>[2573]</sup>
- [Record Test Result](#) <sup>[2573]</sup>

### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Execution Analysis | xUnit Tests**

### 16.5.1 Set Up Unit Testing

This topic explains the actions you should take in setting up Unit Testing, after having downloaded and installed the JUnit and/or NUnit applications.

### Topics

Topic	Detail	See also
<b>Create Unit Test Stubs</b>	<p>By using the <b>JUnit</b> or <b>NUnit</b> transformations and code generation you can create test method stubs for all of the public methods in each of your Classes.</p> <pre>( Test Fixture ) public class CalculatorTest {     ( Test )     public void testAdd() {     }      ( Test )     public void testDivide() {     } }</pre>	<a href="#">Unit Testing</a> <sup>[2573]</sup> <a href="#">JUnit Transformation</a> <sup>[2037]</sup> <a href="#">NUnit Transformation</a> <sup>[2038]</sup> <a href="#">Generate Source Code</a> <sup>[2111]</sup>

Topic	Detail	See also
	<pre> ( Test ) public void testMultiply(){  }  ( Test ) public void testSubtract(){  }  } </pre>	
<b>Define Test Cases</b>	<p>Write your unit test in the generated code stubs (either in Enterprise Architect or your preferred IDE).</p> <p>The following is an <i>NUnit</i> example in C#, although it could also be any other .NET language, or Java and JUnit.</p> <pre> ( Test Fixture ) public class CalculatorTest {      ( Test )     public void testAdd(){         Assert.AreEqual ( 1+1, 2 );     }      ( Test )     public void testDivide(){         Assert.AreEqual ( 2/2, 1 );     }      ( Test )     public void testMultiply(){         Assert.AreEqual ( 1*1, 1 );     }      ( Test )     public void testSubtract(){         Assert.AreEqual ( 1-1, 1 );     }  } </pre> <p>Alternatively, if you have not performed an xUnit transformation, you can reverse engineer the above code into Enterprise Architect so that Enterprise Architect can record all test results against this Class.</p>	
<b>Compile Your Code</b>	Ensure the source code being tested is compiled without errors, so that the test scripts can be run against it.	
<b>Set up the Test Scripts</b>	Set up the Test scripts against the required package, and then run the tests.	<a href="#">Add Testing Command</a> <sup>[2162]</sup> <a href="#">Run Unit Tests</a> <sup>[2575]</sup>

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Execution Analysis | xUnit Tests | Set up Unit Tests**

## 16.5.2 Run Unit Tests

On running a test script you generate test results that are stored as Test Cases against the Classes being tested.

**Access** [Analyzer | Test | Run Test Script](#)

### Topics

Topic	Detail	See also
<b>Run Tests</b>	<p>Select the appropriate package in the Project Browser.</p> <p>Select the <b>Run Test Script</b> option to run the test script you previously set up for that package, in the Execution Analyzer.</p>	<a href="#">Set Up Unit Testing</a> <sup>[2573]</sup> <a href="#">Add Testing Command</a> <sup>[2182]</sup>
<b>View Results</b>	<p>The results of xUnit tests are displayed in the Output window, identifying which tests have run and which of these have failed.</p> <p>The results also show which method failed and the file and line number the failure occurred at.</p> <p>Double-click on an error message; Enterprise Architect opens the editor to that line of code, enabling you to quickly find and fix the error.</p> <p>Enterprise Architect also records the run status of each test against the Class being tested; these are stored in the element Test Cases.</p> <p>A diagram containing the Class can be set to display these Test Cases, by exposing the test scripts compartment on the diagram elements.</p>	<a href="#">Record Test Results</a> <sup>[2575]</sup> <a href="#">Show Test Scripts Compartment</a> <sup>[2616]</sup>

### Learning Center topics

- (Alt+F1) | [Enterprise Architect | Execution Analysis | Run Unit Tests](#)

## 16.5.3 Record Test Results

Enterprise Architect is able to automatically record all results from tests through a testing script in Enterprise Architect.

### Topics

Topic	Detail	See also
<b>Process</b>	<p>In order to use this feature, you must reverse engineer the test Class into the package containing your test script.</p> <p>Once your model contains your test Class, on the next run of the test script Enterprise Architect adds Test Cases to the Class for each test method found; on this and all subsequent test runs all Test Cases are updated with the current run time and whether they passed or failed, as shown below:</p>	<a href="#">Set Up Unit Testing</a> <sup>[2573]</sup> <a href="#">Importing Source Code</a> <sup>[2136]</sup> <a href="#">Run Unit Tests</a> <sup>[2575]</sup>

Topic	Detail	See also
	<div data-bbox="493 327 743 667"><div><b>Calculator Test</b></div><div>+ testAdd() : void + testDivide() : void + testMultiply() : void + testSubtract() : void</div><div><b>test scripts</b> Unit : (Pass) testAdd Unit : (Pass) testDivide Unit : (Pass) testMultiply Unit : (Fail) testSubtract</div></div> <p>The error description for each failed test is added to any existing results for that Test Case, along with the current date and time.</p> <p>Over time this provides a log of all test runs where each Test Case has failed, which can then be included in generated documentation, resembling the following:</p> <div data-bbox="517 936 1038 1122"><div>Failed at 05- Jul - 2006 1: 02: 08 PM expected: &lt;0&gt; but was: &lt;1&gt;</div><div>Failed at 28- Jun- 2006 8: 45: 36 AM expected: &lt;0&gt; but was: &lt;2&gt;</div></div>	



## 16.6 Testpoint Management

Testpoints define constraints that are applied to Classes and Operations, ensuring that a system is behaving as expected during runtime. Testpoints are evaluated during a debugging session and the result of each evaluation is recorded as either a Pass or Fail. Test results can be viewed from the Testpoints Window in real-time as the program is executed. After a test run has completed, the recorded results can be saved for later reference.

Tests	Id	Constraints	Status	Evals	Passes	Fails	Parent Collections:
CTrain				1945	1944	1	CityLoop
CTrain				1945	1944	1	
Invariant				1927	1926	1	
Passengers > 0	3		✗	1927	1926	1	
DWORD Disembark(int)			✓	6	6		
Pre-Condition				6	6		
PeopleOFF > 0	1			6	6		
DWORD Embark(int)			✓	6	6		
Pre-Condition				6	6		
PeopleON > 0	2			6	6		
DWORD OnArrival(CStation*)			✓	6	6		
Post-Condition				6	6		
ScheduleTime > 0	4			6	6		

### Topics

Topic	Detail	See also
<b>Overview</b>	<p>Testpoint design and management is performed using the <b>Test Domain diagram and Toolbox</b>. Testpoints are defined on Classes and Operations, which can be refined into Test Cuts, then grouped into Test Sets and Test Suites.</p> <p>Testpoints define <b>Constraints</b>. These can be Invariant constraints on Classes, or Pre-condition, Post-Condition and Line-Condition constraints on Operations.</p> <p>Constraints are defined and executed using the <b>Testpoint Window</b>. The Testpoint Window is context-sensitive based upon the currently selected item in the Project Browser or Diagram View.</p> <p>Testpoints are defined within the model and do not make any changes to your application source code. This makes it possible to quickly edit a testpoint constraint at any time and restart the test run without needing to re-build the application.</p>	<a href="#">The Testpoints Window</a> <sup>[2578]</sup> <a href="#">Testpoint Editor</a> <sup>[2584]</sup>
<b>Test Domain Diagram and Toolbox</b>	<p>The Test Domain diagram and toolbox allow definition of Test Cut, Test Set and Test Suite objects to aid in the logical composition of tests.</p> <p>Testpoints are always defined on Classes and Operations, but can be filtered and aggregated into logical Test Cuts and Test Sets.</p>	<a href="#">Test Domain Toolbox</a> <sup>[819]</sup> <a href="#">Test Cut</a> <sup>[2588]</sup> <a href="#">Test Set</a> <sup>[2588]</sup> <a href="#">Test Suite</a> <sup>[2589]</sup>

Topic	Detail	See also
	A Testpoint run can be performed for on an individual Class, a single Test Set, or an entire Test Suite.	<a href="#">Combine Testpoints</a> <sup>[2586]</sup>
<b>Constraints</b>	<p>The Testpoint facility is based on a programming-by-contract model. When executing a test run, the Execution Analyzer evaluates constraints defined for Classes and Operations:</p> <ul style="list-style-type: none"> <li>• A Class Invariant is evaluated by the Analyzer whenever any method called on an object of this Class type is completed. An invariant defines the rules or constraints of a Class; it is expected that no objects of the Class nor its methods can break this constraint, thus preserving the state of the object.</li> <li>• Pre-conditions are evaluated immediately before an operation is called. Post-conditions are evaluated (at the same time as a Class invariant) when the method is completed. Line-conditions are evaluated only when a specific line of source code is reached during execution.</li> </ul> <p>Testpoint constraints are composed using member and local variables in logical expressions which return either a true or false result. Any Local and Member variables referenced by a constraint must be in scope at the time the expression is evaluated.</p>	<a href="#">Constraint Composition</a> <sup>[2581]</sup>
<b>Trace Statements</b>	<p>When defining a Testpoint, an optional <b>Trace Statement</b> can also be specified. A Trace Statement is a message that is output during execution of a debug session. The Trace Statement <b>Level</b> can be configured to either output every time the constraint is evaluated, or only when the constraint condition has failed.</p> <p>Output from a Testpoint trace statement can either be directed to the Testpoints tab of the <b>Output Window</b>, or to an external file, as configured in the <b>Analyzer Script</b> for the parent package.</p>	<a href="#">Specifying a Trace Statement</a> <sup>[2228]</sup> <a href="#">The System Output Window</a> <sup>[169]</sup> <a href="#">Managing Analyzer Scripts</a> <sup>[2175]</sup>

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Execution Analysis | Testpoints | Introducing Testpoints**






### 16.6.1 The Testpoints Window

The Testpoints Window is used to configure, view and execute test constraints on the currently selected Testpoint object. This window is context-sensitive, responding to the item currently selected within either the Project Browser or Diagram View.

Configured tests can be executed using the Testpoints Window Toolbar, and test results can be saved for later review.

**Access** **Analyzer | Testpoint Manager**

#### Testpoint Window Columns

Column	Usage	See also
<b>Tests</b>	<p>Displays the name of the selected Testpoint object and the hierarchy of objects beneath it.</p> <p>The selected object can be a:</p> <ul style="list-style-type: none"> <li>• Class</li> <li>• Operation</li> <li>• Test Cut</li> <li>• Test Set or</li> <li>• Test Suite</li> </ul>	<a href="#">Test Cut</a> <sup>[2588]</sup> <a href="#">Test Set</a> <sup>[2588]</sup> <a href="#">Test Suite</a> <sup>[2589]</sup>
<b>Id</b>	<p>For an <b>Operation</b>, this column shows a Testpoint marker icon (  ) if the Analyzer has been able to successfully bind this operation with a defined Testpoint. If Testpoints are defined for an operation but no icon appears in this column during a test run, this may indicate a failure to bind.</p> <p>For a <b>Testpoint</b>, this column shows a generated id number. This id number is used in trace output to indicate which constraint is being referenced.</p>	
<b>Constraints</b>	<p>A pencil icon (  ) in this column indicates that one or more constraints are defined for this Class or Operation.</p>	<a href="#">Testpoint Editor</a> <sup>[2584]</sup>
<b>Status</b>	<p>During a test run, indicates the following possible statuses:</p> <ul style="list-style-type: none"> <li>• (  ) <b>Failed</b> - Constraint has evaluated as false one or more times.</li> <li>• (  ) <b>Invalid Statement</b> - Constraint failed to parse due to invalid syntax.</li> <li>• (  ) <b>Variable not found</b> - A referenced variable name was not found at the location where the constraint was evaluated.</li> </ul> <p>No icon is shown if a constraint has <b>Passed</b>.</p>	
<b>Evals</b>	<p>During a test run, indicates the number of times the Execution Analyzer has evaluated this constraint.</p>	
<b>Passes</b>	<p>During a test run, indicates the number of times the test passed.</p>	
<b>Fails</b>	<p>During a test run, indicates the number of times the test failed.</p>	
<b>Last Run By</b>	<p>Displays the username of the last person to run this test.</p>	

Column	Usage	See also
<b>Last Run Date</b>	Displays the date and time this test was last evaluated.	
<b>Last Run Result</b>	Displays the result of the last test run.	
<b>Parent Collections Pane</b>	<p>Lists any parent collections that include the selected object as part of their design.</p> <p>Double-click this collection to make it the selected object in the left pane.</p> <p>The Parent Collections pane can be hidden by clicking the Show / Hide Parent Collections pane button on the <b>Testpoints Window Toolbar</b>.</p>	

#### Learn more

- [The Testpoints Window Toolbar](#)<sup>[2580]</sup>

#### Learning Center topics

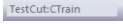

- (Alt+F1) | **Enterprise Architect | Execution Analysis | Testpoints |**
  - **Add Class**
  - **Add Test Cut**
  - **Add Test Set**
  - **Add Test Suite**








### 16.6.1.1 The Testpoints Window Toolbar

The Testpoints Window Toolbar provides options to execute configured tests on the currently selected Testpoint object, stop a test run currently in progress, filter the displayed items, and save the results of a completed test run.

**Access**    **Analyzer | Testpoint Manager**

#### Testpoints toolbar options

Toolbar Button	Action	See also
	Field showing the name of the currently selected Testpoint object.	
	Execute the test run.	

	Stop the test run currently in progress.	
	Toggle between showing all items and showing only those items that have constraints defined.	
	<p>Toggle between showing all items and showing only operations that have been marked for inclusion in this Test Cut; this button is only enabled when a Test Cut object is selected.</p> <p>When a Test Cut is selected, each of the operations of its associated Class are displayed with a checkbox; you use this checkbox to mark the operations that apply to this Test Cut.</p>	<a href="#">Test Cut</a> <sup>[2588]</sup>
	<p>Click on the drop arrow next to this icon to display the <b>Test Run Options</b> menu, providing the following options:</p> <ul style="list-style-type: none"> <li>• <b>Prefix Trace output With Function Call</b> - Prefix all trace output lines with the executing function name</li> <li>• <b>Enable Standard Breakpoints during Testing</b> - When not checked, test run ignores any breakpoints in current breakpoint set, and attempts to set breakpoints during run are ignored</li> <li>• <b>View Trace output</b> - Display the Testpoints tab of the Output window</li> </ul>	<a href="#">Breakpoint and Marker Management</a> <sup>[2224]</sup> <a href="#">The Output Window</a> <sup>[169]</sup>
	<p>Click on this icon after completion of a test run to save the results to Test item on the current object. Saved tests can be viewed using the <b>Testing Workspace</b>.</p> <p>A prompt displays to select the Test Class - Unit, Integration, System, Acceptance or Scenario. Select the appropriate Test Class and click on the <b>OK</b> button.</p>	<a href="#">Working on Test Records</a> <sup>[2605]</sup>
	Display the Testpoint Management help topic.	
	Show or Hide the <b>Parent Collections</b> pane	<a href="#">The Testpoints Window</a> <sup>[2578]</sup>

**Learn more**

- [Testpoint Management](#) <sup>[2577]</sup>

**16.6.1.2 Constraint Composition**

A Constraint is typically composed using local and member variables in expressions, separated by operators to define one or more specific criteria that must be met. A constraint must evaluate as **true** to be considered as **Passed**. If a constraint evaluates as **false**, it is considered as **Failed**.

Any variables referenced within the constraint must be in scope at the position where the Testpoint or Breakpoint is evaluated.

**General/Arithmetic Operators**

Operator	Description	Example
+	Add	$a + b > 0$
-	Subtract	$a - b > 0$
/	Divide	$a / b == 2$
*	Multiply	$a * b == c$
%	Modulus	$a \% 2 == 1$
( )	Parentheses - Used to define <i>precedence</i> in complex expressions.	$((a / b) * c) \leq 100$
[ ]	Square Brackets - Used for accessing Arrays.	<code>Names[0].Surname == "Smith"</code>
.	Dot operator - Used to access member variables of a class.	<code>Station.Name == "Flinders"</code>
->	Alternative notation for the Dot operator.	<code>Station-&gt;Name == "Flinders"</code>

**Comparison Operators**

Operator	Description	Example
=	Equal To	$a = b$
==	Equal To	$a == b$
!=	Not Equal To	$a != b$
<>	Not Equal To	$a <> b$
>	Greater Than	$a > b$

Operator	Description	Example
<code>&gt;=</code>	Greater Than or Equal To	<code>a &gt;= b</code>
<code>&lt;</code>	Less Than	<code>a &lt; b</code>
<code>&lt;=</code>	Less Than or Equal To	<code>a &lt;= b</code>

### Logical Operators

Operator	Description	Example
<b>AND</b>	Logical AND	<code>(a &gt;= 1) AND (a &lt;= 10)</code>
<b>OR</b>	Logical OR	<code>(a == 1) OR (b == 1)</code>

### Bitwise Operators

Operator	Description	Example
<b>&amp;</b>	Bitwise AND	<code>(1 &amp; 1) = 1</code> <code>(1 &amp; 0) = 0</code>
<b> </b>	Bitwise OR	<code>(1   1) = 1</code> <code>(1   0) = 1</code>
<b>^</b>	Bitwise XOR (exclusive OR)	<code>(1 ^ 1) = 0</code> <code>(1 ^ 0) = 1</code>

### Additional Examples

Example	Description
<code>(( m_nValue &amp; 0xFFFF0000) == 0)</code>	Use a Bitwise AND operator (&) with a hexadecimal value as the right operand to test that no bits are set in high order bytes of the

Example	Description
	variable.
(( m_nValue & 0x0000FFFF) == 0)	Use a Bitwise AND operator (&) with a hexadecimal value as the right operand to test that no bits are set in low order bytes of the variable.
m_value[0][1] = 2	Accessing a multi-dimensional array
a AND (b OR c)	Combining AND and OR operators, using parentheses to ensure precedence. In this example, variable 'a' must be true, and either 'b' or 'c' must be true.

#### Notes

- String comparisons are *case-sensitive*

#### Learn more

- [Testpoint Editor](#)  <sup>2584</sup>
- [Breakpoint Properties](#)  <sup>2226</sup>

#### Learning Center topics

- (Alt+F1) | [Enterprise Architect](#) | [Execution Analysis](#) | [Testpoints](#) | [About Constraints](#)

## 16.6.2 Testpoint Editor

The Testpoint Editor is used to compose constraints for Classes and Operations. The types of constraints permitted are dependent on the selected object. For Classes, the type will always be **Invariant**. For operations, the type can be either **Pre-Condition**, **Post-Condition** or **Line-Condition**.

**Invariants** are evaluated by the Analyzer when any method called on an object of the selected Class type completes. **Pre-conditions** are evaluated at the beginning of each call to the specified operation. **Post-conditions** are evaluated upon completion of each call to the specified operation. **Line-conditions** are evaluated each time the specified line of code is executed.

**Access** [Analyzer](#) | [Testpoint Manager](#): Double-click on a Class or Operation in the Testpoints window

#### Constraint Group fields

Field	Usage	See also
<b>Type</b>	The type of constraint for the selected Class or Operation:	



Field	Usage	See also
	<ul style="list-style-type: none"> <li>• <b>Invariant</b> - Evaluated after any method called on the specified Class has completed</li> <li>• <b>Pre-Condition</b> - Evaluated at the beginning of each call to a specific Operation</li> <li>• <b>Post-Condition</b> - Evaluated after completion of each call to a specific Operation</li> <li>• <b>Line-Condition</b> - Evaluated upon execution of a specific line of code within an Operation</li> </ul>	
<b>Offset</b>	<p>Line number within the specified operation upon which to evaluate the constraint. Only applicable for <b>Line-Condition</b> constraint types.</p> <p>An offset value is automatically set if the Testpoint was created using the Code Editor Context Menu.</p>	<a href="#">Code Editor Context Menu</a> <sup>[2152]</sup>
<b>Condition</b>	The constraint to be evaluated when this Testpoint is triggered. A status of pass or fail will be recorded depending upon whether this constraint condition evaluates as true or false.	<a href="#">Constraint Composition</a> <sup>[2581]</sup>
<b>Action on Fail</b>	<p>Click on the drop-down arrow and select from the three options:</p> <ul style="list-style-type: none"> <li>• <b>Continue</b> - ignore failure of this constraint and continue execution</li> <li>• <b>Break execution</b> - halt execution and display the Stack trace</li> <li>• <b>Disable on fail</b> - do not execute the constraint again after failing once</li> </ul>	
<b>Evaluate When</b>	(Optional) An additional constraint which must be met before the main Testpoint Condition is evaluated, providing greater control over test coverage.	<a href="#">Constraint Composition</a> <sup>[2581]</sup>

#### Trace Group fields

Field	Usage	See also
<b>Level</b>	<p>Specifies when the trace statement (if defined) will be output. Available options are:</p> <ul style="list-style-type: none"> <li>• <b>Fail Only</b> - Output trace statement only when this Testpoint condition fails</li> <li>• <b>Always</b> - Output trace statement every time this Testpoint is evaluated</li> </ul>	
<b>Statement</b>	<p>(Optional) A message to be output when this Testpoint is evaluated.</p> <p>Variables currently in scope can be included in a trace statement output by prefixing the variable name with a <b>\$</b> token for string variables, or a <b>@</b> token</p>	<a href="#">Specifying a Trace Statement</a> <sup>[2228]</sup>

Field	Usage	See also
	<p>for primitive types such as int or long.</p> <p>Output from a Trace Statement may either be directed to the Testpoints tab of the <b>Output Window</b>, or to an external file, as configured by the <b>Analyzer Script</b> for the parent package.</p>	

#### Learn more

- [The Testpoints Window](#)<sup>[2578]</sup>

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Execution Analysis | Testpoints |**
  - **Edit Invariant**
  - **Edit Pre Condition**
  - **Edit Post Condition**

### 16.6.3 Combine Testpoints

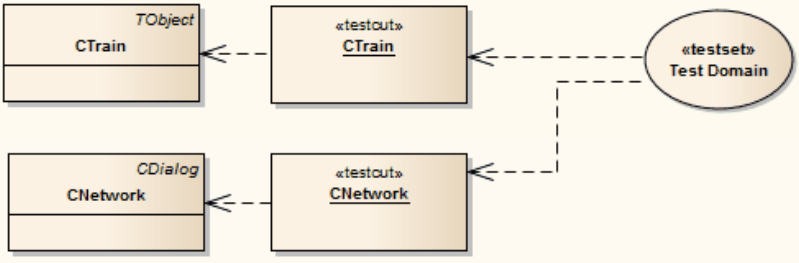
Testpoints are applied to a Class. Where functionality requires tests across multiple Classes, you can group the tests together to allow a single test run across the group of Classes.

You group the tests on a Test Domain diagram, either:

- Manually using the Test Domain Toolbox pages and Project Browser, or
- Automatically from the results of using the debugger and Record & Analyze window

#### Topics

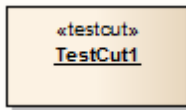
Topic	Detail	See also
<b>Create Test Domain diagram manually</b>	<p>Create a new Test Domain diagram under the required package, and drag a Test Cut element from the Test Domain Toolbox page onto the diagram; give the element an appropriate name, such as the name of the Class it acts on.</p> <p>Drag the Class containing the tests from the Project Browser onto the diagram, and select the 'paste as simple link' option on the Paste Element dialog; create a Dependency connector between the Test Cut element and the Class.</p> <p>(When you select the Test Cut element on the diagram when the Testpoints window is open, the operations of the Class are listed in the window, each followed by a checkbox; you then select the checkbox against each Class operation to include in the Test Cut).</p> <p>Drag and connect further Test Cut and Class element pairs on the diagram as appropriate.</p> <p>Drag a Test Set element from the Toolbox onto the diagram, and create a</p>	<p><a href="#">Test Domain Toolbox</a><sup>[819]</sup></p> <p><a href="#">Test Cut</a><sup>[2588]</sup></p> <p><a href="#">Test Set</a><sup>[2588]</sup></p>

Topic	Detail	See also
	<p>Dependency relationship between this element and each of the Test Cut elements; the resulting diagram should resemble the following:</p>  <p>If you want to further combine tests, set up additional Test Set hierarchies and then drag a Test Suite element onto the diagram from the Toolbox; create a Dependency relationship between this element and each of the Test Set elements.</p> <p>When you open the Testpoints window, you can select any element on the diagram to display that element and its hierarchy in the window; you can then execute the test run on your selection.</p> <p>If the selection checkboxes do not display, select the Testcut element on the diagram again.</p>	<p><a href="#">Test Suite</a> <sup>[2589]</sup></p> <p><a href="#">The Testpoints Window</a> <sup>[2578]</sup></p>
<b>Generate Test Domain diagram from Sequence Record</b>	<p>To generate a Testpoint Domain diagram automatically:</p> <ul style="list-style-type: none"> <li>• Build and debug your code successfully</li> <li>• Work through the process of recording the stack trace to generate a Sequence diagram</li> <li>• When you have completed recording (whether or not you display the Sequence diagram), click on the <b>Testpoints</b> icon in the Record &amp; Analyze window toolbar</li> <li>• Open the Testpoints window, select any element on the diagram to display that element and its hierarchy in the window, and execute the test run on your selection</li> </ul> <p>You cannot automatically generate a <b>Test Suite</b> element from the Sequence record, only Test Sets, Test Cuts and Classes.</p>	<p><a href="#">Recording Sequence Diagrams</a> <sup>[2531]</sup></p> <p><a href="#">Recorder Toolbar</a> <sup>[2544]</sup></p> <p><a href="#">The Testpoints Window</a> <sup>[2578]</sup></p>

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Execution Analysis | Testpoints | Add Test Cut**
- (Alt+F1) | **Enterprise Architect | Execution Analysis | Testpoints | Add Test Set**
- (Alt+F1) | **Enterprise Architect | Execution Analysis | Testpoints | Add Test Suite**

### 16.6.3.1 Test Cut



#### Description

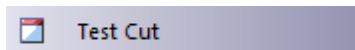
A Test Cut element is a stereotyped Object element, used internal to Enterprise Architect for defining test sets using the Testpoint code testing facilities.

A task, such as 'Print', might involve operations on different Classes. In order to create a 'Print' test, you would want to include only the 'Print' operations of these Classes and exclude any other operations.

A Test Cut enables you to capture only the operations that represent the behavior (in this case, 'Print') defined for a single Class. You might then place the Test Cut from each of several Classes into a single task as a Test Set.

When you drag a Test Cut element onto a Test Domain diagram, you create a Dependency relationship with the required Class element. As a result, when you select the *Test Cut* element on the Testpoints Window, the operations of the *Class* are listed in the window, each with a checkbox. You then select the checkbox against each Class operation to include in the Test Cut.

#### Toolbox icon



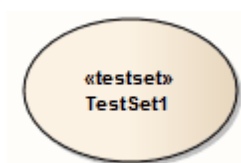
#### Learn more

- [Test Set](#)<sup>[2588]</sup>
- [The Testpoints Window](#)<sup>[2578]</sup>
- [Combine Testpoints](#)<sup>[2586]</sup>

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Execution Analysis | Testpoints | Add Test Cut**

### 16.6.3.2 Test Set

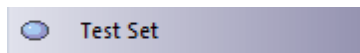


#### Description

A *Test Set* element is a stereotyped Use Case element used to aggregate one or more groups of methods (Test Cuts), which perhaps span multiple Classes, into a single task. Test Sets can also be aggregated into Test Suites.

You link the Test Cut elements to the Test Set using Dependency connectors.

#### Toolbox icon



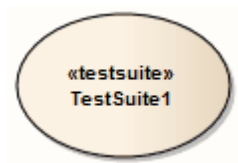
#### Learn more

- [Test Cut](#) [2588]
- [Test Suite](#) [2589]
- [Combine Testpoints](#) [2586]

#### Learning Center topics

- (Alt+F1) | [Enterprise Architect](#) | [Execution Analysis](#) | [Testpoints](#) | [Add Test Set](#)

### 16.6.3.3 Test Suite

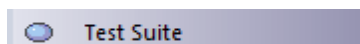


#### Description

A *Test Suite* element is a stereotyped Use Case element, used to aggregate one or more groups of tasks (Test Sets).

You link the Test Set elements to the Test Suite using Dependency connectors.

#### Toolbox icon



#### Learn more

- [Test Set](#) [2588]
- [Combine Testpoints](#) [2586]

Learning Center topics

- (Alt+F1) | **Enterprise Architect | Execution Analysis | Testpoints | Add Test Suite**

## 16.7 Object Run State Diagrams

The run-time of an object instance can be defined using the local variables window of the Enterprise Architect debugger. During a debug session variables can be dragged from this window on to an Object diagram. The run-time of the element will then be composed using the attributes of the variable. Dragging members of the same variable will create an association. Using this method the complex run-of an element time can be composed quite quickly.

Requires that an [Analyzer Script](#)<sup>[2175]</sup> has been configured for debugging.

**Part**

---





## 17 Testing



It is important in any project to perform quality control of both the process and the output of the project; Enterprise Architect provides several facilities for testing and validating your model structure and content, including:

- **Model Validation Checks** - check UML elements, diagrams or packages against known UML rules (identified in configuring validation) and constraints defined within the model, using the Object Constraint Language (OCL)
- **Testing** - create and manage test scripts for model elements, covering unit, integration, scenario, system and acceptance tests
- **Testpoint Management** - pass or fail application tasks, viewing test results in real time as the program executes and results are saved
- Integration with the **unit testing** tools **JUnit** and **NUnit**

### Learn more

- [Model Validation](#) <sup>[2594]</sup>
- [Testing](#) <sup>[2604]</sup>
- [TestPoint Management](#) <sup>[2577]</sup>
- [Unit Testing](#) <sup>[2573]</sup>

## 17.1 Model Validation



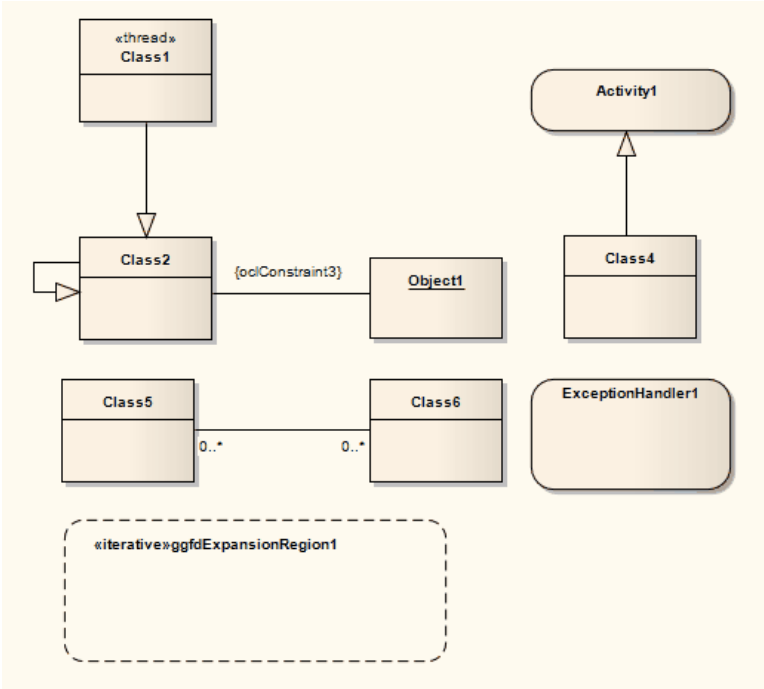
You use Model Validation to check UML models against known **UML rules** (which you identify when **configuring validation**) as well as any constraints defined within the model using the Object Constraint Language (OCL). You can run Model Validation against a single UML element, a diagram or an entire Package.

You can add to the system-provided validation rules by using an Add-In or an MDG Technology that contain their own rules.

**Access** **Project | Model Validation | Validate Selected (Ctrl+Alt+V)**

### Validation Concepts

Concept	Detail	See also
<b>Validating Objects</b>	Validating a UML: <ul style="list-style-type: none"> <li>• <b>Element</b> validates the element and its children, its features (attributes and operations) and its relationships (connectors)</li> <li>• <b>Diagram</b> validates the diagram itself (for correctness) as well as any elements and connectors within the diagram</li> <li>• <b>Package</b> validates the Package and all sub-Packages, elements, connectors and diagrams within it</li> </ul>	
<b>Example - Model Violation</b>	The following UML diagram contains several basic violations of the UML rules:	

Concept	Detail	See also
	 <p>If you run Model Validation on this diagram, the following violations are identified in the System Output window:</p> <ul style="list-style-type: none"> <li>• A UML ExpansionRegion (<i>ExpansionRegion1</i>) is missing its child input ExpansionNode</li> <li>• An invalid self-generalization exists on <i>Class2</i> (UML elements cannot be self-generalized)</li> <li>• An OCL violation exists for the anonymous Association (between <i>Class2</i> and <i>Object1</i>)</li> <li>• A UML ExceptionHandler (<i>ExceptionHandler1</i>) is missing its child input ObjectNode</li> </ul>	

### Learn more

- [Run Validation](#) <sup>[2596]</sup>
- [Configure Model Validation](#) <sup>[2596]</sup>
- [Rules Reference](#) <sup>[2597]</sup>
- [Model Validation Broadcasts](#) <sup>[3047]</sup> (Add-Ins)
- [Define Validation Configuration](#) <sup>[1576]</sup> (MDG Technologies)

### 17.1.1 Configure Model Validation

Before you perform a model validation, you enable or disable the rules that are applied by the model validator. You can define additional rules from any additional Add-Ins that might be installed besides Enterprise Architect.

**Access**    **Project | Model Validation | Configure**

#### Configure Validation

On the Model Validation Configuration dialog, click on the checkbox against each Validation Rule to apply in performing a model validation. Click on the **OK** button to set the validation rules selected.

#### Learn more

- [Model Validation](#)<sup>[2594]</sup>
- [Run Validation](#)<sup>[2596]</sup>
- [Rules Reference](#)<sup>[2597]</sup>
- [Model Validation Broadcasts](#)<sup>[3047]</sup> (in Add-Ins)
- [Model Validation Example](#)<sup>[3058]</sup> (Add-In)

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Testing | Model Validation | How to Configure**

### 17.1.2 Run Validation

After you have configured the validation you want to perform and set the validation rules to be applied, you can execute the validation on objects you have selected from either the Project Browser or a diagram. If you do not want to continue a validation you have started, you can cancel it.

**Access**    **select object | Project | Model Validation | Validate Selected**

#### Validation

While the validation executes it displays a progress window. When validation is complete, the results are displayed in the System Output window. You can highlight and select on a diagram the element that an error message refers to, by double-clicking on the error line in the System Output window.

#### Cancel Validation

At any time during a validation, either:

- Click on the **Cancel Validation** button on the validation progress window, or
- Select the **Project | Model Validation | Cancel Validation** menu option

#### Notes

- If the System Output window does not automatically display, select the **View | System Output** menu option

#### Learn more

- [The System Output Window](#)<sup>[169]</sup>
- [Rules Reference](#)<sup>[2597]</sup>

#### Learning Center topics


- (Alt+F1) | **Enterprise Architect | Testing | Model Validation | How to Run**

### 17.1.3 Rules Reference

Model Validation works by applying a set of validation rules to the selected object. These rules are arranged in groups.

#### Rule Groups

Group	Description	See also
<b>Well-Formedness</b>	<p>The validator applies these rules to check whether or not an element, relationship, feature or diagram is well-formed; for example, whether the object is a valid UML item or whether a diagram contains valid elements within it.</p> <p>As explained in <i>Error Message Format</i>, below, these rules generate messages with the codes:</p> <ul style="list-style-type: none"> <li>• MVR01.... (<b>Element: Well Formedness</b> checkbox selected)</li> <li>• MVR05.... (<b>Relationship: Well Formedness</b> checkbox selected)</li> <li>• MVR08.... (<b>Feature: Well Formedness</b> checkbox selected)</li> <li>• MVR0B.... (<b>Diagram: Well Formedness</b> checkbox selected)</li> </ul>	<a href="#">Well-Formedness</a> <sup>[2599]</sup>
<b>Element Composition</b>	<p>The validator applies these rules to check whether or not a UML element contains valid children, whether it contains the right number of valid children, and whether or not the element is missing any required children.</p> <p>These rules generate messages with the code:</p> <ul style="list-style-type: none"> <li>• MVR02.... (<b>Element: Composition</b> checkbox selected)</li> </ul>	<a href="#">Element: Composition</a> <sup>[2600]</sup>
<b>Property Validity</b>	<p>The validator applies these rules to check whether or not the element, relationship or feature has the correct UML properties defined, and whether the properties contain incorrect or conflicting values.</p> <p>These rules generate messages with the codes:</p>	<a href="#">Property Validity</a> <sup>[2600]</sup>

Group	Description	See also
	<ul style="list-style-type: none"> <li>MVR03.... (<b>Element: Property Validity</b> checkbox selected)</li> <li>MVR06.... (<b>Relationship: Property Validity</b> checkbox selected)</li> <li>MVR09.... (<b>Feature: Property Validity</b> checkbox selected)</li> </ul>	
<b>Custom Properties</b>	<p>The validator applies these rules to check an element, relationship or feature against any defined constraints in OCL.</p> <p>These rules generate messages with the codes:</p> <ul style="list-style-type: none"> <li>MVR04.... (<b>Element: OCL Conformance</b> checkbox selected)</li> <li>MVR07.... (<b>Relationship: OCL Conformance</b> checkbox selected)</li> <li>MVR0A.... (<b>Feature: OCL Conformance</b> checkbox selected)</li> </ul>	<a href="#">OCL Conformance</a> 

### Error Message Format

When you perform a validation, any violations of the rules are listed on the System Output window. Each violation has a violation ID of the format:

*MVRxxnnnn*

- MVR* stands for Model Validation Rule
- xx* is a hexadecimal number corresponding to the position of the validation rule in the Model Validation Configuration dialog, thus indicating which rule is applied and violated
- nnnn* is the number of the violation message

For example:

- Messages with the ID *MVR01nnnn* indicate that the **Element: Well-Formedness** checkbox is selected and a violation of that rule has been detected
- Messages with the ID *MVR0Annnn* indicate that the **Feature: OCL Conformance** checkbox (10th in order on the dialog, or *Ath* in hexadecimal) is selected and a violation of that rule has been detected

### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Testing | Model Validation | Viewing Results**

### 17.1.3.1 Well-Formedness

**Well-formedness** rules are applied to check whether or not an element, relationship, feature or diagram is **well-formed**. For example, whether the target item is a valid UML item, or whether a diagram contains valid elements within it.

#### Reported violations

Violation ID	Description	Error
<b>MVR01000</b> <b>1</b>	«Element» is not a valid UML Element	The element is not a recognized UML 2.4.1 element.
<b>MVR05000</b> <b>1</b>	«Relationship» is not a valid UML Relationship	The relationship is not a recognized UML 2.4.1 relationship.
<b>MVR05000</b> <b>2</b>	«Relationship» is not legal for «Start Element» --> «End Element»	The relationship between the given start and end elements is not valid for those elements.
<b>MVR05000</b> <b>3</b>	«Parent Element»:isLeaf=true and cannot be generalized by «Child Element»	The Generalization relationship cannot exist between parent and child elements because the parent element is defined as a leaf element.
<b>MVR05000</b> <b>4</b>	«Child Element»:isRoot=true and cannot generalize «Parent Element»	The Generalization relationship cannot exist between parent and child elements because the child element is defined as a root element.
<b>MVR05000</b> <b>5</b>	«Element» cannot generalize self	The element cannot be self-generalized.
<b>MVR0B00</b> <b>01</b>	Statechart violation: «extended information»	The State diagram contains a UML violation; «extended information» provides more details on the specific violation.
<b>MVR0C00</b> <b>01</b>	Sequence Violation: «extended information»	The Sequence diagram contains a violation, «extended information» provides more details about the specific violation.  (This rule validates operations being called by Messages - as selected from the drop down combo box of the Messages Properties dialog - and will report a violation if the operation does not belong to the Lifeline.)
<b>MVR7F000</b> <b>1</b>	«Requirement» Not realized	A Requirement element must be at the target end of a Realization, Association (including Aggregation and Composition) or Generalization connector.

Violation ID	Description	Error

#### Learn more

- [Rules Reference](#) <sup>[2597]</sup>
- [Generalization](#) <sup>[1409]</sup>
- [Association](#) <sup>[1393]</sup>
- [Realization](#) <sup>[1440]</sup>

### 17.1.3.2 Element Composition

**Element Composition** rules are applied to check whether or not a UML **element** contains valid children, whether it contains the right number of valid children, and whether or not the element is missing any required children.

#### Reported violations

Error ID	Description	Information
<b>MVR0200 01</b>	«Element» is missing required child element «Child Element»	The element is missing a child element of type <b>Child Element</b> .
<b>MVR0200 02</b>	Invalid UML Package child	The element cannot be a direct child of the Package and must be a child of another element (for example, Ports must be children of other elements, and not direct UML Package members).
<b>MVR0200 03</b>	Invalid child «Child Element name» («Child Element Type»)	The child element is invalid on the tested parent element.

#### Learn more

- [Rules Reference](#) <sup>[2597]</sup>

### 17.1.3.3 Property Validity

**Property Validity** rules are applied to check whether or not an element, relationship or feature has the correct UML **properties** defined for it and whether they contain incorrect or conflicting values.

#### Reported violations

Error ID	Description	Information
<b>MVR03000 1</b>	«Element»:«Property» property is undefined	The element property contains no value.



Error ID	Description	Information
<b>MVR03000 2</b>	«Element»:«Property» property has invalid value: "«Value»"	The element property contains an invalid value.
<b>MVR03000 3</b>	«Element»:isLeaf=true and cannot be abstract	The element's <i>isLeaf</i> and <i>isAbstract</i> properties are both set to <b>true</b> , which is invalid.
<b>MVR06000 1</b>	«Relationship»:«Property» property is undefined	The relationship property contains no value.
<b>MVR06000 2</b>	«Relationship»:«Property» property has invalid value: "«Value»"	The relationship property contains an invalid value.
<b>MVR09000 1</b>	Attribute/AssociationEnd mismatch, «Attribute»: «Mismatch description», ...	The given attribute has an <i>associationEnd</i> of the same name but they differ in the listed details.

#### Learn more

- [Rules Reference](#)<sup>[2597]</sup>
- [Properties Dialog](#)<sup>[956]</sup> (Element)
- [Connector Properties](#)<sup>[1126]</sup>

### 17.1.3.4 OCL Conformance

**OCL Conformance** rules are applied to validate an element, relationship or attribute against any defined constraints in the **Object Constraint Language** (OCL). OCL is used to describe expressions on UML models, and to express constraints free of side-effects. You can add OCL constraints to any element, relationship or attribute in Enterprise Architect.

#### Reported violations

Error ID	Description	Information
<b>MVR040001</b>	OCL violation: «violated OCL»	The <b>element</b> violates the specified OCL constraint.
<b>MVR070001</b>	OCL violation: «violated OCL»	The <b>relationship</b> violates the specified OCL constraint.
<b>MVR0A0001</b>	OCL violation: «violated OCL»	The <b>attribute</b> violates the specified OCL constraint .

### Define OCL Constraints

You can add an OCL constraint to a model object by setting **OCL** as the Constraint type in the object Properties dialog.

For	Detail	See also
<b>An Element</b>	<p>Select <b>Element   Properties &gt; Constraints</b>; in the <b>Type</b> field click on the drop-down arrow and select <b>OCL</b>.</p> <p>To perform an OCL <b>Validation</b>:</p> <ol style="list-style-type: none"> <li>1. Select <b>Project   Model Validation   Configure</b> and select the <b>Element: (OCL) Conformance</b> checkbox</li> <li>2. Click on the element and select <b>Project   Model Validation   Validate Selected</b></li> </ol> <p>Any OCL violations are recorded in the Model Validation tab of the System Output window.</p>	<a href="#">Model Validation Configuration</a> <sup>[2596]</sup> <a href="#">Model Validation</a> <sup>[2594]</sup>
<b>A Relationship</b>	<p>On a diagram, right-click on the connector and select the <b>&lt;type&gt; Properties</b> context menu option; select the Constraints page, and in the <b>Type</b> field click on the drop-down arrow and select <b>OCL</b>.</p> <p>To perform an OCL <b>Validation</b>:</p> <ol style="list-style-type: none"> <li>1. Select <b>Project   Model Validation   Configure</b> and select the <b>Relationship: (OCL) Conformance</b> checkbox</li> <li>2. Click on the connector and select <b>Project   Model Validation   Validate Selected</b></li> </ol> <p>Any OCL violations are recorded in the Model Validation tab of the System Output window.</p>	
<b>An Attribute</b>	<p>Select <b>Element   Attributes &gt; Constraints</b>; in the <b>Type</b> field click on the drop-down arrow and select <b>OCL</b>.</p> <p>To perform an OCL <b>Validation</b>:</p> <ol style="list-style-type: none"> <li>1. Select <b>Project   Model Validation   Configure</b> and select the <b>Feature: (OCL) Conformance</b> checkbox</li> <li>2. Click on the element and select <b>Project   Model Validation   Validate Selected</b></li> </ol> <p>Any OCL violations are recorded in the Model Validation tab of the System Output window.</p>	

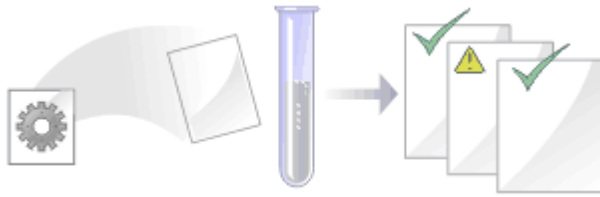
### Notes

- To have a valid OCL constraint, the syntax must be correctly formed; if the expression is not correct, a message displays stating that the OCL constraint is not valid

### Learn more

- [Rules Reference](#) 

## 17.2 Testing



Enterprise Architect is not only a UML Modeling environment, it is also a complete **Test Management environment**. Using Enterprise Architect you can create and manage **test scripts** for model elements, covering unit, integration, scenario, system and acceptance tests; these can include **test cases** generated from **xUnit testing** and **Testpoint Management**.

You can also **import** tests from other elements, **generate** them from scenarios, and generate **test documentation** and reports; you can indicate the presence of tests on an element by displaying test information on the element in a diagram.

It is simple to attach even complex tests to any model element. Keeping the model elements and the testing documentation in one integrated model significantly improves the communication between the test-team and the software developers and architects.

The system's detailed **search facilities** make it easy to find failing test cases, test cases not run and test cases that have been passed; using the testing and search capabilities, it is easy to navigate through the model and quickly locate problem spots, design flaws and other critical issues.

### Test Tasks

Tasks	Detail	See also
<b>Create Tests</b>	<p>You create tests in the <b>Testing Workspace</b>, and using the <b>Test Details</b> dialog.</p> <p>Typically, you create:</p> <ul style="list-style-type: none"> <li>• <b>Unit tests</b> for things that are being built, such as Classes and components</li> <li>• <b>Integration tests</b> to test how components work together</li> <li>• <b>System tests</b> to ensure the system meets business requirements</li> <li>• <b>Acceptance tests</b> to test user satisfaction, and</li> <li>• <b>Scenario tests</b> to test the end-to-end suitability and functionality of the application</li> </ul> <p>These test categories are otherwise referred to as test <b>Classes</b>; the type of Class is internally identified by the <i>C/ass</i> attribute, which has a value from <b>1</b> to <b>5</b> corresponding to the test types in the order listed above.</p>	<p><a href="#">Working on Test Records</a> <sup>[2605]</sup></p> <p><a href="#">Create Test Records</a> <sup>[2607]</sup></p> <p><a href="#">Test Class</a> <sup>[2908]</sup></p>
<b>Using Tests</b>	<p>Tasks that you might perform when working with tests include:</p> <ul style="list-style-type: none"> <li>• Import a scenario as a test</li> <li>• Move or copy tests between test types</li> <li>• Import a test from other elements</li> </ul>	<p><a href="#">Import a Scenario as a Test</a> <sup>[2611]</sup></p> <p><a href="#">Move or Copy Tests Between Test Types</a> <sup>[2610]</sup></p>

Tasks	Detail	See also
	<ul style="list-style-type: none"> <li>• Import a responsibility or constraint as a test</li> <li>• Create a maintenance item from a test</li> <li>• Generate a Test Details report</li> <li>• Show test script compartments</li> <li>• Create test documentation</li> </ul>	<a href="#">Import a Test from Other Elements</a> <sup>[2613]</sup> <a href="#">The Element Browser</a> <sup>[989]</sup> <a href="#">Import a Responsibility or Constraint as a Test</a> <sup>[2614]</sup> <a href="#">Create a Maintenance Item from a Test</a> <sup>[2615]</sup> <a href="#">Generate Test Details Report</a> <sup>[2736]</sup> <a href="#">Show Test Script Compartments</a> <sup>[2616]</sup> <a href="#">Create Test Documentation</a> <sup>[2617]</sup>

### Notes

- Most of the tasks identified above relate to a tests for a single element
- You can make a set of tests available to a number of elements by performing the above tasks on a **Test Case** element and then associating that Test Case with each of the other elements; the Test Case element also helps to make tests more visible in diagrams, the Project Browser, windows and searches

### Learn more

- [Test Case](#) <sup>[2010]</sup>
- [xUnit Testing](#) <sup>[2573]</sup>
- [Testpoint Management](#) <sup>[2577]</sup>

### Learning Center topics




- (Alt+F1) | **Enterprise Architect | Testing | Testing | Introduction to Testing**
- (Alt+F1) | **Enterprise Architect | Testing | Testing | Test Case Elements**


## 17.2.1 Working On Test Records

Creating and working on element Test records is quick and convenient, using the **Testing window** or **Workspace**. If the Testing window is open, when you select an element in a **diagram** or in the **Project Browser**, the tests for that element are immediately listed in the window ready for modification or addition. The window provides several facilities for creating and managing the test records.

Access    **Element | Testing**    (Alt+3)

### Facilities

Facility	Detail	See also
<b>Window Formats</b>	<p>The Testing window has two formats - <b>Item</b> mode and <b>List</b> mode.</p> <ul style="list-style-type: none"> <li>• <b>Item</b> mode provides a list on the left hand side from which you select a test record, the details of which then display in the fields in the rest of the window; you can edit these fields directly</li> <li>• <b>List</b> mode consists of a list of test records with summary data distributed across the columns; you cannot directly edit these fields</li> </ul> <p>To toggle between the modes, click on the  (<b>Show/Hide Properties</b>) button in the window toolbar.</p>	
<b>Adding New Items</b>	<p>To add new items, click on the <b>New</b> icon in the window toolbar. In:</p> <ul style="list-style-type: none"> <li>• Item mode, this clears the fields for new data</li> <li>• List mode, this displays the Test Details dialog, which you complete in the same way as the Testing window in Item mode</li> </ul>	<a href="#">Create Test Records</a>  <sup>2607</sup>
<b>Applying Automatic Naming/Numbering</b>	<p>On the Testing window in Item mode, or on the Test details dialog, you can apply an automatic naming/numbering convention that you have previously defined, to each new test record. To do this, simply click on the <b>Auto</b> button next to the <b>Test</b> field.</p> <p>If you already have some text in the <b>Test</b> field, it is over-written by the auto-counter text.</p>	<a href="#">Set Auto Naming and Auto Counters</a>  <sup>904</sup>
<b>Test Types</b>	<p>The five types of test you can create records for are each managed on a separate tab of the window:</p> <ul style="list-style-type: none"> <li>• Unit tests - to test Classes, Components and other elements as programmers build them</li> <li>• Integration tests - to test how the constructed components work together</li> <li>• System tests - to test that the system performs the right business functions correctly</li> <li>• Acceptance tests - to test the system against user requirements</li> <li>• Scenario tests - to test the application with real-world situations and scenarios; an end-to-end test of all functions</li> </ul> <p>The Testing window opens at the Unit test tab by default.</p> <p>The fields you complete for each type of test are identical.</p> <p>An asterisk on a tab indicates that the tab contains saved information; if there is no information for a type of test, or the information has not yet been <b>saved</b>, its tab has no asterisk.</p> <p>You toggle the display for each tab between Item mode and List mode independently.</p>	

Facility	Detail	See also
<b>Element Browser</b>	<p>You can also use the Element Browser window to select and display specific items on the Testing window; click on the  icon in the Testing window toolbar to display the Element Browser, open the Testing folder and select the required test record.</p> <p>In the folder, the 'page' icon against each record contains a:</p> <ul style="list-style-type: none"> <li>• <b>U</b> for Unit Test items</li> <li>• <b>I</b> for Integration Test items</li> <li>• <b>Sy</b> for System test items</li> <li>• <b>A</b> for Acceptance Test items</li> <li>• <b>Sc</b> for Scenario Test items</li> </ul>	<a href="#">Element Browser</a> [989]

### Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Tests** permission to add, update and delete test records
- The Testing window can be docked to the application workspace
- Columns in the List mode can be reorganized, added, removed, grouped, filtered and sorted using the options provided in the **List Header** facilities

### Learn more

- [Permission List](#)<sup>[329]</sup>
- [List Header](#)<sup>[677]</sup>

### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Testing | Testing | Testing Workspace**



## 17.2.2 Create Test Records

When you need to create or edit a test record on an element, for any of the five types of test, you can do so using either the Testing window in **Item** mode, or the Test Details dialog. The Test Details dialog displays when you select the **New** option or double-click on a test entry on the Testing window in **List** mode.

Whichever type of test you are recording, and whichever dialog you use, you complete the same fields.

**Access** **Select element | Element | Testing (Alt+3) > <test type tab> | Item mode: New toolbar icon** or **Select element | Element | Testing (Alt+3) > <test type tab> > List mode (:Show/Hide Properties if necessary): New toolbar icon**


### Create a test record

Field/Button	Action	See also
<b>Test</b>	Type the name of the test. This can be any alphanumeric text string you want to use.	
<b>Auto</b>	As an alternative to typing in the name, click on this button to insert predefined auto-counter text.  If you already have some text in the <b>Test</b> field, it is over-written by the auto-counter text.	<a href="#">Set Auto Naming and Auto Counters</a> 
<b>Status</b>	Click on the drop-down arrow and select the value indicating the current status of the test ( <b>Not Run, Pass, Fail, Deferred</b> and <b>Canceled</b> ).	
<b>Type</b>	Click on the drop-down arrow and select the value indicating the type of test ( <b>Load, Regression</b> or <b>Standard</b> ).	
<b>Run By</b>	Click on the drop-down arrow and select the name of the person who ran the test.	
<b>Checked By</b>	Click on the drop-down arrow and select the name of the person who checked the test run.	
<b>Last Run Date</b>	Click on the drop-down arrow and select the date on which the test was last run.	
<b>Description</b>	Type a description of the test; you can format the text using the Notes toolbar at the top of the field.  This text is also reflected in the Notes window, but cannot be edited there.	<a href="#">Notes Toolbar</a> 
<b>Input</b>	Type in the input data provided to the test; you can format the text using the Notes toolbar at the top of the field.  This text is also reflected in the Notes window, but cannot be edited there.	
<b>Acceptance Criteria</b>	Type the acceptance or test success conditions; you can format the text using the Notes toolbar at the top of the field.  This text is also reflected in the Notes window, but cannot be edited there.	
<b>Results</b>	Type the results of the last test; you can format the text using the Notes toolbar at the top of the field.	



Field/Button	Action	See also
	This text is also reflected in the Notes window, but cannot be edited there.	
<b>Previous</b>	(Test Details dialog) If earlier test records exist, click on this button to reset the fields to the values of the previous test record.	
<b>Next</b>	(Test Details dialog) If you are looking at an earlier test record, click on this button to reset the fields to the values of the next most recent test record.	
<b>New</b>	(Test Details dialog) Click on this button to clear the fields so that you can enter the information for a new test record.	
<b>OK</b>	(Test Details dialog) Click on this button to <b>save</b> the new or changed data <b>and close</b> the dialog.	
<b>Close</b>	(Test Details dialog) Click on this button to <b>close</b> the dialog <b>without</b> saving the new or changed data.	
<b>Apply</b>	(Test Details dialog) Click on this button to <b>save</b> the new or changed data without <b>closing the dialog</b> .	

### Notes

- On the Test Details dialog, you can add multiple test cases in one batch using the **New** and **Apply** buttons
- On the Test Details dialog, if you have several tests of one category (such as Unit or Integration), once you have saved a new test (click on the **Apply** button) or displayed an existing test, you can work backwards and forwards through any other existing tests of that category, by clicking on the **Previous** and **Next** buttons
- To display an existing item for editing, click on the item in the left-hand panel in Item mode, or double-click on the item in List mode to display the Test Details dialog
- To delete an item, right-click on it on the Testing window and click on the **Delete** icon in the window toolbar; in response to the confirmation prompt, click on the **Yes** button
- A further possibility for editing and deleting items is to right-click on items in the Element Browser and select menu options there; to access an item through the Element Browser, click on the  icon in the Testing window toolbar and click on the required item in the *Testing* folder in the Element Browser window
- To change the **element** for which to create or edit test items, click on the new element in the Project Browser
- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Tests** permission to add, update and delete test records

### 17.2.3 Move or Copy Tests Between Categories

After you define a test within one test category (Unit, Integration, System, Acceptance or Scenario), you might decide that the test either is better suited to another category, or forms a good template for tests in other categories. If so, you can either **move** or **copy** the test to the other categories.

**Access** **Select element | Element | Testing (Alt+3) > <test type tab> | right-click on test in list to move/copy** or  
**Select element | Element | Element Browser (Alt+9) > Testing folder | right-click on test to move/copy**

#### Move or copy a test

Step	Action	See also
1	Click on the appropriate option - <b>Move to</b> or <b>Copy to</b> . A list of test categories displays.	
2	Click on the test category to which to move or copy the test. A confirmatory prompt displays.	
3	Click on the <b>Yes</b> button to confirm the move or copy.	
4	Click on the target tab of the Testing window to ensure that the test has been added, and make any necessary changes.	<a href="#">Create Test Records</a> <sup>[2607]</sup>
5	If you are copying the test to more than one other category, repeat steps 1 to 4 for the next category to copy to.	

#### Notes

- If you move or copy a test into the **Scenario** category, some unassociated data could be lost

#### Learn more

- [Element Browser](#) <sup>[989]</sup>

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Testing | Testing | Changing Test Category**

## 17.2.4 Import Scenario as Test

If you are creating a test for a scenario from either a single element or many elements in a package, you do not have to manually re-type the scenario details into the test record in the Testing window. You can generate the test into the Scenario tab of one element from one or more scenarios in any element in the model.

Within the Scenario test record, the scenario description is copied to the Description tab. If a scenario contains a Structured Specification, its **Action** steps are also copied to the Description tab under the heading *Structured Specification*.

### Import a scenario from a single element

**Access** **Select target element | Element | Testing (Alt+3) > Scenario | right-click on tests | Import element scenario(s)** or  
**Select target element | Element | Element Browser (Alt+9) > Testing folder | right-click on tests | Import element scenario(s)**

Field/Button	Action	See also
<b>Select element</b>	If you are copying scenarios from a <b>different</b> element to the target element, click on the drop-down arrow and select the source element. The list identifies elements that have scenarios that can be imported.  Otherwise, leave this field blank.	
<b>Show related elements only</b>	Select this checkbox to restrict the list of selectable elements to those that are related to the target element.	
<b>Limit selection to these Object Types only</b>	If you want to restrict the list of selectable elements to only those of specific types, type in those element types in a comma-separated list.	
<b>Refresh</b>	Click on this button after changing any of the above field values, to refresh the list of available elements in the <b>Select element</b> field.	
<b>Select items to import</b>	Lists the scenarios defined in the source element. Select the scenario (s) to import.  If you do not use the <b>Select element</b> field, any scenarios listed are from the current element to which this test record belongs.  If you have selected a different element, the scenarios come from that element.	
<b>All</b>	Click on this button to select all scenarios listed in the <b>Select items to import</b> field.	
<b>None</b>	Click on this button to clear the selection of scenarios listed in the	

Field/Button	Action	See also
	<b>Select items to import</b> field.	
<b>Import</b>	Click on this button to import the selected scenario(s).	

#### Import scenarios from the elements in a package

Access **Select target element | Element | Testing (Alt+3) > Scenario > right-click on tests | Import Package Scenarios** or  
**Select target element | Element | Element Browser (Alt+9) > Testing folder > right-click on tests | Import Package Scenarios**

Field/Button	Action	See also
<b>Limit selection to these Object Types only</b>	If you want to restrict the selected elements to only those of specific types, type in those element types in a comma-separated list.  This version of the Import Scenario dialog lists all scenarios against all elements in the package; it does not enable you to select a specific element, but does enable you to filter the list of scenarios to those from specific types of element.	
<b>Refresh</b>	Click on this button after changing any of the above field values, to refresh the list of available elements in the <b>Select element</b> field.	
<b>Select Items to import</b>	Lists the scenarios defined in the selected elements in the package. Select the scenario(s) to import.	
<b>All</b>	Click on this button to select all scenarios listed in the <b>Select items to import</b> field.	
<b>None</b>	Click on this button to clear the selection of scenarios listed in the <b>Select items to import</b> field.	
<b>Import</b>	Click on this button to import the scenarios from each element as Scenario tests.	

#### Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Tests** permission to add, update

and delete test records

#### Learn more

- [Element Browser](#) 

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Execution Analysis | Testing | Import from Scenario**

### 17.2.5 Import Test From Other Elements

If you have created useful tests in one element, you can import those tests into any other element through the Testing window and so avoid having to duplicate the test information manually. You open the Testing window for the empty, target element and then select the source element that contains the tests to import.

**Access** **Select target element | Element | Testing (Alt+3) | Import tests from other element** or **Select target element | Element | Element Browser (Alt+9) > Testing folder | right-click on tests | Import tests from other element**

#### Import a test

Field/Button	Action	See also
<b>Select element</b>	Click on the drop-down arrow and locate and select the source element.  This list identifies elements that have tests that can be imported.	
<b>Show related elements only</b>	Select this checkbox to restrict the list of selectable elements to those that are related to the target element.	
<b>Limit selection to these Object Types only</b>	If you want to restrict the list of selectable elements to only those of specific types, type in those element types in a comma-separated list.	
<b>Refresh</b>	Click on this button after changing any of the above field values, to refresh the list of available elements in the <b>Select element</b> field.	
<b>Select items to import</b>	Lists the tests defined in the source element. Select the test(s) to import.  If you have not used the <b>Select element</b> field, any tests listed are from the <b>current</b> element to which this test record belongs. There is no purpose in importing these.	
<b>All</b>	Click on this button to select all tests listed in the <b>Select items to import</b> field.	

Field/Button	Action	See also
<b>None</b>	Click on this button to clear the selection of tests listed in the <b>Select items to import</b> field.	
<b>Import</b>	Click on this button to import the selected test(s).	

### Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Tests** permission to add, update and delete test records

### Learning Center topics

- (Alt+F1) | [Enterprise Architect](#) | [Execution Analysis](#) | [Testing](#) | [Import from Other Elements](#)

## 17.2.6 Import Responsibility or Constraint as Test

If you are creating a test against a **responsibility** (internal requirement) or internal **constraint** of an element, you do not have to manually **re-type** the details into the test record in the Testing window. You can **generate** a testing record on the element from the responsibility or constraint.

The test record is generated into the test-type tab that you currently have open, and the responsibility or constraint description is copied to the Description tab for the test record.

**Access** [Select element](#) | [Element](#) | [Testing](#) (Alt+3) > <test type tab> | right-click on test list | [Import element constraint\(s\)](#) or [Select element](#) | [Element](#) | [Testing](#) (Alt+3) > <test type tab> | right-click on test list | [Import element requirement\(s\)](#) or [Select element](#) | [Element](#) | [Element Browser](#) (Alt+9) > [Testing folder](#) | right-click on test of required type | <option>

### Import a requirement or constraint as a test

Step	Action	See also
<b>1</b>	On the Import Constraint or Import Requirements dialog (the two dialogs are identical) review the list of internal requirements or constraints in the selected element.	
<b>2</b>	Click on one of the items to import as a test, or <b>(Ctrl)+click</b> on more than one to import several.  Click on the <b>OK</b> button.  Each item is added to the list of tests in the Testing window, on the selected <test-type> tab, as a standard, 'Not Run' test.	

Step	Action	See also
3	Edit the items to complete their definition as tests.	

### Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Manage Tests** permission to add, update and delete test records

### Learn more

- [Element Browser](#)

### Learning Center topics

- (Alt+F1) | **Enterprise Architect** | **Execution Analysis** | **Testing** | **Import from Constraints**
- (Alt+F1) | **Enterprise Architect** | **Execution Analysis** | **Testing** | **Import from Responsibility**


## 17.2.7 Create Maintenance Item From Test

If an element fails a test, one likely consequence is that a Defect (Issue) item has to be raised in model maintenance to correct the problem. You can generate this Defect item directly from the test that failed.

**Access** **Select element | Element | Testing (Alt+3) > <test type tab> | right-click on test in list | Create a Maintenance Defect from this test** or  
**Select element | Element | Element Browser (Alt+9) > Testing folder | right-click on test | Create a Maintenance Defect from this test**

### Create a Maintenance item from a test

Step	Action	See also
1	The system immediately creates the Defect item and displays a confirmation message box.  Click on the <b>OK</b> button to clear the message.	
2	Open the Maintenance window ( <b>Element   Maintenance</b> ), which defaults to the Defects tab.  The tab shows a Defect item having the same name as the test. The <b>Description</b> , <b>Input</b> , <b>Acceptance Criteria</b> and <b>Results</b> texts from the test are all displayed in the defect Description tab under separate headings.	

Step	Action	See also
3	Complete the Defect item as necessary - you might provide values for the <b>Reported By</b> , <b>Status</b> and <b>Priority</b> fields.	<a href="#">Create Maintenance Items</a>  <small>2625</small>

#### Notes

- You can create Maintenance Defect items from several Test items at once; press and hold **(Shift)** as you select the Test items, and then right-click and proceed as above - **each** selected Test item then generates a Defect item

#### Learning Center topics

- (Alt+F1) | Enterprise Architect | Testing | Testing | Create Maintenance Items**

### 17.2.8 Show Test Script Compartments

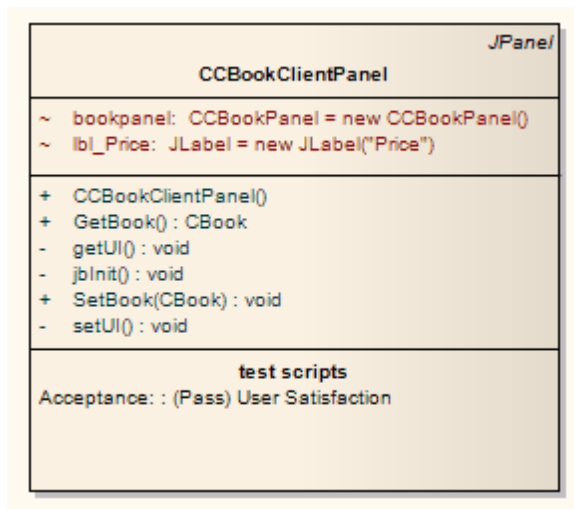
When you have created a Test record, it is useful to make the test **visible** on its parent **element**. You can do this by displaying the test within a Test Script **compartment** on the element as it **displays** in a diagram. Any element that is capable of displaying a compartment, and that has a test assigned to it, can show test scripts in a diagram.

#### Show the Test Scripts on an element in a diagram

Step	Action	See also
1	Open a diagram containing the element with the attached test items.	
2	Double-click on the diagram background to display the Diagram Properties dialog.  Click on the Elements tab.	
3	In the Show Compartments panel, select the <b>Testing</b> checkbox.	
4	Click on the <b>OK</b> button to save the setting.  Each test now appears as an item in the test scripts compartment of the diagram element.	

#### Example





#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Testing | Testing | Show on Diagrams**

### 17.2.9 Test Documentation

After you have recorded a number of test scripts and results against elements in the model, you can output the information as a **report** in Rich Text Format, using the Generate Test Documentation dialog. You can configure which tests to **include or exclude** in the report, whether to include **child packages**, and the file location to which to generate the report.

Access    **Project Browser | right-click on package | Documentation | Testing Report**    or  
**Project | Documentation | Testing Report**

#### Learn more

- [Testing Report](#)<sup>[2742]</sup>
- [Testing](#)<sup>[2604]</sup>

#### Learning Center topics

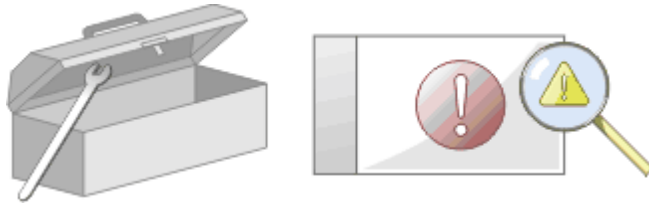
- (Alt+F1) | **Enterprise Architect | Testing | Testing | Test Reports**

**Part**

---



## 18 Maintenance



In the course of your team's work on a model changes and issues can arise at a number of levels, for problems that apply system-wide through areas of the model and down to within a specific element. A change, very broadly, defines an addition or alteration to a requirement, whilst an issue identifies either a failure to meet a requirement, or a risk in meeting the requirement.

There are two mechanisms that can be used to identify a change or issue, and the work required to resolve it:

- Change and Issue (or Defect) **elements** - structured comments that identify a problem at system-level, although they can also be attached to a specific element from which a problem arises; both types of element can be linked to one or more other elements that have to be reviewed, with relationships such as Association, Dependency and Realize, and for complex problems can also form hierarchies or groups
- Maintenance items - properties raised against a specific element, and recorded for that element in the Maintenance window; these provide a distinction between Defects (a failure to meet a requirement) and Issues (a risk factor that might affect satisfying the requirement) and also include Tasks, which record **work items** associated with the element

Maintenance items are very specific, but if an item begins to have a wider impact on other elements or the system in general, you can translate that item into a Change, Issue, or any other type of element that best identifies the problem and its solution

*Maintenance items* are defects, changes, issues and tasks that apply at the **model element level**. They are properties of individual model elements that can be used to record and capture problems, changes, issues and tasks as they arise, and document the solution and associated details.

### Maintenance Tasks

Task Area	Detail	See also
<b>Maintenance Items - Element-level</b>	<p>You create maintenance items in the <b>Maintenance window</b>, and using the <b>Item Details</b> dialog. Typically you create:</p> <ul style="list-style-type: none"> <li>• A <b>defect</b> to record a failure to meet a requirement for the current model element</li> <li>• A <b>change</b> to record a change in requirement for the current model element</li> <li>• An <b>issue</b> to record a risk factor that might affect the project being recorded for the current model element</li> <li>• A <b>task</b> to record work in progress and work outstanding for the current model element</li> </ul>	<a href="#">Working on Maintenance Items</a> <small>[2623]</small> <a href="#">Create Maintenance Items</a> <small>[2625]</small>
<b>Creating and Editing Maintenance</b>	<p>Tasks that you might perform when working with maintenance items include:</p> <ul style="list-style-type: none"> <li>• Moving or copying maintenance items between</li> </ul>	<a href="#">Move or Copy Maintenance Items</a> <small>[2628]</small>

Task Area	Detail	See also
<b>Items</b>	<p>maintenance types, or generate new items of one type from an item of a different type</p> <ul style="list-style-type: none"> <li>• Create elements from maintenance Items</li> <li>• Show maintenance items on elements in a diagram</li> <li>• Add a maintenance item directly to an element via in-place editing</li> <li>• Generate a report on the status of maintenance items of specific types or all types</li> </ul>	<p><a href="#">Create Elements From Maintenance Items</a> <sup>[2629]</sup></p> <p><a href="#">Show Maintenance Items in Diagram</a> <sup>[2630]</sup></p> <p><a href="#">Insert Maintenance Feature</a> <sup>[1042]</sup></p> <p><a href="#">Maintenance Report</a> <sup>[2740]</sup></p>
<b>Maintenance - System-level</b>	<p>To represent changes, defects and issues that apply to the model you can create Change and Issue elements with - if necessary - subordinate structures.</p> <p>To represent issues and tasks that apply to the progress of the project as a whole and that are not related to model structures, you create Project Status records in Project Management.</p>	<p><a href="#">Changes and Issues</a> <sup>[2631]</sup></p> <p><a href="#">Maintenance Diagram</a> <sup>[2634]</sup></p> <p><a href="#">The Project Status View</a> <sup>[525]</sup></p>

#### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Maintenance**

## 18.1 Introduction to Maintenance



### Introduction to Maintenance

Topic	Detail	See also
<b>Maintenance Items</b>	<p><i>Maintenance</i> items are defects, changes, issues and tasks. They all apply to individual model elements and can be used to record and capture problems, changes, issues and tasks as they arise, and document the solution and associated details. They are defined as follows:</p> <ul style="list-style-type: none"> <li>• A <b>defect</b> can be considered as a failure to meet a requirement for the current model element</li> <li>• A <b>change</b> can be considered as a change in requirement for the current model element</li> <li>• An <b>issue</b> records a risk factor that might affect the project being recorded for the current model element</li> <li>• A <b>task</b> is a means of recording work in progress and work outstanding for the current model element</li> </ul> <p>Note that each of these maintenance items applies at the <b>model element level</b>. For changes, defects and issues that apply to the <b>whole system</b>, see the <i>Changes and Defects</i> topic; for tasks that apply to the whole system, see the <i>Project Tasks</i> topic.</p>	<p><a href="#">Changes and Issues</a> <sup>[2631]</sup></p> <p><a href="#">Project Tasks</a> <sup>[526]</sup></p>
<b>Creating and Editing Maintenance Items</b>	<p>The following topics explain how to create and edit Maintenance items:</p> <ul style="list-style-type: none"> <li>• <i>Working on Maintenance Items</i> - describes the Maintenance window</li> <li>• <i>Create Maintenance Items</i> - describes how to complete the Maintenance window tabs for the various maintenance items</li> <li>• <i>Move or Copy Maintenance Items</i> - describes how to move items between maintenance categories or generate items from an item in a different category</li> <li>• <i>Create Elements From Maintenance Items</i> - describes how to generate elements from maintenance items</li> <li>• <i>Show Maintenance Items in Diagram</i> - describes how to display maintenance items in elements on diagrams</li> <li>• <i>Insert Maintenance Feature</i> - describes how to add a maintenance item directly to an element via in-place editing</li> </ul>	<p><a href="#">Working on Maintenance Items</a> <sup>[2623]</sup></p> <p><a href="#">Create Maintenance Items</a> <sup>[2625]</sup></p> <p><a href="#">Move or Copy Maintenance Items</a> <sup>[2628]</sup></p> <p><a href="#">Create Elements From Maintenance Items</a> <sup>[2629]</sup></p> <p><a href="#">Show Maintenance Items in Diagram</a> <sup>[2630]</sup></p> <p><a href="#">Insert Maintenance Feature</a> <sup>[1042]</sup></p>

Learning Center topics

- (Alt+F1) | **Enterprise Architect | Maintenance**


## 18.2 Working on Maintenance Items


Creating, viewing and modifying the Maintenance items - changes, issues, defects and tasks - associated with a particular model element is quick and convenient, using the **Maintenance window** or **Workspace**. If the Maintenance window is open, when you select an element in a **diagram** or in the **Project Browser**, the Maintenance Items for that element are immediately listed in the window ready for modification or addition. The window provides several facilities for creating and managing the Maintenance items.

You can include the maintenance items in the document and web reports generated on your model. The Document Setup dialog has checkboxes to show or hide element maintenance items.

**Access**    **Element | Maintenance**    (Alt+4)

### Facilities

Facility	Detail	See also
<b>Window Formats</b>	<p>The Maintenance window has two formats - <b>Item</b> mode and <b>List</b> mode.</p> <ul style="list-style-type: none"> <li><b>Item</b> mode provides a list on the left hand side from which you select a maintenance item, the details of which then display in the fields in the rest of the window; you can edit these fields directly</li> <li><b>List</b> mode consists of a list of maintenance items with summary data distributed across the columns; you cannot directly edit these fields</li> </ul> <p>To toggle between the modes, click on the  (<b>Show/Hide Properties</b>) button in the window toolbar.</p> <p>When you first open the window, in either mode, the oldest maintenance item of the type is shown at the top of the list and, in Item mode, in the data fields.</p>	
<b>Adding New Items</b>	<p>To add new items, click on the <b>New</b> icon in the window toolbar. In:</p> <ul style="list-style-type: none"> <li>Item mode, this clears the fields for new data</li> <li>List mode, this displays the &lt;item type&gt; details for &lt;element type&gt; &lt;element name&gt; dialog, which you complete in the same way as the Maintenance window in Item mode</li> </ul> <p>The fields you complete for each <b>type</b> of maintenance item are almost identical.</p>	<a href="#">Create Maintenance Items</a> <sup>[2625]</sup>
<b>Applying Automatic Naming/ Numbering</b>	<p>On the Maintenance window in Item mode, or on the item details dialog, you can apply an automatic naming/numbering convention that you have previously defined, to each new item record. To do this, simply click on the <b>Auto</b> button next to the <b>Name</b> field.</p> <p>If you already have some text in the <b>Name</b> field, it is over-written by the auto-counter text.</p>	<a href="#">Set Auto Naming and Auto Counters</a> <sup>[904]</sup>

Facility	Detail	See also
<b>Maintenance Item Types</b>	<p>The four types of Maintenance Item you can create records for are each managed on a separate tab of the window:</p> <ul style="list-style-type: none"> <li>• <b>Defects</b> - each recording a failure to meet a requirement for the current model element</li> <li>• <b>Changes</b> - each recording a change in requirement for the current model element</li> <li>• <b>Issues</b> - each recording a risk factor that might affect the project, associated with the current model element</li> <li>• <b>Tasks</b> - each recording work in progress and work outstanding for the current model element</li> </ul> <p>The Maintenance window opens at the Defects tab by default.</p> <p>An asterisk on a tab indicates that the tab contains saved information; if there is no information for a type of item, or the information has not yet been <b>saved</b>, its tab has no asterisk.</p> <p>You toggle the display for each tab between Item mode and List mode independently.</p>	
<b>Element Browser</b>	<p>You can also use the Element Browser window to select and display specific items on the Maintenance window; click on the  icon in the Maintenance window toolbar to display the Element Browser, open the <i>Maintenance</i> folder and select the required item.</p> <p>In the folder, the 'page' icon against each item contains a:</p> <ul style="list-style-type: none"> <li>• <b>C</b> for Change items</li> <li>• <b>D</b> for Defect items</li> <li>• <b>T</b> for Task items, or</li> <li>• <b>I</b> for Issue items</li> </ul>	<a href="#">Element Browser</a> <sup>[989]</sup>

**Notes**

- Columns in the item list can be reorganized, added, removed, grouped, filtered and sorted using the options provided in the **List Header** facilities

**Learn more**

- [Maintenance](#)<sup>[2619]</sup>
- [List Header](#)<sup>[677]</sup>



## 18.3 Create Maintenance Items


When you need to create or edit a maintenance record on an element on a diagram or in the Project Browser, for any of the four types of item, you can do so using either the Maintenance window in **Item** mode, or the <item type> Details for <element type> <element name> dialog. The Item Details dialog displays when you select the **New** option or double-click on an entry on the Maintenance window in **List** mode.

Whichever type of item you are recording, and whichever dialog you use, you complete effectively the same fields.


**Access** **Select element | Element | Maintenance (Alt+4) > <item type tab> > Item mode: New toolbar icon** or  
**Select element | Element | Maintenance (Alt+4) > <item type tab> > List mode (:Show/Hide Properties if necessary): New toolbar icon**

### Create maintenance items

Field/Button	Usage	See also
<b>Name</b>	Type the name or a short description of the defect, change, task or defect.	
<b>Auto</b>	As an alternative to typing in the name, click on this button to insert predefined auto-counter text.  If you already have some text in the <b>Name</b> field, it is over-written by the auto-counter text.	<a href="#">Set Auto Naming and Auto Counters</a> [904]
<b>Reported by Requested Raised by</b>	Click on the drop-down arrow and select the user name of the person who initiated the maintenance item.	
<b>Date</b>	Defaults to today's date as the date on which the maintenance item was raised; if necessary, change this by clicking on the drop-down arrow and selecting a different date.	
<b>Status</b>	Click on the drop-down arrow and select the appropriate status of the maintenance item, such as <b>New</b> or <b>Complete</b> .	
<b>Resolved by Implemented by Completed by</b>	Click on the drop-down arrow and select the user name of the person who completed and closed the maintenance item.	
<b>Date Resolved Date Date Completed</b>	Defaults to today's date as the date on which the maintenance item was completed; click on the checkbox to select today's date or, if necessary, change the field by clicking on the drop-down arrow and selecting a different date.	

Field/Button	Usage	See also
<b>Priority</b>	Click on the drop-down arrow and select the priority for completing the maintenance item.	
<b>Version</b>	Type the version number associated with this defect, change, issue or task.	
<b>Description</b>	Type a longer description of the maintenance item; you can format the text using the Notes toolbar at the top of the field.  This text is also reflected in the Notes window, but cannot be edited there.	<a href="#">Notes Toolbar</a> 
<b>History</b>	Enter any notes or references to previous occurrences of this maintenance item; you can format the text using the Notes toolbar at the top of the field.  This text is also reflected in the Notes window, but cannot be edited there.	
<b>New</b>	Click on this button to clear the fields ready to create a new maintenance item.	
<b>OK</b>	Click on this button to save the record data and close the dialog.	
<b>Close</b>	Click on this button to close the dialog without saving the record.	
<b>Apply</b>	Click on this button to save the record without clearing the data fields or closing the dialog.	

### Notes

- To edit an item, right-click on it on the Maintenance window and select the **Modify Selected** option; the Item Details dialog displays with the values for the selected item in the fields, and you can change those values
- To delete an item, right-click on it on the Maintenance window and click on the **Delete** icon in the window toolbar; in response to the confirmation prompt, click on the **Yes** button
- A further possibility for editing and deleting items is to right-click on items in the Element Browser and select menu options there; to access an item through the Element Browser, click on the  icon in the Maintenance window toolbar and click on the required item in the *Maintenance* folder in the Element Browser window
- To change the **element** for which to create or edit maintenance items, click on the new element in the Project Browser

### Learn more

- [Maintenance](#) 


- [Working on Maintenance Items](#) 

## 18.4 Move or Copy Maintenance Items

After you define a maintenance item of one type (Defect, Change, Issue or Task), you might decide that the item either is better suited to another type or forms a good template for items of other types. If so, you can either **move** or **copy** the item to the other types.

**Access** **Select element | Element | Maintenance (Alt+4) > <item type tab> | right-click on item in list to move/copy** or  
**Select element | Element | Element Browser (Alt+9) > Maintenance folder | right-click on item to move/copy**

### Move or copy a maintenance item

Step	Action	See also
1	Click on the appropriate option - <b>Move to</b> or <b>Copy to</b> . A list of maintenance item types displays.	
2	Click on the item type to which to move or copy the item. A confirmatory prompt displays.	
3	Click on the <b>Yes</b> button to confirm the move or copy.	
4	Click on the target tab of the Maintenance window to ensure that the item has been added, and make any necessary changes.	<a href="#">Create Maintenance Items</a> 
5	If you are copying the item to more than one other item type, repeat steps 1 to 4 for the next item type to copy to.	

### Learn more

- [Working on Maintenance Items](#) 

## 18.5 Create Elements From Maintenance Item

A maintenance item identifies a defect, change, issue or task concerning an element. The maintenance item could **itself** be represented by an **element** if it has wider implications for the project or identifies - for example - an actor, activity or action that requires further definition.

You can create one or more elements from any maintenance item, using the Maintenance window. The new element is connected to the maintenance item's **parent element** by a Dependency connector. The item itself remains unchanged as a characteristic of its parent element.

**Access** **Element | Maintenance (Alt+4) | <Item Type tab> | right-click item in list | Create as New Element**

### Create an element from a maintenance item

Step	Action	See also
1	<p>On the New Element dialog, complete the fields (follow the link on the right).</p> <p>In the <b>Type</b> field, you select the <b>required element</b> type; you might create:</p> <ul style="list-style-type: none"><li>• An Issue element for a Defect or Issue maintenance item</li><li>• A Change element for a Change item, or</li><li>• An Action for a Task item</li></ul> <p>You can, however, create a wide range of <b>other</b> element types should any of these be more appropriate.</p> <p>When you click on the <b>Create</b> button, the new element is added to the Project Browser. You can drag and drop it onto a diagram from there.</p>	<a href="#">Add Elements Directly To a Package</a> <sup>[903]</sup>

### Learn more

- [Working on Maintenance Items](#)<sup>[2623]</sup>

### Learning Center topics

- (Alt+F1) | **Enterprise Architect | Maintenance | Element from Item**

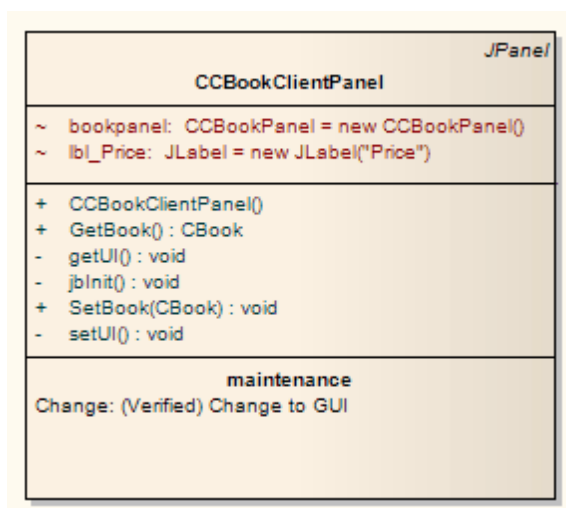
## 18.6 Show Maintenance Items in Diagram

When you have created a maintenance item, it is useful to make the record **visible** on its parent element. You can do this by displaying the record within a maintenance **compartment** on the **element** as it displays in a diagram. Any element that is capable of displaying a compartment, and that has maintenance items assigned to it, can show the items in a diagram.

### Show maintenance items in a diagram

Step	Action	See also
1	Open a diagram containing the element with the attached maintenance items.	
2	Double-click on the diagram background to display the Diagram Properties dialog. Click on the Elements tab.	
3	In the Show Compartments panel, select the <b>Maintenance</b> checkbox.	
4	Click on the <b>OK</b> button to save the setting.  Each maintenance item now appears in the maintenance compartment of the diagram element.	

### Maintenance Compartment - Example



### Learn more

- [Working on Maintenance Items](#) <sup>2623</sup>

## 18.7 Changes and Issues

If your work in managing a project identifies issues, defects or required changes in the **model** rather than internal to a single element, you can represent these using Change or Issue (Defect) elements as structured comments with which you can track and manage the problems.

- A **Change** element corresponds to a **change in requirements** for the current system
- An **Issue** element corresponds to a **failure to match** the requirements for the current system

You can also define any specific work to be done in researching or resolving the problem using **Task** elements, and assign **resources** to these or directly to the Change or Issue elements internally, in Resource Allocation, or externally as Actors.

The use of maintenance **elements** provides a broad scope for managing changes and issues, providing the facility to fully define the issue and its resolution with linked documents, both internal and external Notes, and composite structure diagrams including Interaction and Activity diagrams.

You can create Change and Issue elements in various UML diagrams (especially **Maintenance diagrams**) and connect them using Realization, Dependency, Aggregation and other relationships to show what model elements each issue affects and how each is resolved. Within the element Properties dialog for Changes and Issues you can identify the issue as the element name and record relevant management details, such as owner and dates.

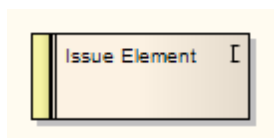
A useful tool in managing issues and changes is the Relationship Matrix, in which you might - for example - link staff (Actors) through Realization connectors to Issues. Each highlighted square in the Matrix indicates the responsibility of a staff member to work on or correct a named Issue

### Learn more

- [Changes](#) <sup>[2633]</sup>
- [Issues \(Defects\)](#) <sup>[2631]</sup>
- [Task](#) <sup>[2010]</sup>
- [Maintenance Diagram](#) <sup>[2634]</sup>
- [Relationship Matrix](#) <sup>[727]</sup>
- [Resource Allocation](#) <sup>[512]</sup>

### 18.7.1 Issues (Defects)

In your modeling issues concerning, or defects in, the development of the system or model might arise; more formally, you encounter a failure to meet defined requirements for the current system. You can represent this failure using an **Issue (or Defect)** element, which is a structured comment containing information about the issue and the measures taken to manage it. The element is rendered as shown.



You can link Issues to model elements that are responsible for the defect, using *Realize* connectors. You can also create a hierarchy of related Issues using *Aggregation* connectors. Each Issue element has a status band at the left end, which is color coded to visually represent the value of the **Status** field in the element properties. The element has an identifying **I** in the top right corner, which you can hide if you prefer not to show it.

You can create Issue elements on most types of diagram, although the Maintenance diagram is specifically designed for displaying and managing them. You can also create your own issue-management diagram as a Custom diagram. You can add the Issue (and other) elements to the diagram from the Diagram Toolbox, or directly to a package in the Project Browser.

#### Add an Issue to the model using the Toolbox

Step	Action	See also
1	Open the Maintenance or Custom (or other) diagram to depict the problem.	
2	From the Custom pages or Common page of the Diagram Toolbox, drag the <b>Issue</b> icon onto the diagram.	<a href="#">Custom Toolbox</a> <sup>[813]</sup> <a href="#">Common Page</a> <sup>[800]</sup>
3	If the Properties dialog does not automatically display, double-click on the element. Record the name and details of the Issue, in the Properties dialog.	<a href="#">Properties Dialog</a> <sup>[956]</sup> <a href="#">Create Maintenance Items</a> <sup>[2625]</sup>

#### Add an Issue to the model using the New Element dialog

Step	Action	See also
1	Identify the package in which to record the Issue, in the Project Browser.  Right-click on this package and select the <b>Insert   New Element</b> context menu option.	
2	Complete the fields on the New Element dialog - in the <b>Type</b> field, click on the drop-down arrow and select <b>Issue</b> .	<a href="#">Add Elements Directly To Packages</a> <sup>[903]</sup>
3	When you have created the element, double-click on it. Record the name and details of the Issue, in the Properties dialog.	<a href="#">Properties Dialog</a> <sup>[956]</sup> <a href="#">Create Maintenance Items</a> <sup>[2625]</sup>

#### Notes

- To toggle display of the letter I in the top right corner of the element, select or deselect the **Show**



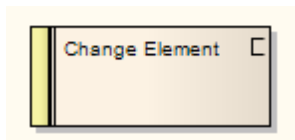
**stereotype icon for requirements** checkbox on the Options dialog, Objects page

#### Learn more

- [Changes and Issues](#) <sup>[2631]</sup>
- [Color Codes](#) <sup>[1776]</sup>
- [Object Display Options](#) <sup>[631]</sup>

## 18.7.2 Changes

In your modeling it might become necessary to change an aspect of the system or model; more formally, you need to request and manage a change in the defined requirements for the current system. You can represent this change request using a **Change** element, which is a structured comment containing information about the change and the measures taken to manage it. The element is rendered as shown.



You can link Changes to model elements that are impacted by the change, using *Realize* connectors. You can also create a hierarchy of related Changes using *Aggregation* connectors. Each Change element has a status band at the left end, which is color coded to visually represent the value of the **Status** field in the element properties. The element has an identifying **C** in the top right corner, which you can hide if you prefer not to show it.

You can create Change elements on most types of diagram, although the Maintenance diagram is specifically designed for displaying and managing them. You can also create your own change-management diagram as a Custom diagram. You can add the Change (and other) elements to the diagram from the Diagram Toolbox, or directly to a package in the Project Browser.

#### Add a Change to the model using the Toolbox

Step	Action	See also
1	Open the Maintenance or Custom (or other) diagram to depict the problem.	
2	From the Custom pages or Common page of the Diagram Toolbox, drag the <b>Change</b> icon onto the diagram.	<a href="#">Custom Toolbox</a> <sup>[813]</sup> <a href="#">Common Page</a> <sup>[800]</sup>
3	If the Properties dialog does not automatically display, double-click on the element. Record the name and details of the Change, in the Properties dialog.	<a href="#">Properties Dialog</a> <sup>[956]</sup> <a href="#">Create Maintenance Items</a> <sup>[2625]</sup>

**Add a Change to the model using the New Element dialog**

Step	Action	See also
1	Identify the package in which to record the change, in the Project Browser.  Right-click on this package and select the <b>Insert   New Element</b> context menu option.	
2	Complete the fields on the New Element dialog - in the <b>Type</b> field, click on the drop-down arrow and select <b>Change</b> .	<a href="#">Add Elements Directly To Packages</a> <sup>[903]</sup>
3	When you have created the element, double-click on it. Record the name and details of the Change, in the Properties dialog.	<a href="#">Properties Dialog</a> <sup>[956]</sup>  <a href="#">Create Maintenance Items</a> <sup>[2625]</sup>

**Notes**

- To toggle display of the letter **C** in the top right corner of the element, select or deselect the **Show stereotype icon for requirements** checkbox on the Options dialog, Objects page

**Learn more**

- [Changes and Issues](#) <sup>[2631]</sup>
- [Color Codes](#) <sup>[1776]</sup>
- [Object Display Options](#) <sup>[631]</sup>

**18.7.3 Maintenance Diagram**

A Maintenance diagram is a type of Custom diagram, an extension to the UML model. It is a change management tool, used to record:








- Requests for change to the model structure or project process, as **Change** elements
- Issues that impact the development and progress of the project, as **Issue** elements, and
- Groups of tests that can be applied to many elements rather than specific elements, as **Test Cases**

Each Change, Issue or Test Case element can link to other model elements in the project, to illustrate how they contribute to or are impacted by the item, and how they must be modified, removed, applied or extended to provide a solution. This includes identifying areas of work, represented by Task elements, to which resources can be allocated as either external (Actor) elements or internal properties (resource allocation).

**Example Diagram** [Example Maintenance Diagram](#) <sup>[2635]</sup>

## Tools

Select Maintenance diagram elements and connectors from the Maintenance pages of the Toolbox. For further information on a type of element or connector, click on the corresponding icon.

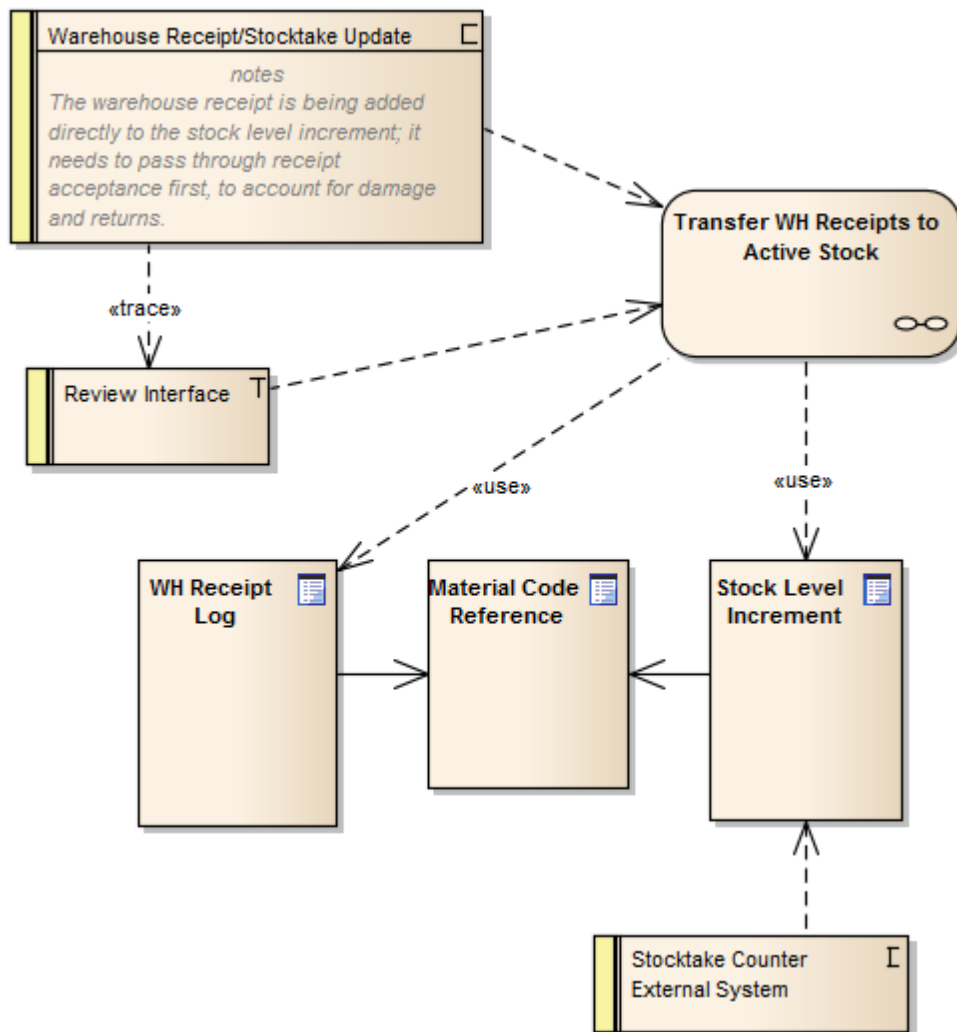
Maintenance Diagram Elements	Maintenance Diagram Connectors
 Package	 Aggregate
 Issue	
 Change	
 Task	
 Test Case	
 Entity	

## Learn more

- [Custom Diagram](#) <sup>[1796]</sup>
- [Actor](#) <sup>[1284]</sup>
- [Resource Allocation](#) <sup>[512]</sup>

### 18.7.3.1 Example Maintenance Diagram

Very simply, this example diagram depicts a request for a change (C) to a process represented by an Activity element (Transfer WH Receipts to Active Stock). The change currently involves the Task (T) of reviewing the interface between two recording systems, one of which could be affected by an Issue (I) concerning an external stocktaking system.



**Part**

---








## 19 Reporting



As you develop your models in Enterprise Architect, you will want to report on, review and disseminate information on a number of aspects of the model through both online and printed documentation. Enterprise Architect provides a powerful mechanism for generating high quality, customized documentation directly from the model, in document format or web page format.

### Report Generation Facilities

Facility	Detail	See also
<b>Specifying Content</b>	<p>There are many ways to specify the model content being documented; you can:</p> <ul style="list-style-type: none"> <li>Document a Package and/or its child Packages by manually highlighting the Package and selecting a documentation control</li> <li>Specify embedded Packages for <b>exclusion</b> if child Packages are recursively documented</li> <li>Generate your documentation using a range of <b>system-provided templates</b> that support many different aspects of your model, or develop your own, <b>customized templates</b> to meet your specific requirements</li> <li>Use system-provided or user-customized Stylesheets, Tables of Contents and Cover Pages to further tailor individual reports or sets of reports, independent of the template used</li> <li>Link a Package to a document template to simplify generating consistent types of documentation (for example, Use Case reports) using the <b>Resource Documents</b> feature</li> <li>Select, group and order packages in a different organization from the Project Browser by creating <b>'Virtual' Documents</b>, either linked through a master document with headers, footers and contents list, or as separate individual documents</li> </ul>	<p><a href="#">Generate Documents</a> <small>[2642]</small></p> <p><a href="#">Exclude Package From Report</a> <small>[2761]</small></p> <p><a href="#">Document Templates</a> <small>[2681]</small></p> <p><a href="#">Generate Documentation</a> <small>[2644]</small></p> <p><a href="#">Save As Document</a> <small>[2728]</small></p> <p><a href="#">Virtual Documents</a> <small>[2669]</small></p>
<b>Document Reports</b>	<p>You can generate Document reports in a number of document formats, which you can tailor using templates and Stylesheets to alter the content and appearance of the generated output.</p> <p>Enterprise Architect has a fully-featured document generator that features:</p> <ul style="list-style-type: none"> <li>Powerful WYSIWYG template editor support</li> <li>An easy-to-use document generator</li> </ul>	<p><a href="#">Document Reports</a> <small>[2640]</small></p> <p><a href="#">Generate Documents</a> <small>[2642]</small></p> <p><a href="#">Design Custom Document Templates</a> <small>[2684]</small></p>

Facility	Detail	See also
	<ul style="list-style-type: none"> <li>An embedded viewer that enables you to view documents generated within the system</li> </ul> <p>Using Virtual Documents, you can draw out information on structurally separate model elements in a set of individual reports, or in a combined report under a <b>master document</b> with common headers, footers and styles.</p>	<a href="#">Virtual Documents</a>  <sup>[2669]</sup>
<b>System Reports</b>	You can also generate a number of system-provided reports on different aspects of your model, such as the elements in a package in Implementation relationships or in Dependency relationships.	<a href="#">System Documents</a>  <sup>[2736]</sup>
<b>Web Documentation</b>	<p>It is possible to provide automated web-based publishing of your Enterprise Architect models, making it simple to explore large models efficiently on-line.</p> <p>You can export an entire model or a single branch of the model to Web pages, and also create web style templates to customize the web page output.</p>	<a href="#">Web Reports</a>  <sup>[2744]</sup>
<b>Charts</b>	You can capture statistical information on a range of element properties across the model, and display this information on a diagram in one of several formats, such as a pie chart, bar chart or 3-D column chart. You can also print and distribute the charts you have created, using the standard Diagrams Only Report facility.	<a href="#">Charts</a>  <sup>[2762]</sup> <a href="#">Diagrams Only Report</a>  <sup>[2741]</sup>

#### Learning Center topics

- **Alt+F1 | Enterprise Architect | Reporting | Rich Text Reports | Generate an RTF Report**
- **Alt+F1 | Enterprise Architect | Reporting | Virtual Documents | Generate RTF**
- **Alt+F1 | Enterprise Architect | Reporting | Virtual Documents | Generate HTML**
- **Alt+F1 | Enterprise Architect | Reporting | HTML | Generating an HTML Report**





## 19.1 Document Reports

If you want to produce immediate, screen-viewable or printable reports on the contents of your model, you can generate documents in a number of formats, on selected Packages or elements. You have the options of:



- Generating one of a set of system-provided reports on specific aspects of your model
- Generating a report from a system-provided template, with control of the report output through Stylesheets, add-in cover pages and tables of contents, filters and a wide range of dialog options
- Creating your own templates, Stylesheets and add-in pages from which to generate reports, applying an even greater degree of control over the contents and format of the output

The document formats are common to many word processors, so you can also work on the reports in other word-processing tools to, for example, integrate the report into a larger, externally-created document.

### Documentation Solutions

Feature	Detail	See also
<b>Document Generation</b>	<p>The Document Generator features:</p> <ul style="list-style-type: none"> <li>• A powerful WYSIWYG Document editor that you can use to: <ul style="list-style-type: none"> <li>• Import and export document templates using XML</li> <li>• Customize copies of the basic, system-provided templates</li> <li>• Create entirely new templates of your own design</li> <li>• Test templates as you design them, by generating ad-hoc reports within the designer</li> <li>• Create and customize independent Stylesheets, Tables of Contents and Cover Sheets to apply to any of your reports</li> </ul> </li> <li>• A documentation generator that: <ul style="list-style-type: none"> <li>• Provides a wide range of simple options for filtering and modifying the report output</li> <li>• Generates detailed and comprehensive documents based on the templates</li> <li>• Has an embedded viewer for reading the generated documents directly within Enterprise Architect</li> </ul> </li> </ul> <p>A tutorial on using the Document Generator and creating document reports is available on the Sparx Systems website.</p>	<p><a href="#">Design Custom Document Templates</a>  <sup>2684</sup></p> <p><a href="#">Document Templates</a>  <sup>2681</sup></p> <p><a href="#">Generate Documents</a>  <sup>2642</sup></p> <p><a href="http://www.sparxsystems.com/resources/whitepapers/">http://www.sparxsystems.com/resources/whitepapers/</a></p>
<b>Virtual Documents</b>	<p>If you want to create very focussed and specific documents, with a greater complexity and presenting different views of your model data, you can do so using <b>virtual documents</b>. For these, you set up a Master Document (package) element and/or Model Document elements, and link packages or model searches (for elements) into the document in whatever order or combination is</p>	<p><a href="#">Virtual Documents</a>  <sup>2669</sup></p>





Feature	Detail	See also
	<p>most appropriate to your requirements.</p> <p>You can select packages from different areas of the model, arrange them in any order, and edit or delete the packages; the virtual document automatically incorporates the changes each time you generate it.</p>	
<b>Word Master Documents</b>	<p>As an alternative to virtual documents, you can link a number of document reports into a single Word <b>master document</b>. This is quite possible with the current report generator but is of greater use with the original, Legacy RTF Report Generator.</p> <p>Typically you create a Word master document, then in Enterprise Architect generate some reports that you link back into sub-sections of the master document and refresh as required during project development. In this way the project document becomes an easily-managed and feature-rich work product.</p> <p>You can also populate a Word document from specific sections of reports, based on <b>bookmarks</b> - for example, a Word document might have a section for a small part of your component model; using bookmarks you can generate a full component model, and then link into just one section of the report.</p> <p>This way you can maintain a complex Word document from parts of Enterprise Architect reports; the Document Generator performs one pass for one template, but using a Word master document and Enterprise Architect bookmarks you can incorporate material from several documents with different formats based on different templates.</p> <p>By adding tables of contents, figure tables, sections, and headers and footers, you can manage a complex document with relative ease; simply update the Enterprise Architect reports then refresh the links in MS Word.</p>	<p><a href="#">Using MS Word</a>  <sup>2729</sup></p> <p><a href="#">Document Bookmarks</a>  <sup>2730</sup></p>

### Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Generate Documents** permission to generate document reports
- You can have several reports open at the same time, as separate **tabs** in the central view area of the Enterprise Architect work area; you can also close the reports individually or all together, leaving views of other types (such as diagrams or code editors) still open

### Learn more

- [List of Available Permissions](#)  <sup>329</sup>
- [Diagram Tabs](#)  <sup>790</sup>

### Learning Center topics

- **Alt+F1 | Enterprise Architect | Reporting | RTF Templates**
- **Alt+F1 | Enterprise Architect | Reporting | Virtual Documents**

## 19.1.1 Generate Document Reports

When you want to generate a document report on part of your model, you have a number of options for selecting the **object** to document, and for defining how your document is to be generated.

A document report describes an **element** or, more usually, a **Package** in the model; therefore, your first step is to select the Package or element to generate the document for, from the:

- Project Browser
- Diagram
- Package Browser
- Specification Manager
- Diagram List or
- Model Search

For a Package, you can also define how any **diagrams** in the Package are to be set out in the document; you do this in the **Diagram Properties dialog** for each diagram. When you have prepared and selected your Package, you open the Generate Documentation dialog and configure the parameters for generating the document.

A tutorial for quickly creating a basic document from the Example model is provided in the second of these two tables. Repeat the tutorial with different selections in the drop-down fields, to see what the system can generate for you automatically.

**Access**    **Project Browser package:** ( F8 ) or  
**Diagram element:** ( F8 )

### Generating reports

Option	Detail	See also
<b>Alternative ways to open the Generate Documentation dialog</b>	<p>Use one of the following methods:</p> <ul style="list-style-type: none"> <li>• Click on a Package, then select the <b>Project   Documentation   Generate Documentation (RTF/PDF)</b> menu option</li> <li>• Right-click on the Package and select the <b>Documentation   Rich Text Format (RTF) Report</b> context menu option</li> <li>• In a diagram, click on a specific element and select the <b>Element   Advanced   Rich Text Format (RTF) Report</b> menu option</li> <li>• In the Specification Manager, click on the Reporting icon drop-down arrow and select the <b>Generate Documentation (RTF/PDF)</b> menu option</li> <li>• In the Diagram List, Package Browser or Model Search, select one or more items, right-click and, from the context menu, select either the <b>Documentation   RTF report for</b></li> </ul>	<a href="#">Generate Documentation</a> <small>2644</small>

Option	Detail	See also
	<b>each selected Item</b> option or the <b>Documentation   RTF report on selected Items</b>	
<b>Generate report on elements from Package Browser or Model Search</b>	<p>When you select to create a report from the Diagram List, Package Browser or Model Search tools, you can generate an element-level report rather than a Package-level report, and you have additional flexibility in selecting:</p> <ul style="list-style-type: none"> <li>• The type of element to report on</li> <li>• The specific elements to report on, together or separately, whether in the same Package or not</li> </ul> <p>For example, you might want to find all elements with test cases, with the intention of reporting on some or possibly all such elements; with the Diagram List or Package Browser, you would identify these elements yourself within the list of all elements in a selected diagram or package, but with the Model Search you can automatically pick out the elements across a section of the model or across the whole model, as required.</p> <p>The search filtering could be for specific test cases; however, the results are by element so if there are test cases outside the range in any element that has a filtered test, these elements are listed as well.</p> <p>Having generated the list of elements, you can select individual elements, blocks of elements, or all elements from it and then use the context menu to generate one report on all of the selected elements, or a separate report on each element.</p>	<a href="#">Diagram List</a> <sup>[684]</sup> <a href="#">Package Browser</a> <sup>[673]</sup> <a href="#">Model Search</a> <sup>[700]</sup>

#### Generate and view an example report immediately

Step	Action	See also
1	Open the <i>EAExample</i> project.	
2	In the Project Browser, expand the <i>Testing</i> View and right-click on the <b>Testing</b> Package.	
3	Select the <b>Documentation   Rich Text Format (RTF) Report</b> context menu option. The Generate Documentation dialog displays.	
4	In the <b>Output to file</b> field, click on the ( ... ) button and browse for a convenient file location in which to hold the generated report.	
5	In the <b>Template</b> field, click on the drop-down arrow and select <b>Basic Report</b> .	

6	In the <b>Table of Contents</b> field, click on the drop-down arrow and select <b>TOC (portrait)</b> .	
7	In the <b>Stylesheet</b> field, click on the drop-down arrow and select <b>Styles - Print Medium</b> .	
8	In the <b>Cover Page</b> field, click on the drop-down arrow and select <b>Cover Page (portrait)</b> .	
9	In the <b>Output Format</b> field, click on the drop-down arrow and select <b>Portable Document Format (PDF)</b> .	
10	Click on the <b>Generate</b> button.	
11	When the report has been generated, click on the <b>View</b> button and examine the report.	

#### Notes

- Reports can be configured to include all Packages within a parent Package, or just the top level

#### Learn more

- [Diagram Options](#)<sup>[2667]</sup>
- [Exclude Package from Report](#)<sup>[2761]</sup>

#### Learning Center topics

- **Alt+F1 | Enterprise Architect | Reporting | Rich Text Reports | Generate an RTF Report**







### **19.1.1.1 Generate Documentation**

It is possible to specify a wide range of parameters to control how your document report is generated and output. You can specify, for example, where the document file is written to, what format it is generated in, and what elements are included in the document. You set these options on the Generate Documentation dialog, on which the options are provided on a number of tabs.

The first, and default, tab is the Generate tab, on which you define how and where the document is generated. After you have set options on any or all of the other tabs, you return to the Generate tab and click on the **Generate** button to produce your document.

**Access** **Project Browser package: ( F8 ) > Generate** or  
**Diagram element ( F8 ) > Generate**

#### Set Report Generation Options

Option	Action	See also
<b>Model Document</b>  <b>Root Element</b> <b>Root Package</b>	<p>Check the name of the element selected from the Project Browser, diagram, Diagram List, Specification Manager or Model Search.</p> <p>If this is the specially-created Model Document element for a virtual document, the field is <b>Model Document</b>.</p> <p>Otherwise, this field identifies the selected element of the hierarchy to be reported on; that is, the <b>Root Element</b> or <b>Root Package</b>.</p>	<a href="#">Virtual Document</a>  <sup>[2669]</sup>
<b>Output to File</b>	<p>Type or select the location and filename for the generated documentation.</p> <p>Click on the ( ... ) (Browse) button to navigate to the location.</p>	
<b>Template</b>	<p>Click on the drop-down arrow and select the name of the template to use for your document report.</p> <p>You can select any template from the drop-down list, which is divided into <b>User</b> (or custom) templates and <b>System</b> templates. If you have loaded an MDG Technology containing document templates, those are also listed under the Technology name.</p> <p>You can also immediately create a new template, by selecting the <b>&lt;new template&gt;</b> option.</p> <p>The system checks all template folders for the specified template, in the order <b>User Templates &gt; Technology Templates &gt; System Templates</b>. If there are different templates available under the same name, the system uses the first found in this sequence.</p>	<a href="#">Selecting a Template</a>  <sup>[2648]</sup> <a href="#">Custom Document Templates</a>  <sup>[2681]</sup> <a href="#">System Document Templates</a>  <sup>[2648]</sup>
<b>Table of Contents</b>	<p>Click on the drop-down arrow and select the name of the Table of Contents definition to apply to your document report.</p> <p>You can select any definition from the drop-down list, whether a standard TOC format or a user-designed format.</p> <p>Use the default value <b>&lt;none&gt;</b> to not include a Table of Contents, or to use a table of Contents built in to the selected user-defined template.</p>	<a href="#">Selecting a Table of Contents</a>  <sup>[2652]</sup>
<b>Stylesheet</b>	<p>Click on the drop-down arrow and select the name of the Stylesheet to apply to your document report.</p> <p>You can select any Stylesheet from the drop-down list, whether a standard definition or a set of user-designed styles.</p> <p>Use the default value <b>&lt;none&gt;</b> to not use a Stylesheet and instead apply the styles defined in the template you have selected in the <b>Template</b> field (above).</p>	<a href="#">Selecting a Stylesheet</a>  <sup>[2653]</sup>
<b>Cover Page</b>	Click on the drop-down arrow and select the name of the cover	<a href="#">Selecting a Cover</a>

Option	Action	See also
	<p>page definition to apply to your document report.</p> <p>You can select any definition from the drop-down list, whether a standard format or a user-designed page.</p> <p>Use the default value <b>&lt;none&gt;</b> to not include a cover page, or to use a cover page built in to the selected user-defined template.</p>	<a href="#">Sheet</a> <sup>[2654]</sup>
<b>Output Format</b>	<p>Click on the drop-down arrow and select the format in which to output the report. The default format is <b>Rich Text Format (.RTF)</b>, but you can also select:</p> <ul style="list-style-type: none"> <li>• <b>Portable Document Format (.PDF)</b> or</li> <li>• <b>Microsoft Document Format (.DOCX)</b></li> </ul>	
<b>Use Language Substitutions</b>	<p>Select this checkbox to switch custom language word substitutions on.</p> <p>Deselect the checkbox to switch custom language word substitutions off.</p>	<a href="#">Word Substitution</a> <sup>[2665]</sup>
<b>View Document On Completion</b>	Select this checkbox to open the document report as soon as it has been generated.	
<b>Use Internal Viewer</b>	<p>Select this checkbox to enable the <b>View</b> button to launch the generated document in the Enterprise Architect internal viewer.</p> <p>Deselect this checkbox to enable the <b>View</b> button to launch the generated document in the MS Windows default file viewer.</p>	<a href="#">Document Options</a> <sup>[2654]</sup>
<b>Include all Diagram Elements in Report</b>	<p>Select this checkbox to include elements in the report from external packages that are referenced from a diagram, for every diagram covered by the report; the checkbox defaults to selected.</p> <p>To include the external elements in the document, also select the <i>Package.Diagram.Element</i> and <i>Package.Element</i> checkboxes in the current template.</p> <p>If external elements are to be included only for specific diagrams, deselect this checkbox and - only for the diagrams for which you want to include external elements in the report - for each diagram select the <b>Document each contained element in RTF</b> checkbox in the diagram properties.</p> <p>When both options are deselected, or when neither of the <i>Package.Diagram.Element</i> or <i>Package.Element</i> checkboxes are selected in the template, only elements in the current package are documented.</p>	<a href="#">Diagram Options</a> <sup>[2667]</sup>
<b>Use Whiteboard mode for all diagrams</b>	All diagrams which appear in the RTF Report are displayed as if Whiteboard Mode is enabled.	<a href="#">Configure Diagram Display</a> <sup>[825]</sup>

Option	Action	See also
<b>Theme</b>	<p>Click on the drop-down arrow and select a Theme to apply the color, line thickness and font style to depictions of elements in the report, if those elements are using default values.</p> <p>If an element has a tailored appearance assigned using, for example, the Default Appearance dialog, that appearance overrides the Theme settings.</p> <p>The default option is the Theme you have currently set in your Local options. You can, however, select any other Theme just for this report. The field setting reverts to your default Theme after the report is generated.</p>	<p><a href="#">Diagram Theme Options</a> <sup>[61]</sup></p> <p><a href="#">Set an Element's Default Appearance</a> <sup>[92]</sup></p>
<b>Resource Document</b>	Click on this button to save the currently-set options as a document definition.	<a href="#">Resource Documents</a> <sup>[266]</sup>
<b>Edit Template</b>	<p>Click on this button to edit the currently-selected template using the Document Template Editor.</p> <p>You can only edit <i>user-defined</i> templates, not the standard templates provided with the system. If you have selected a system template, this button is grayed out.</p>	<a href="#">Design Custom Document Templates</a> <sup>[268]</sup>
<b>Generate</b>	Click on this button to generate the document (after you have set all the options you require, on all tabs of the dialog).	
<b>View</b>	<p>Click on this button to launch the generated document in:</p> <ul style="list-style-type: none"> <li>• The MS Windows default file viewer, or</li> <li>• The Enterprise Architect internal viewer, if you have selected the <b>Use Internal Viewer</b> checkbox (above)</li> </ul>	
<b>Abort</b>	Click on this button to cancel report generation; a prompt displays to confirm cancellation of the report.	

### Notes

- For an introduction to generating document reports, see the *Document Reports* Help topic
- The second tab on the Generate Documentation dialog, **Templates**, is used to develop and maintain **custom**:
  - Templates
  - Stylesheets
  - Tables of Contents and
  - Cover Pages

### Learn more

- [Document Reports](#) <sup>[2640]</sup>
- [Custom Document Templates](#) <sup>[2681]</sup>
- [Element Filters](#) <sup>[2660]</sup>
- [Other Filters](#) <sup>[2663]</sup>
- [Project Constants](#) <sup>[2664]</sup>
- [Language Substitution \(Codepage\)](#) <sup>[2666]</sup>

### Learning Center topics

- **Alt+F1 | Enterprise Architect | Reporting | Rich Text Reports | Generate an RTF Report**

#### **19.1.1.1.1 Selecting a Template**

You use **report templates** to generate reports on your model that present specific information in a particular format, and to help you generate those reports repeatedly and consistently. You can select from a range of **system templates** on the Document Generator, to immediately generate reports from your model, or you can select **custom templates** designed by your project team members specifically for the purposes of your project.

A template principally defines the **type of content** of the report, so that if you generate a report, say, on a Package you will extract and present information on a specific set of structures such as:

- the elements and their features and connectors, or
- the elements at the top of the Package hierarchy, or
- all elements in the Package, or
- all elements in the diagrams whether they are in that Package or from other Packages

A template also presents the report information in a specific **format and style**, initially based on a set of **standard** styles. Your report designers can **redefine** the styles in a custom **template** to whatever extent they need, but it is better to select the alternative styles from a **Stylesheet** so that **you** can choose to apply that set of styles to a report or not, independent of which template you are using, as you need.

The initial range of system templates and Stylesheets give you a great deal of flexibility in quickly and easily determining the content and presentation of the reports you want to generate, and you can further increase this flexibility with additional templates and Stylesheets developed in-house, customized to the requirements of your organization.

### Learn more

- [System Document Templates](#) <sup>[2648]</sup>
- [Custom Document Templates](#) <sup>[2681]</sup>
- [Selecting a Stylesheet](#) <sup>[2653]</sup>

#### **19.1.1.1.1.1 System Document Templates**

When you **generate reports** on your model, you select one of the report **templates** to extract information on specific key aspects of your model. There are several **system templates** available to use, either to generate a report straight away or to copy and customize to suit your own requirements. Each of the system templates supports a specific aspect of modeling or model management.



You usually generate reports on **Packages** selected from the Project Browser; these can be used to produce great hierarchical reports on parts of your model, or the whole model if you wish. There are also system templates for reporting the **diagrams** in a Package, and you can use the system templates to report on **elements** returned in search results - such reports are non-hierarchical, as they extract specific information from different areas of the model.




**Access** **Tools | Document Template Designer** (for review/copying)  
**Project Browser Object: ( F8 ) > General** (to select to generate a report)

#### Available Predefined Templates








- Basic Report
- Basic + Audit Report
- Data Model Specification
- Diagram Report
- Effort, Metrics & Risk Report
- Interaction Messages Report
- Maintenance Report
- Model Document - Master Template
- Project Glossary
- Project Issues
- Project Resources Report
- Software Requirement Specification
- Software Requirements Specification (Relationships)
- Task Allocation
- Test Specification
- Use Case Scenarios Specification
- Use Case Specification

#### Basic Template Variations

Template	Description	See also
<b>Basic</b>	The most generalized template, which includes all of the commonly used Sections and fields. It is more appropriate in its use of field names to UML Classes.  This template provides table formatting for most of the <b>Elements</b> sub-groups, such as Scenarios, Constraints, Attributes and Operations.	<a href="#">Setting Sections for Reporting</a> <sup>[2688]</sup>
<b>Basic + Audit</b>	A variation of the Basic template that incorporates information from the latest Audit runs.	<a href="#">Auditing</a> <sup>[446]</sup>
<b>Data Model Specification</b>	Used for reporting on Physical database models of any of the three types - UML DDL, Information Engineering and IDEF1X.	<a href="#">Physical Data Model</a> <sup>[2335]</sup>

Template	Description	See also
	With data modeling, a Class represents a table and its Attributes represent table fields. In the Data Model template, the section and field names are selected to represent the Tables, Columns (fields) and specific Column attributes.	
<b>Interaction Messages Report</b>	Used for generating documentation on Sequence diagrams and other Interaction diagrams. It represents connectors specifically as Messages, as well as listing the message sequence details (which are not included in the Basic template).	<a href="#">Sequence Diagram</a>  <sup>1249</sup>
<b>Diagram Report</b>	Extracts information on the diagrams for a package. It is primarily used for outputting diagram graphics and titles.  A common use of this template is where diagrams linked into an external Word master document (where virtual documents are not being used) are to be refreshed. Using the bookmarks generated with the documents, you can refresh the master document diagrams.	<a href="#">Diagrams</a>  <sup>778</sup>  <a href="#">Document Bookmarks</a>  <sup>2730</sup>

#### Sub-Element Details

Template	Description	See also
<b>Maintenance Report</b>	Extracts the <i>Element   Change</i> , <i>Element   Defect</i> and <i>Element   Issue</i> details for all elements in a selected package. This is the information defined on the Maintenance window.	<a href="#">Working on Maintenance Items</a>  <sup>2623</sup>
<b>Test Specification</b>	Extracts the <i>Element   Test</i> details for all elements in a selected package, as defined on the Testing window, and the test data generated by xUnit testing and Testpoint code testing.	<a href="#">Working on Test Records</a>  <sup>2602</sup> <a href="#">Record Test Results</a>  <sup>2575</sup> <a href="#">The Testpoints Window</a>  <sup>2578</sup>
<b>Effort, Metrics and Risk Report</b>	Extracts the efforts, metrics and risks associated with the elements in a selected package. It is useful in getting details for analyzing the time/costing on projects.  This reflects the information defined on the Project Management window.	<a href="#">Use Case Estimation</a>  <sup>584</sup> <a href="#">Project Estimation Using Use Case Metrics</a> <a href="#">The Project Management Window</a>  <sup>510</sup>
<b>Task Allocation</b>	Reports the assignment of tasks to people (Authors and Resources).	<a href="#">Resource Allocation</a>  <sup>512</sup>

Template	Description	See also
	The template extracts the details defined in the Resource Allocation tab of the Project Management window, and any Gantt chart assignment of resources to a package or elements. It principally reports the <i>Element / Resource</i> section.	







### Project Specific

Template	Description	See also
<b>Project Glossary</b>	Extracts the Glossary terms and their types, defined in the Project Glossary.  This is commonly used in Virtual documents for appending the Glossary to a larger document.	<a href="#">Project Glossary</a> <sup>[533]</sup> <a href="#">Virtual Documents</a> <sup>[2665]</sup>
<b>Project Issues</b>	Extracts information on the set of issues that are logged globally in the repository.  To report on the <b>non-global</b> , element-specific issues, see the <i>Element / Issue</i> section in the Basic template.	<a href="#">Project Issues</a> <sup>[528]</sup>

### Virtual Document

Template	Description	See also
<b>Model Document Master</b>	Model Document templates are used when creating 'virtual documents'. The Master template provides the common features for a series of documents generated from other templates.  Virtual documents do not have to have a Master Document – a single model document template can provide the basis for a generated document. This allows Package details (such as <i>Package / Package Element / Tagged Value</i> ) to be defined in Model Documents; these sections are not available in Master Documents.	<a href="#">Create Master Document</a> <sup>[2672]</sup> <a href="#">Create Model Document</a> <sup>[2673]</sup>
<b>Model Document Basic</b>		
<b>Model Document Data Model</b>	If you do use a Master template, set any Table of Contents you require in the first <b>Model Document</b> template, to ensure that a Header text is available to create a TOC. A Master document often does not have Header text and hence a TOC cannot be created.	

### Use Case/Requirements

Template	Description	See also
<b>Software Requirements Specification</b>	<p>Extracts information on external requirement elements associated with other elements (often Use Cases). The specific report section <i>Element   External Requirements</i> is selected in the template.</p> <p>To extract information on each element's <b>internal</b> requirements (responsibilities), you would select the <i>Element   Requirements</i> section, as in the Basic template.</p>	<a href="#">Requirements</a>  <a href="#">Dependency Report</a>  <a href="#">Implementation Report</a>  <a href="#">Internal Requirements</a> 
<b>Use Case Specification</b>	Extracts basic element description information on the Use Cases in a selected package. This excludes scenario details and child structures (to filter out the information on any Sequence diagrams set as child diagrams of the Use Case elements).	<a href="#">Use Case</a> 
<b>Use Case Scenarios Specification</b>	<p>Extracts information on the Use Cases in a selected package, <b>including</b> the details of any scenarios. This template also excludes child structures.</p> <p>If you want to include sub-element Use Cases, select the <i>Element   Child Elements</i> section.</p>	<a href="#">Scenarios</a> 

### Notes

- The facility to use the *Diagram | Element* section to report elements contained in a diagram but **defined in another Package**, is inactive in system templates

### Learn more

- [Document Templates](#) 
- [Design Custom Document Templates](#) 

#### 19.1.1.1.2 Selecting a Table of Contents

A very useful feature of any documentation is the inclusion of a **Table of Contents**. This can, however, be quite demanding to create, especially when you have to set up a Table of Contents for individual documents. Even setting up Contents Lists within templates can still be very time consuming, when you have a number of templates to maintain. It is far quicker and easier to apply a specific **Table of Contents template** to your document reports independent of which report you are generating and which content template you are basing the report on.

Enterprise Architect provides two **TOC** templates - one for landscape documents and one for portrait documents - that have the appropriate styles, formatting and numbering system already defined. You can access these templates through the **Table of Contents** field on the General tab of the Generate Documentation dialog.

Your report developers can copy the two system-provided templates as the bases for **custom** Tables of Contents, tailored to the requirements of your organization. These customized templates are also directly available to you through the drop-down list for the **Table of Contents** field.

### Notes

- If you have a Table of Contents defined in any of your custom report templates, and you want to use it, just set the **Table of Contents** field on the General tab to **<none>**

### Learn more

- [Generate Documentation](#) <sup>[2644]</sup>
- [Custom Document Templates](#) <sup>[2681]</sup>
- [Notes on Creating Tables of Contents](#) <sup>[2706]</sup>

#### 19.1.1.1.3 Selecting a Stylesheet

Using **Stylesheets** is an effective method of applying consistent format and appearance standards to categories of documentation, independent of the templates or other methods of defining document **content** you use. Stylesheets increase the flexibility of your documentation generation, in that you can:

- Provide a suite of documents generated from different templates with the same style standards, using one Stylesheet, **and**
- Deliver the same document content in different styles for different customer audiences by applying different Stylesheets to documents generated from one template

Enterprise Architect provides six **standard** Stylesheets - three for printed documentation and three for online documentation. Your report designers can copy these as the bases for **custom** Stylesheets, reflecting the corporate standards of your organization. Each custom Stylesheet can define a totally different page layout and document styles to other Stylesheets and to the Normal.rtf Styleheet. You can access both system and custom Stylesheets from the **Stylesheet** drop-down list on the General tab of the Generate Documentation dialog.

### Style Hierarchy



Stylesheets are the top level of a hierarchy of styles across the system.

- Initially, all documents and templates, whether for linked documents or report documents, reflect the global styles defined in the **Normal.rtf** file; you can edit this file to provide new global styles for all documents and templates created **after** the update
- In a specific document or template, you can edit the styles for that document or for all documents generated from that template; these edited styles override those of the same name in the Normal.rtf file, but any styles that have **not** been changed still reflect the definitions in the Normal.rtf file
- You can use a Stylesheet to apply a set of style definitions to a generated document, independent of the template you use; these styles override any styles of the same name defined in the document templates and/or the Normal.rtf file

### Notes

- If you have custom templates that contain all the style definitions you need, or if you want to just use the Normal.rtf styles, you can continue to apply those simply by setting the **Stylesheet** field on the General tab to **<none>**

### Learn more

- [Generate Documentation](#)  <sup>[2644]</sup>
- [Custom Template Design Options](#)  <sup>[2704]</sup>
- [Notes on Creating Stylesheets](#)  <sup>[2705]</sup>
- [The Normal.rtf Style Template](#)  <sup>[1058]</sup>

#### 19.1.1.1.4 Selecting a Cover Page

A **cover page** defines the text, styles, graphics and layout of the front page of a document. Enterprise Architect provides two **standard** cover pages, one for portrait documents and one for landscape documents, which you can apply to your document reports independent of the report template and Stylesheet you are using. Your report designers can copy these standard cover pages as the bases for **custom** cover pages that reflect your corporate styles, logos and document naming conventions.

The cover page templates can contain fields that automatically extract and display data such as:

- The report type
- The Package or element being reported on
- The version of the report
- The status of the report
- The date on which the report was generated
- The name of the person who generated the report
- The location of the report file
- Any logos or graphics to be displayed on the front page

In using a cover page, you do not need to spend time defining the inclusion and display of the report title and provenance - it is already prepared for you. You can access the cover page templates from the **Cover Page** drop-down list on the Generate tab of the Generate Documentation dialog.

#### Notes

- If you have custom report templates that already contain the cover page definitions you want, you can still apply them by just setting the **Cover Page** field on the Generate tab to **<none>**

#### Learn more

- [Generate Documentation](#)  <sup>[2644]</sup>
- [Design Custom Document Templates](#)  <sup>[2684]</sup>
- [Notes on Creating Cover Pages](#)  <sup>[2707]</sup>

#### 19.1.1.2 Document Options

When defining the contents of your document, you can set a wide range of options for selecting the reported items by type and property, and you can organize the information within each type in order of position in the Project Browser or by created or modified date. These **Document** options are provided in two different ways:

- As the Options tab on the Generate Documentation dialog, to be set for a **specific instance** of a report; the options are **reset** when you exit the dialog or select a different template
- As panels on the Document Options dialog, to be configured in a template as the **default settings** to be applied to **every** report generated using that template; these settings can be overridden by the 'specific instance' settings on the Generate Documentation dialog

The listed options are the same for both dialogs, except that the **Document Options** dialog:

- Includes the Exclude Filters, Element Filters and Other Filters tabs from the Generate Documentation dialog
- Includes the Custom Query tab (for Template Fragments)
- Does not provide these options (as found on the Options tab of the Generate Documentation dialog):
  - The **Switch generator** button
  - The **Disable large OLE file support** option
  - The **Insert page breaks when generating a Master Document** option

**Access** **Project Browser package:** ( F8 ) > Options, or

**Tools | Document Template Designer:** <template> > Content | File | Document Options

### Options

Field	Action	See also
<b>Packages by</b>	Click on the drop down arrow in the first field and select to organize Packages in the generated documentation by <b>Name</b> , <b>Tree Order</b> , <b>Modified Date</b> or <b>Created Date</b> .  Click on the drop down arrow in the second field and select to list the Packages in either <b>Ascending</b> or <b>Descending</b> order.	
<b>Elements by</b>	Click on the drop down arrow in the first field and select to organize elements in the generated documentation by <b>Name</b> , <b>Tree Order</b> , <b>Modified Date</b> or <b>Created Date</b> .  Click on the drop down arrow in the second field and select to list the elements in either <b>Ascending</b> or <b>Descending</b> order.	
<b>Diagrams by</b>	Click on the drop down arrow in the first field and select to organize diagrams in the generated documentation by <b>Name</b> , <b>Tree Order</b> , <b>Modified Date</b> or <b>Created Date</b> .  Click on the drop down arrow in the second field and select to list the diagrams in either <b>Ascending</b> or <b>Descending</b> order.	
<b>Hide 'note-less' elements</b>	Select this checkbox to exclude all elements without notes from the documentation.	
<b>Skip root package</b>	Select this checkbox to exclude the parent Package from the documentation and include only the child Packages.	
<b>No bookmarks</b>	Select this checkbox to prevent document bookmarks from being inserted into the generated document.	

Field	Action	See also
<b>Hide &lt;Anonymous&gt; elements</b>	Select this checkbox to hide anonymous elements in the documentation.	
<b>Use style defined in template for notes</b>	Select this checkbox to override the character formatting specified in your <b>Notes</b> fields with the formatting set for the field in the template.	
<b>Disable large OLE file support</b>	Select this checkbox to disable support for large Object Linking and Embedding (OLE) files.	
<b>Insert page breaks when generating a Master Document</b>	Select this checkbox to insert a page break after each Model Document in a Master Document.	<a href="#">Virtual Documents</a> <small>[2669]</small>
<b>Include child elements even if the parent element is filtered out</b>	Select this checkbox to document all child elements that are not otherwise filtered out by the restrictions, even if the parent elements are filtered out.	
<b>Hide 'note-less' connectors</b>	Select this checkbox to exclude all connectors without notes from the documentation.	
<b>Diagram Format</b>	Click on the drop-down arrow and select the diagram format for the images included within the documentation; either <b>Metafile</b> or <b>Bitmap</b> .	
<b>Adjust Heading Levels</b>	<p>Click on the drop-down arrow and select the lowest heading level that can be generated for nested sub-packages in the document</p> <p>The Document Generator reproduces heading levels down to the value you set; for example, if you have four nested levels of sub-packages and you set this field to:</p> <ul style="list-style-type: none"> <li>• <b>Heading 2</b>, all sub-packages in the report are documented under level 2 headings</li> <li>• <b>Heading 4</b>, the first level of subpackages are documented under level 2 headings, the next level under level 3 headings, and the remainder all under level 4 headings</li> <li>• <b>Heading 6</b>, the first level of subpackages are documented under level 2 headings, the next level under level 3 headings, the next under level 4 headings, and the next under level 5 headings; if you added further levels of sub-package they would all be documented under level 6 headings</li> </ul> <p>The field defaults to <b>Heading 9</b> to accommodate the maximum number of levels of nested subpackages.</p>	



Field	Action	See also
<b>Switch generator</b>	<p>Click on this button to switch from this Generate Documentation dialog (the Enhanced Template Driven Generator) to the Rich Text Format Report dialog (Legacy Generator).</p> <p>This button is not available if you displayed the dialog from the Diagram List, Package Browser or Model Search.</p>	<a href="#">The Legacy Report Generator</a> <small>[2720]</small>
<b>Only include objects</b>	<p>Filter elements according to the date they were created or last modified. In the:</p> <ul style="list-style-type: none"> <li>First field, click on the drop-down arrow and select the value <b>Modified</b> or <b>Created</b></li> <li>Second field, click on the drop-down arrow and select the value <b>After</b> or <b>Before</b></li> <li>Third field, select the checkbox, click on the drop-down arrow and select the appropriate date</li> </ul>	
<b>Where Package Phase</b>	<p>Filter elements according to the value of the <b>Package Phase</b> field.</p> <p>In the first field click on the drop-down arrow and select the qualifier, and in the second field type the required phase (or leave the default value <b>All</b>).</p>	
<b>With element status</b>	<p>Filter elements according to status.</p> <p>In the first field, click on the drop-down arrow and select the qualifier ( <b>like</b>, <b>not like</b>, <b>in</b>, <b>not in</b>) and in the second field type the value to be used.</p> <p>Values should be enclosed in quotes; for example: "<b>Proposed</b>".</p> <p>If you type more than one value, separate them with a comma; for example:</p> <p><b>"Proposed", "Implemented"</b></p>	
<b>Connector Direction</b>	<p>Click on the drop-down arrow and select the connector direction to include only connectors having that direction.</p> <p>If you select <b>Both</b>, the connector is documented twice; once for the source element and once for the target.</p> <p>For the remaining two values, the connector is documented only for the source or target element, as appropriate.</p>	
<b>Except where Query excludes Package</b>	<p>This option excludes the selected Package or a child Package from the reporting filter conditions (above) where that Package is specifically identified by a Custom SQL Query or a Custom Script.</p> <p>Click on the appropriate radio button for the type of query and:</p> <ul style="list-style-type: none"> <li>If you have selected <b>Custom SQL</b>, type the query into the window beneath the radio buttons</li> <li>If you have selected <b>Custom Script</b>, a new field displays; click on</li> </ul>	<a href="#">Exclude Package Query and Script</a> <small>[2658]</small>

Field	Action	See also
	the drop-down arrow and select the script, then type the entry point of the Script into the window beneath the radio buttons	

#### Learn more

- [Generate Documentation](#) <sup>[2644]</sup>
- [Design Custom Document Templates](#) <sup>[2684]</sup>
- [File Control](#) <sup>[1050]</sup>
- [Exclude Filters](#) <sup>[2660]</sup>
- [Element Filters](#) <sup>[2660]</sup>
- [Other Filters](#) <sup>[2663]</sup>
- [Custom Query Fragments](#) <sup>[2717]</sup>

#### Learning Center topics

- **Alt+F1 | Enterprise Architect | Reporting | RTF Templates | Filter Elements**
- **Alt+F1 | Enterprise Architect | Reporting | RTF Templates | Filter Connectors**

#### **19.1.1.2.1 Exclude Package Query and Script**

On the Document Options dialog (Template Editor) or Options tab (Report Generator) you can enter a **custom SQL Query** or select a **custom Script** to tailor the report in a specific way. One possibility is to **exclude** the selected **Package** or a child Package from the reporting filter conditions. You can base your Query or script on the examples provided here.

The two parameters you use in excluding a Package are:

- **#PACKAGEID#** - the **Package\_ID** of the current record in **t\_package** being processed on the exclusion condition
- **#OBJECTID#** - the **Object\_ID** of the current Package element record in **t\_object** being processed on the exclusion condition

#### Custom SQL Query

To exclude the Package from the filter conditions using a **Custom SQL Query**, you can create the Query based on the **t\_package** columns. For example:

```
SELECT Package_ID AS ExcludePackage FROM t_package WHERE Package_ID =
#PACKAGEID# and Name = 'Test'
```

Alternatively, you can create the Query based on the package object columns in the **t\_object** table:

```
SELECT t_package.Package_ID AS ExcludePackage FROM t_package, t_object
WHERE
t_package.Package_ID = #PACKAGEID#
AND
```

```
t_object.Object_ID =#OBJECTID#

AND

t_object.Stereotype = ' NoDoc'
```

### Custom Script

If you have selected the **Custom Script** option and want to exclude the Package from the filter conditions, you can create a script and enter a call to it, such as:

```
ExcludePackage( #PACKAGEID#)
```

This is a sample of the **XML returned** by the script:

```
<?xml version="1.0"?>
  <EADATA version="1.0" exporter="Enterprise Architect">
    <Dataset_0>
      <Data>
        <Row>
          <ExcludePackage>
            89
          </ExcludePackage>
        </Row>
      </Data>
    </Dataset_0>
  </EADATA>
```

This is an example of **JScript** for excluding the Package:

```
!INC Local Scripts.EAConstants-JScript
/*
 * Script Name: RTF Exclude Packages Script Sample
 */

function ExcludePackage(packagelD)
{
  var xml DOM = new ActiveXObject( "MSXML2.DOMDocument.4.0" );
  xml DOM.validateOnParse = false;
  xml DOM.async = false;

  var node = xml DOM.createProcessingInstruction( "xml", "version='1.0'
encoding='ISO-8859-1'" );
  xml DOM.appendChild( node );

  var xml Root = xml DOM.createElement( "EADATA" );
  xml DOM.appendChild( xml Root );

  var xml DataSet = xml DOM.createElement( "Dataset_0" );
  xml Root.appendChild( xml DataSet );

  var xml Data = xml DOM.createElement( "Data" );
  xml DataSet.appendChild( xml Data );

  var xml Row = xml DOM.createElement( "Row" );
  xml Data.appendChild( xml Row );
```

```

var package as EA.Package;
package = Repository.GetPackageById( packageId )

if ( package.StereotypeEx == "NoDoc" )
{
    var xmlName = xmlDOM.createElement( "ExcludePackage" );

    xmlName.text = "" + package.PackageId;
    xmlRow.appendChild( xmlName );
}

return xmlDOM.xml;
};

```

Learn more

- [Document Options](#) 

**19.1.1.3 Exclude Filters**

As you specify the output of your report, you might want to **exclude** objects of a specific **type**. You can do this using the **Exclude Filters** tab, through which you can select to exclude from your report one or more:

- Types of **element**
- Types of **connector**, and/or
- Types of **diagram**

You can select the objects in one of two ways:

- Click on the **All** button underneath the appropriate panel, to **exclude** every type of element, connector or diagram, and then click on the individual types that you **DO** want to include in the report, or
- Click on the **None** button underneath the appropriate panel to clear all selections (which includes the objects of every type in the report) and then click on each type of object that you do **NOT** want to show in the report

Notes

- If you want to specifically include or exclude elements on the basis of their specific **properties** or **features**, rather than their type, use the Element Filters and Other Filters tabs

Learn more

- [Generate Documentation](#) 
- [Element Filters](#) 
- [Other Filters](#) 

**19.1.1.4 Element Filters**

As you specify the output of your report, you might want to include only elements that have specific properties, rather than every element encountered. Using the Element Filters tab, you can define which element properties to report on, and which fields and field values. You can also delete the element filters if you no longer want to apply them.

The Element Filters tab is provided in two different ways, as a tab of:

- The Generate Documentation dialog, to set filters for a **specific instance** of a report; the filters are **reset** when you exit the dialog or select a different template
- The Document Options dialog, to set filters in a template as the **default settings** to be applied to **every** report generated using that template; these settings can be overridden by the 'specific instance' settings on the Generate Documentation dialog


The options and behavior are the same for both dialogs, except that the **Document Options** dialog provides a facility for importing a previously-defined set of Model Search filters from within your model or from an external file (created by exporting a Model Search from another model).

The facilities are also provided through the Element Filters page of the Properties dialog for a Standard Chart element. This page most closely resembles the tab of the Document Options dialog in the Document Template editor.

**Access** **Project Browser package: ( F8 ) > Element Filters, or**  
**Tools | Document Template Designer: <template> > Content | File | Document Options >**  
**Element Filters**

#### Review Element Filters

Column/Button	Action	See also
<b>Search In</b>	Shows the type and name of each element property field to search on. Select the checkbox against a property field to include it in the current search.	
<b>Condition</b>	Displays the condition of the search parameter. The possible values are <b>Contains</b> , <b>Equal To</b> , <b>Not Equals</b> and <b>One Of</b> .	
<b>Look For</b>	Displays the search term to perform the conditional search on. This value is specific to the selected field; for example, the value could be a date for <i>DateCreated</i> or a text or numerical value for other fields. The search term can contain multiple values, separated by commas.	
<b>Required</b>	Indicates that the search results must include elements with the search term in that field; these checkboxes are set on the <b>Add Filters dialog</b> . When no <b>Required</b> checkboxes are selected, the fields listed as filters have an OR relationship; that is, if the search term is found in any <b>one</b> of those fields, then the element is displayed. Any field having the <b>Required</b> checkbox ticked overrides fields where the <b>Required</b> checkbox is not ticked.	
<b>Element Features</b>	Specify whether element features are optional or required; these	

Column/Button	Action	See also
<b>- Optional</b> <b>- Required</b>	<p>appear as a new branch underneath the root element term in the <b>Search In</b> column.</p> <p>If you scroll down the <b>Search In</b> column, you might see sub-branches such as <i>Attribute</i>, <i>Change</i> and <i>Custom Property</i>; these are the element features. You can add these features using the Add Filters dialog.</p> <p><b>Select the:</b></p> <ul style="list-style-type: none"> <li><b>Optional</b> radio button to generate a list of elements that have <b>one</b> of the element features (<b>Element Type = Object</b>), or <b>one</b> of the feature filters (<b>Attribute Name = Att1</b>); for example, if your search is:   <b>Element Name = Class11, Attribute Name = m_Att1 or Scope = Public</b>   the search results list all the elements that have the name of <b>Class11</b> <i>and</i> all the elements that have an <b>Attribute Name</b> of <b>m_Att1</b> <i>or</i> a <b>Scope</b> of <b>Public</b>.</li> <li><b>Required</b> radio button to generate a list of elements that must have the element features; for example, if your search is:   <b>Element Name = Class, Attribute Name = m_Att1 or Scope = Public</b>   you would get elements that must have the name of <b>Class</b> <i>AND</i> have an Attribute with a name of <b>m_att1</b> <i>OR</i> a <b>Scope</b> of <b>Public</b>.</li> </ul>	<a href="#">Add Filters</a> 
<b>Import Search</b>	<p>Click on this button and select to import the filters defined within a search, from either:</p> <ul style="list-style-type: none"> <li>An external XML file exported from another project, located on a Browser dialog, or</li> <li>A list of the standard and custom Model Searches within your installation of Enterprise Architect</li> </ul>	
<b>Add Filter</b>	Click on this button to display the Add Filters dialog, to add a new set of parameters to filter the search on.	
<b>Edit Filter</b>	Click on this button to display the Edit Filters dialog, to change the search parameters.	
<b>Remove Filter</b>	Click on this button to immediately remove the selected filter from the search.	

#### Notes

- Check that the values you provide for the **Look for** fields are exactly what is contained in the element

data fields to be searched; for example, the Test Classes Unit, Integration, System, Acceptance and Scenario are represented by attribute values **1 - 5**, not by their respective tab names

#### Learn more

- [Model Search](#) <sup>[700]</sup>
- [Generate Documentation](#) <sup>[2643]</sup>
- [File Control](#) <sup>[1050]</sup>
- [Design Custom Document Templates](#) <sup>[2684]</sup>

#### Learning Center topics

- **Alt+F1 | Enterprise Architect | Reporting | RTF Templates | Filter Elements**

### 19.1.1.5 Other Filters

As you specify the output of your report, you might want to include only specific **features** of elements (sub-element components such as attributes, responsibilities or constraints). If the feature does not have the defined characteristics, it is not reported for the element. Using the **Other Filters** tab, you can define which features to report on, and which fields and field values. You can also **delete** the filters if you no longer want to apply them.

The Other Filters tab is provided in two different ways, as a tab of:

- The Generate Documentation dialog, to set filters for a **specific instance** of a report; the filters are **reset** when you exit the dialog or select a different template
- The Document Options dialog, to set filters in a template as the **default settings** to be applied to **every** report generated using that template; these settings can be overridden by the 'specific instance' settings on the Generate Documentation dialog

The options and behavior are the same for both dialogs.

**Access** **Project Browser package: ( F8 ) > Other Filters, or Tools | Document Template Designer: <template> > Content | File | Document Options > Other Filters**

#### Manage Other Filters

Column/Button	Action	See also
<b>Search In</b>	Shows the type and name of each element feature field to search on. Select the checkbox against a field to include it in the current search.	
<b>Condition</b>	Displays the condition of the search parameter. The possible values are <b>Contains</b> , <b>Equal To</b> , <b>Not Equals</b> and <b>One Of</b> .	
<b>Look For</b>	Displays the search term to perform the conditional search on. This value is specific to the selected field; for example, the value	

Column/Button	Action	See also
	could be a date for <i>DateCreated</i> or a text value for other fields. The search term can contain multiple values, separated by commas.	
<b>Required</b>	Indicates that the search results must include element features with the search term in that field; these checkboxes are set on the <b>Add Filters</b> dialog.  When no <b>Required</b> checkboxes are selected, the fields listed as filters have an OR relationship; that is, if the search term is found in any <b>one</b> of those fields, then the element is displayed.  Any field having the <b>Required</b> checkbox ticked overrides fields where the <b>Required</b> checkbox is not ticked.	<a href="#">Add Filters</a> <sup>[716]</sup>
<b>Add Filter</b>	Click on this button to display the Add Filters dialog, to add a new set of parameters to filter the search on.	
<b>Edit Filter</b>	Click on this button to display the Edit Filters dialog, to change the search parameters.	
<b>Remove Filter</b>	Click on this button to remove the selected filter from the search.	

#### Learn more

- [Generate Documentation](#) <sup>[2644]</sup>
- [Design Custom Document Templates](#) <sup>[2684]</sup>
- [File Control](#) <sup>[1050]</sup>

#### 19.1.1.6 Project Constants

As you develop your report, you can incorporate flags that represent project-wide terms and values. The flags - or **Project Constants** - are replaced with the actual text during document compilation. Using this facility, you can use a particular text string or number extensively throughout your project documentation, whilst maintaining the value in a single location. You define Project Constants on the Generate Documentation dialog; they are then available to be inserted into **all** custom document report templates in the text, section headers, end-notes, page headers or page footers, using the Document Template Designer context menu.

**Access** **Project Browser package: ( F8 ) > Project Constants or Diagram element ( F8 ) > Project Constants**

#### Manage Project Constants

Field/ Button	Action	See also
<b>Add</b>	Click on this button to add a new Project Constant to the list.	



Field/ Button	Action	See also
	A prompt displays for the name and value; type these in and click on the <b>OK</b> button.	
<b>Delete</b>	Click on this button to remove the selected Project Constant from the list. A prompt displays to confirm the deletion; click on the <b>OK</b> button to complete the task..	
<b>Import</b>	Click on this button to import a set of fields from an external XML file. A browser displays to select the source file and location.	
<b>Export</b>	Click on this button to export all fields to an external XML file. A browser displays to specify the target file and location.	

#### Learn more

- [Styles, Special Text & Table of Contents](#) 

### 19.1.1.7 Word Substitution

In generating document reports, you will call in many English technical terms that are used across the system, in particular field names. It is possible to replace these field names in your generated documents with other terms that might be more meaningful to the readers, such as a translation of each term into your own language.

**Access** **Project Browser package: ( F8 ) > Word Substitution or  
Diagram element ( F8 ) > Word Substitution**

#### Add a translation for a term

Step	Action	See also
<b>1</b>	Double-click on the term in the <b>English</b> column; the <b>Enter Value</b> field displays.	
<b>2</b>	Type the substitute text in the <b>Enter Value</b> field, and click on the <b>OK</b> button. The text displays in the <b>Substitute</b> column, against the English term. This substitute will display in your generated reports in place of the English term.	

#### Note

- To put the word substitutions into effect in a generated report, select the **Language Substitution**

checkbox on the Generate tab

### 19.1.1.8 Language Substitution

If you want to generate and export document reports from Enterprise Architect in languages or dialects other than US English, you can customize the codepage, default language ID and character set that the system uses when generating the documentation. This makes it much easier to generate documentation appropriate to your country or location.

**Access** **Project Browser | Click on Package: F8 > Codepage or**  
**Diagram element: F8 > Codepage**

#### Set Reporting Language

Field/Button	Action	See also
<b>Language</b>	Click on the drop-down arrow and select the language code appropriate to the language you want to generate reports in.	
<b>Codepage</b>	Click on the drop-down arrow and select the codepage appropriate to the language you want to generate reports in.	
<b>Charset</b>	Click on the drop-down arrow and select the character set appropriate to the language you want to generate reports in.	
<b>English Tag</b>	Lists the default US English language code tags within Enterprise Architect.	
<b>Substitute Tag</b>	<p>(For advanced use only.) If you want to apply a different language code tag in place of a US English tag, double-click on the item line. A prompt displays for the substitute tag code; type in the code and click on the <b>OK</b> button.</p> <p>To clear a substitute tag, double-click on the item and delete the code value.</p> <p>When you generate your document reports, the substitute tags are used in the output.</p>	

#### Notes

- If you want to generate reports in a language that requires Unicode support, please note that Enterprise Architect **.EAP** files default to using **JET 3.5** as the database engine, which does not support Unicode character sets (see the *Getting Started > Basics* topic) ; to provide Unicode support in your project, either upsize to a **DBMS repository** or set **JET 4.0** as the database engine, and download a copy of the **Jet 4.0 EABase** model from the Sparx Systems website and do an **EAP to EAP transfer** of your model into the Jet 4.0 file
- The codes that you select remain set for all reports you generate, until such time as you specifically change them again

- You can transport these language tag definitions between models, using the **Export Reference Data** and **Import Reference Data** options

#### Learn more

- [Basics](#) <sup>[43]</sup>
- [Server Based Repositories](#) <sup>[214]</sup>
- [Perform a Project Data Transfer](#) <sup>[504]</sup>
- [Export Reference Data](#) <sup>[376]</sup>
- [Import Reference Data](#) <sup>[380]</sup>

### 19.1.1.9 Diagram Options

When generating reports on a Package that includes diagrams, you can set options in each diagram's Properties dialog to define whether that diagram is to be included in the report and, if so, how it is to be represented.

**Access** **Diagram background context menu | Properties > Diagram**

#### Document Options

Option	Action	See also
<b>Exclude Image from RTF Documents</b>	Select this checkbox to exclude the image of the this diagram from any document reports.	
<b>Document Each Contained Element in RTF</b>	<p>Select this checkbox to include, in the report, details of any elements that belong in other (external) Packages but that form part of this diagram within the Package being reported on; the report includes external elements in each diagram on which you have selected the checkbox.</p> <p>On the Generate Documentation dialog, the <b>Include all Diagram Elements in Report</b> checkbox defaults to selected to include external elements for <i>every</i> diagram covered by the report; therefore, to include external elements only for <b>specific</b> diagrams, <b>deselect</b> that checkbox.</p> <p>In either case, you must enable the <i>Package.Diagram.Element</i> and <i>Package.Element</i> sections in your (customized) document generation templates in order to include this external element information in the report.</p>	<p><a href="#">Generate Documentation</a> <sup>[2644]</sup></p> <p><a href="#">Report Elements From External Packages</a> <sup>[2695]</sup></p>
<b>Divide Diagram Into Multiple Pages</b>	<p>If the diagram is large, select this checkbox to divide the diagram into separate pages in the document report.</p> <p>This option is only effective when the <b>Scaled Printing option</b> on the Print Advanced dialog is set to <b>None</b>.</p>	<a href="#">Scaled Printing Option</a> <sup>[871]</sup>
<b>Rotate Images</b>	Select this checkbox to rotate each diagram image by 90 degrees in	

Option	Action	See also
	the document report; this is only valid for <b>bitmap</b> (.bmp) images.	

#### Learn more

- [Configure Diagram Display](#)<sup>[825]</sup>

### 19.1.1.10 Resource Documents

When you set the parameters on the Generate Documentation dialog to define an output document as you need it, you might want to re-use that configuration to obtain a consistent view of changes to your model over time. You can capture the configuration as a **Resource Document**, which saves the:

- Package name
- Output file destination
- Template
- Output format
- Options page settings
- Exclude Filters
- Element Filters and
- Other Filters

The document is saved into the *Document Generation > Defined Documents* folder of the Resources window. Saving your often-used report configurations as Resource Documents can simplify future document generation, both in producing specific documents and in batch-generating all documents or a selection of documents.

**Access** **Project Browser package:** F8 > **Generate:** Resource Document **or**  
**Diagram element:** F8 > **Generate:** Resource Document

**Project | Resources (Alt+6) | Document Generation | Defined Documents**

#### Manage Resource Documents

Task	Action	See also
<b>Create a Resource Document</b>	On the Generate Documentation dialog, click on the <b>Resource Document</b> button and, in the <b>Enter Value</b> field, type a name for the document.  Click on the <b>OK</b> button. The document is added to the Resources window for easy future access.	
<b>Open, Generate or Delete Documents</b>	In the Resources window, expand the <i>Defined Documents</i> folder and right-click on the required <b>document</b> .  Select the required context menu option: <ul style="list-style-type: none"> <li>• <b>Open Document</b> - Open the corresponding .document file, as specified by the document template <b>Filename</b> property</li> </ul>	

Task	Action	See also
	<ul style="list-style-type: none"> <li>• <b>Generate Document</b> - Open the Generate Documentation dialog, loaded with the specified template</li> <li>• <b>Auto Generate Document</b> - Generate documentation, with the document located at the path specified by the template's <b>Filename</b> property</li> <li>• <b>Delete Document</b> - Remove the specified document</li> </ul>	
<b>Batch-generate a number of documents</b>	<p>Right-click on the <i>Defined Documents</i> <b>folder</b> name and select the <b>Generate Documents</b> context menu option; the Batch Document Generation dialog displays.</p> <p>The dialog lists all resource documents in the <i>Defined Documents</i> folder, defaulted to selected; deselect the checkbox against each document that you do not want to generate (or click on the <b>Deselect All</b> button to clear all selections, then select those you require).</p> <p>Click on the <b>OK</b> button to generate each of the remaining reports into their respective target file locations.</p> <p>Alternatively, to automatically generate every document in the <i>Defined Documents</i> folder, without displaying the Batch Document Generation dialog, click on the <b>Generate All Documents</b> context menu option.</p>	

#### Learn more

- [Generate Documentation](#) <sup>[2644]</sup>
- [Document Options](#) <sup>[2654]</sup>
- [Exclude Filters](#) <sup>[2660]</sup>
- [Element Filter](#) <sup>[2660]</sup>
- [Other Filters](#) <sup>[2663]</sup>

#### Learning Center topics

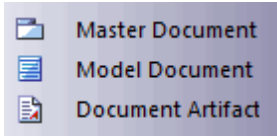

- **Alt+F1 | Enterprise Architect | Reporting | Rich Text Reports | Generate an RTF Report**

### 19.1.2 Virtual Documents

Both standard and custom report templates generate documents that present information reflecting the organization of the Project Browser. You can also generate document and web reports independent of that organization, grouping and ordering individual Packages and elements according to whatever criteria you want to apply. For this purpose, you use **virtual documents**.

You can create separate virtual documents describing, say, the Requirements, Use Cases or Design elements of a project, based on individual **Model Document** elements. Alternatively, you can make these separate sections of one report, retaining their own different formats but with common headers and footers, organized under a **Master Document** Package element. Any Table of Contents, Stylesheet and/or Cover Page you select would be applied to the combined document and could apply your corporate standards across all the Model Document sections. You can create as many virtual documents as you want, for as many combinations of information as you need.

#### Elements, Templates and Content Definitions

Construct	Detail	See also
<b>Document Elements</b>	<p>Master Document and Model Document elements are available from the Documentation page of the Diagram Toolbox; on the Toolbox, select <b>More tools   Documentation</b>.</p>  <p>When you drag the Master Document and Model Document elements onto a diagram, the following symbols display, respectively:</p>  <p>The Documentation Toolbox page also provides the <b>Document Artifact</b> element, into which you can generate a virtual document or through which you can add a linked document to an element.</p>	<p><a href="#">Diagram Toolbox</a> <sup>[792]</sup></p> <p><a href="#">Create Model Document</a> <sup>[2673]</sup></p> <p><a href="#">Create Master Document</a> <sup>[2672]</sup></p> <p><a href="#">Document Artifact</a> <sup>[1373]</sup></p> <p><a href="#">Generate Report to an Artifact Element</a> <sup>[2680]</sup></p>
<b>Templates</b>	<p>For each <b>Model Document</b> element, you can create a custom template; for example, a specifically-designed Requirements template for a Requirements document, or a Use Case template for a section on Use Cases.</p> <p>For web reports, you identify the template on the Publish as HTML dialog; for document reports, you specify the template name as the value of the <b>RTFTemplate Tagged Value</b>.</p> <p>The Master Document element has its own <b>RTFTemplate</b> Tagged Value, which defines the headers, footers and any other common content; you can import your <b>corporate standards template</b> and insert its name in the Tagged Value.</p> <p>The template in the Master Document overrides the templates in the Model Documents; for example, header and footer definitions in the Master Document template override any header and footer definitions in the Model Document templates. In this way, you can apply consistent and continuous styles and page numbering throughout the report generated through the Master Document.</p> <p>If you want the Model Documents to have their own styles, applied through their own <b>RTFTemplate</b> Tagged Values, either leave the Master Document <b>RTFTemplate</b> Tagged Value blank (for completely separate overall styles) or remove the definition of specific styles from</p>	<p><a href="#">Design Custom Document Templates</a> <sup>[2684]</sup></p> <p><a href="#">Import Template</a> <sup>[2707]</sup></p>

Construct	Detail	See also
	the Master Document template.	
<b>Content</b>	<p>The content of the report is defined as either:</p> <ul style="list-style-type: none"> <li>A list of Packages (defined as attributes) dragged onto the element in whatever order or combination is most appropriate to your requirements; you can easily <b>add</b> or <b>delete</b> Packages as necessary</li> <li>or</li> <li>(Not for web reports) a standard search (identified by Tagged Values) created within the Model Search facility - when you generate the document, this search captures the required data throughout the model and populates the document; note that diagram searches are not supported</li> </ul> <p>You can control the sequence in which information is presented in the document by, for example, changing the sequence of Packages in the Model Document or the sequence of Model Documents in the Master Document.</p>	<p><a href="#">Add Packages to Model Document</a><sup>[2675]</sup></p> <p><a href="#">Delete Package in Model Document</a><sup>[2676]</sup></p> <p><a href="#">Create and Modify Searches</a><sup>[709]</sup></p> <p><a href="#">Document Order</a><sup>[2677]</sup></p>

### Notes

- In a Model Document for a document report, you should not define both a list of Packages and a search; if both are present, when you generate the document it is created from the Package list only
- You cannot use **Bookmarking** in Master Document elements, which effectively replace Bookmarking in Word

Bookmarking requires each bookmark to be unique; when you generate a report with a standard template (including in a single Model Document element), each bookmark is unique and there is a 1:1 association between the Elements-details being generated and the elements in the repository

As Master Documents are intended to contain multiple sub-documents, the association ceases to be 1:1; there is no simple method that enables the generated data to be uniquely identified directly in association with the original element

### Learn more

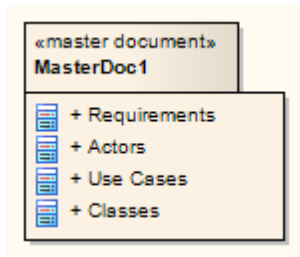
- [Model and Master Documents](#)<sup>[2759]</sup>
- [Document Bookmarks](#)<sup>[2730]</sup>

### Learning Center topics

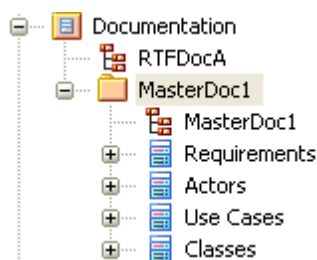
- Alt+F1 | Enterprise Architect | Reporting | Rich Text Reports | Generate a Virtual Document**

### 19.1.2.1 Create Master Document

If you want to create a series of **documents** - or **sections** of a document - that are individually tailored but are generated and organized as a single unit, you can design each document in its own **Model Document** element but organize these elements under a **Master Document**. This is a Package element with a child Common diagram to which you add the Model Document elements. You can then generate a document with a corporate template for the common contents and the headers and footers, and each section having its own appearance defined by a template appropriate to the section content. When you add Model Document elements to a Master Document, the Master Document element resembles the following:



The Master Document element and child diagram are listed in the Project Browser as shown:



#### Create a Master Document

Step	Action	See also
1	Open or create a diagram in which to create the Master Document. The diagram can be of any type, although <b>Documentation</b> automatically provides the correct Toolbox page.	<a href="#">Add New Diagrams</a> <sup>[822]</sup>
2	If the Toolbox Documentation page is not already shown, select <b>More tools   Documentation</b> .	<a href="#">Diagram Toolbox</a> <sup>[792]</sup> <a href="#">Virtual Documents</a> <sup>[2669]</sup>
3	Drag the <i>Master Document</i> icon onto the diagram. The system prompts you for the name of the Master Document element.	
4	Type the element name and click on the <b>OK</b> button. The system creates the Master Document element and a child Custom diagram of	



Step	Action	See also
	the same name.	
5	<p>If creating a Master Document for an HTML report, go to step 7.</p> <p>Otherwise, open the Tagged Values window (<b>View   Tagged Values</b>) and click on the Master Document element.</p> <p>The <b>RTFTemplate</b> Tagged Value displays in the window; this Tagged Value is filled with the valid template names.</p>	<a href="#">Tagged Values</a> <sup>[1134]</sup>
6	<p>The <b>RTFTemplate</b> Tagged Value defaults to <i>(model document: master template)</i>.</p> <p>If you want to use a customized master template that you have imported, click on the drop down arrow at the right of the field and click on that template in the list.</p>	<a href="#">Import a Document Template</a> <sup>[2707]</sup>
7	<p>Return to the Project Browser and open the Master Document child diagram. At this point, to provide the content for the generated document, you either:</p> <ul style="list-style-type: none"> <li>• Create the Model Document elements in the child diagram, or</li> <li>• Drag existing Model Document elements from the Project Browser into the child diagram</li> </ul>	<a href="#">Create Model Document</a> <sup>[2673]</sup>
8	Having set up the Master Document, and created or added and set up its Model Documents as required, you can generate the document report on the Master Document; this pulls in the contents of the Model Documents.	<a href="#">Generate the Document</a> <sup>[2679]</sup>

#### Learning Center topics

- **Alt+F1 | Enterprise Architect | Reporting | Virtual Documents | Create a Master Document**

#### **19.1.2.2 Create Model Document**

If you want to create a single document that captures specific details from specific areas of your model, you can define that document using a **Model Document** element. You can also create a number of individual documents and combine them as sections of a larger document, defining the larger document in a **Master Document** element. You can create as many Model Document elements as are necessary to provide the independent documents or the sections of the document you require.

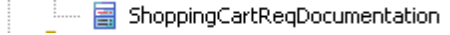
#### Prerequisites

Before you create a Model Document element, you might have to:

- Create the appropriate document or web page template to define the structure and appearance of the document
- For a document, determine whether to provide the content of the document using selected Packages or a Model Search and - if necessary - create the Model Search to use

**Create a Model Document element**

Step	Action	See also
1	<p>Either:</p> <ul style="list-style-type: none"> <li>Open the child Custom diagram of the Master Document element you have created or</li> <li>If you are creating separate Model Documents, create a new Documentation diagram (<b>Extended   Documentation</b> in the New Diagram dialog)</li> </ul> <p>The new diagram can live anywhere in the model hierarchy, outside the Packages you are adding to the document; you could create a diagram called Documentation within a specific Documentation Package, and use this to hold the independent Model Document elements for your virtual documents.</p> <p>You can also create Model Documents on any other convenient type of diagram, changing the Toolbox page to Documentation (step 2).</p>	<a href="#">Create Master Document</a> <sup>[2672]</sup> <a href="#">Add New Diagrams</a> <sup>[822]</sup>
2	<p>From the Documentation page of the Toolbox (<b>More tools   Documentation</b>) drag the <b>Model Document</b> icon onto the diagram to create a new Model Document element.</p> <p>Give the element an appropriate name; for example, if the documentation is relevant to the shopping cart requirements of a model, you could call it <i>ShoppingCartReqDocumentation</i>.</p> <p>Click on the <b>OK</b> button.</p>	
3	<p>If you are creating a Model Document element for a web report, go now to the <i>Add Packages to Model Document</i> topic.</p> <p>Otherwise, open the Tagged Values window (<b>View   Tagged Values</b>) and click on the Model Document element.</p> <p>The <b>RTTemplate</b>, <b>SearchName</b> and <b>SearchValue</b> Tagged Values display in the window.</p>	<a href="#">Add Packages to Model Document</a> <sup>[2675]</sup>
4	<p>Click on the drop-down arrow to the right of the <b>RTTemplate</b> field, and click on the template to use for this Model Document.</p>	
5	<p>If you are creating a list of Packages for the Model Document, go now to the <i>Add Packages to Model Document</i> topic.</p> <p>Otherwise, click on the drop-down arrow to the right of the <b>SearchName</b> field, and click on the model search type to populate this Model Document.</p> <p>In a Model Document for a document report, do not define both a list of Packages <b>and</b> a search; if both are present, when you generate the document it is created from the Package list only.</p>	<a href="#">Add Packages to Model Document</a> <sup>[2675]</sup>
6	<p>If necessary for your Model Search, type a search term in the <b>SearchValue</b> field.</p>	

Step	Action	See also
7	<p>Create further Model Document elements as required.</p> <p>Your Model Document element appears in the Project Browser with a Class icon, as shown below:</p>  <p>When you have created all the required Model Document elements, go to the <i>Document Order</i> topic.</p>	<a href="#">Document Order</a> <small>2677</small>

### Notes

- Diagram Searches are not supported
- Custom SQL searches are supported if they are returning elements; the SQL must include *ea\_guid* AS *CLASSGUID* (case sensitive) and the *object type*

### Learn more

- [Virtual Documents](#) 2669
- [Creating Search Definitions](#) 711

### Learning Center topics

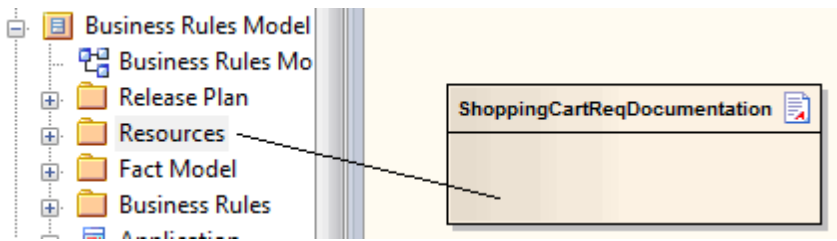
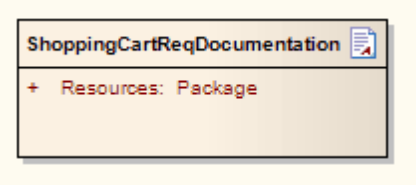
- **Alt+F1 | Enterprise Architect | Reporting | Virtual Documents |**
  - **Create from a Search**
  - **Generate RTF**

#### 19.1.2.3 Add Packages to Model Document

After you create a Model Document element, one of the two options available to you is to identify the Packages to document in the reports generated from the element. You can add as many Packages as you need, from anywhere in the model.

#### Add Packages to Model Document element

Step	Action	See also
1	Open the Documentation diagram, and locate a Package in the Project Browser to add to the documentation; for example, a Resources Package in a Dynamic view.	
2	Drag and drop the Package from the Project Browser onto the Model Document	

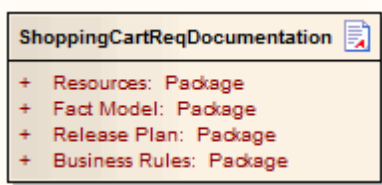
Step	Action	See also
	<p>element, as shown below:</p> 	
3	<p>Adding the Package creates an attribute with the same name as the Package; the attribute can be seen in the Model Document element in the <i>Attributes</i> compartment:</p>  <p>Repeat step 2 for each of the Packages you want to include in the document.</p>	
4	<p>You could now generate your report as a document or as a web page, but you might prefer to first review how your Package list impacts the order in which information is presented in the report.</p> <p>You can change the sequence of Packages, or delete any Packages that are no longer required.</p>	<p><a href="#">Generate the Document</a> <sup>[2679]</sup></p> <p><a href="#">Master Documents and Model Documents</a> <sup>[2759]</sup></p> <p><a href="#">Document Order</a> <sup>[2677]</sup></p> <p><a href="#">Delete Package in Model Document</a> <sup>[2676]</sup></p>

#### Learning Center topics

- **Alt+F1 | Enterprise Architect | Reporting | Virtual Documents | Reorder Packages**
- **Alt+F1 | Enterprise Architect | Reporting | Virtual Documents | Remove a Package**

#### **19.1.2.4 Delete Package in Model Document**

In your use of Model Document elements to generate documents on specific Packages, you might determine that a Package you have identified to be documented is no longer necessary. You can omit the Package from your documentation by deleting the corresponding attribute from the Model Document element.



### Delete Package from Model Document element

Step	Action	See also
1	In the Project Browser, expand the Model Document element to list the Package attributes.	
2	Right-click on the attribute for the Package to omit, and select the <b>Delete Attribute</b> context menu option.	

### Alternatively

Step	Action	See also
1	In either the Project Browser or the diagram, right-click on the Model Document element and select the <b>Attributes</b> context menu option. The Attributes dialog displays.	
2	On the <b>Attributes</b> list, click on the attribute for the Package to omit.	
3	Click on the <b>Delete</b> button to remove the attribute from the document element.	

### Learning Center topics

- **Alt+F1 | Enterprise Architect | Reporting | Virtual Documents | Remove a Package**

### **19.1.2.5 Document Order**

The order in which information is compiled into a virtual document depends on:

- The sequence of Model Document elements in a Master Document element
- Whether you define a Model Search in a Model Document element (not for web reports)
- Whether you define a Package list in a Model Document element, and how you order that list

You can review and, if necessary, amend the order in which information is to be compiled, before you generate the document or web page.

### Document Sequences

Aspect	Detail	See also
<b>Model Document Sequence</b>	<p>When you generate a document from a Master Document element, the sequence in which the sections are created is determined by the order in which the child Model Document elements are listed <b>in the Project Browser</b>.</p> <p>You can create elements anywhere in a diagram, therefore for sequencing information the document generator refers to the Project Browser.</p> <p>If necessary, change the sequence using the green <b>Move Element Up</b> and <b>Move Element Down</b> arrows in the Project Browser toolbar to move a Model Document element up or down within the Package.</p>	<p><a href="#">The Project Browser</a><sup>[646]</sup></p> <p><a href="#">Project Browser Toolbar</a><sup>[649]</sup></p>
<b>Model Search</b>	<p>A Model Search operates on the database, and accesses records in the order <b>in which they are stored</b>; this order depends on many factors, and can change with database maintenance.</p> <p>Therefore, the sequence of information provided by the search is unpredictable.</p>	
<b>Package Order</b>	<p>When you create a Package list in a Model Document element, the sequence of information is determined by the order in which the <b>Package attributes are listed within the element</b>.</p> <p>You can change the sequence using the Attributes dialog and, if you prefer a Package to be in a different section of the document, you can move the attribute from one Model Document element to another.</p>	



### Change sequence of Packages in a Model Document element

Step	Action	See also
1	<p>In the Project Browser or diagram, right-click on the Model Document element and select the <b>Attributes</b> context menu option.</p> <p>The Attributes dialog displays.</p>	
2	<p>On the Attributes list, click on attribute to move and click on the <b>Up</b> or <b>Down</b> (arrow) buttons to change the order in which the corresponding Packages are included in the documentation.</p>	
3	<p>Click on the <b>Close</b> button.</p>	

### Move a Package attribute from one Model Document to another

Step	Action	See also
1	In the Project Browser, expand both Model Document elements so that they list their Package attributes.	
2	Click-and-hold on the attribute to move, and drag it onto the name of the target Model Document element.	
3	Release the mouse button.  The attribute is removed from the source element and added to the top of the list of attributes in the target element.	
4	If necessary, move the attribute down the attribute list, as described in the <i>Change sequence of Packages in a Model Document element</i> section, above.	

#### Learn more

- [Generate the Document](#) 
- [Master Documents and Model Documents](#) 

#### Learning Center topics


- **Alt+F1 | Enterprise Architect | Reporting | Virtual Documents | Reorder Packages**

### 19.1.2.6 Generate the Document

When you have created one or more Model Document elements and, if necessary, a Master Document element, and you have set the document order and established any section numbering you want to apply, you can generate printable documentation from these elements.

**Access** **Project Browser | Master Document or Model Document: F8 or Diagram | Master Document or Model Document: F8**

#### Generate virtual documentation

Step	Action	See also
1	On the Generate Documentation dialog, set the generating options for your document as required.	<a href="#">Generate Documentation</a> 
2	Click on the <b>Generate</b> button to create the documentation.  The Report Generator works through the defined content of the Master Document element and/or the Model Document elements and pulls in the information from	

Step	Action	See also
	either the listed Packages or the executed searches, formatted according to the templates identified in the <i>RTFTemplate</i> Tagged Value for each document element.	
3	If you have not selected the <b>View Document on Completion</b> checkbox, click on the <b>View</b> button to view the documentation.	

#### Learn more

- [Create Model Document](#)<sup>[2673]</sup>
- [Create Master Document](#)<sup>[2672]</sup>
- [Add Packages to Model Document](#)<sup>[2675]</sup>
- [Document Order](#)<sup>[2677]</sup>
- [Generate Report to an Artifact Element](#)<sup>[2680]</sup>

#### Learning Center topics

- **Alt+F1 | Enterprise Architect | Reporting | Virtual Documents | Generate RTF**

### 19.1.2.7 Generate Report to an Artifact Element

When working with virtual documents, it is also possible to directly generate a report as a **linked document** associated with a **Document Artifact** element. Because the generated report is stored directly in the model, rather than in the current user's file system, it is much easier to share the report in a team-based modeling environment. The linked document also eliminates the separation of the modeled system and the generated document output, keeping everything integrated and readily accessible.

To generate a report to an Document Artifact's linked document, you start with either:

- A **Master Document** element with Model Document elements, or
- An **independent Model Document** element

These elements will contain templates, the Master Document will contain defined Model Documents and the Model Documents will identify Packages or Model Searches. At this point you can quickly and easily create a Document Artifact element on which to generate the report.



#### Create a Document Artifact element

Step	Action	See also
1	On the Documentation diagram, click on the Master Document element or the independent Model Document element.	
2	Drag and release the Quick Linker arrow to the right of the Master or Model Document element, and select the <b>Common   Document</b> menu option to create a connector and	<a href="#">Create New Elements</a> <sup>[897]</sup>



Step	Action	See also
	<p>the Artifact element.</p> <p>Optionally, you can name the element as well.</p>	

#### Generate a report to a Document Artifact element

Step	Action	See also
1	<p>Double-click on the Document Artifact.</p> <p>The Generate Documentation dialog displays.</p>	
2	<p>The text field values are automatically set.</p> <p>Select any checkboxes required (such as <b>View Document on Completion</b>), and click on the <b>Generate</b> button.</p>	<a href="#">Generate Documentation</a>  [2644]
3	<p>When the report generation is complete, the report displays as a linked document of the Document Artifact element (if not, click on the <b>View</b> button on the Generate Documentation dialog).</p>	<a href="#">Linked Documents</a>  [1044]

#### Note

- If you have a Master Document with Model Document elements, you can easily create a Document Artifact to test one of the Model Documents individually; in this case, you need to remove the Document Artifact before creating a report from the Master Document, otherwise the Artifact becomes part of the report instead of being a vehicle for the report

#### Learning Center topics

- **Alt+F1 | Enterprise Architect | Reporting | Virtual Documents | Generate Report to an Artifact Element**

### 19.1.3 Custom Document Templates

You use **report templates** to generate reports on your model that present specific information in a particular format, and to generate those reports repeatedly and consistently. You can select from a range of **system templates** on the Document Generator, to immediately generate reports from your model.

A template consists of:

- 'Sections' that identify the model components to report on
- Fields that identify the type of information to extract on each component
- Styles and formatting instructions that define the layout of the report
- Template Fragments containing smaller sub-templates


The system provides more specialized document templates for separate Cover Pages, Stylesheets and Tables of Contents.






You can also design your own **custom templates**, which provide the additional advantages of being able to generate reports tailored to your own organization, and to update certain aspects of the report without having to re-define every other specification. The report generator helps you **manage** these custom templates; that is:

- Review the custom templates available to you for generating reports
- Create additional custom templates
- Edit custom templates
- Import templates that were saved to XML files, and
- Delete any custom templates that are no longer required

**Access** **Project Browser package context menu | Documentation | Generate Documentation (F8) > Templates**

#### Manage your Custom Templates

Field/Button	Description	See also
<b>User Templates</b>	<p>Lists the <b>custom</b> Templates, Fragments, Stylesheets, Cover Pages and Tables of Contents currently available to you for generating reports through the Generate tab.</p> <p>Your custom templates can be grouped within these standard folders (by Template Type), or in another folder that you have created to hold specific templates - see below.</p>	<a href="#">Generate Documentation</a> 
<b>New</b>	<p>To create a new custom template, click on this button. The New Document Template dialog displays.</p> <p>In the <b>New template</b> field, type the template name.</p> <p>In the <b>Template Type</b> field, click on the drop-down arrow and select the type of template to create.</p> <p>In the <b>Copy Template</b> field, click on the drop down arrow and select either:</p> <ul style="list-style-type: none"> <li>• <b>None</b>, to create an empty template to develop from scratch, or</li> <li>• The name of an existing system, Technology or custom template to act as a base; this list is filtered to show templates of the type you specified in the <b>Template Type</b> field</li> </ul> <p>In the <b>Template Group</b> field either:</p> <ul style="list-style-type: none"> <li>• Select an existing group or</li> <li>• Enter the name for a new template group, or</li> <li>• Leave this field blank, to place the template in the appropriate standard type group</li> </ul>	

Field/Button	Description	See also
	<p>The <b>Template Group</b> determines the organization of document templates in the Resources window. You cannot create a template of a specific type and assign it to the standard group of a different type. For example, you cannot specify a Fragment and select to create it in the <i>Cover Page</i> folder. It either goes to a <b>user-created</b> group that <b>you select</b> or is automatically redirected to the <i>Fragment</i> folder.</p> <p>Click on the <b>OK</b> button; the template opens in the Document Template Designer.</p>	<a href="#">Design Custom Document Templates</a>  <sup>2684</sup>
<b>Edit</b>	<p>To make changes to an existing template, click on the template name and then click on this button.</p> <p>The template opens in the Document Template Designer.</p>	<a href="#">Design Custom Document Templates</a>  <sup>2684</sup>
<b>Import From Reference File</b>	<p>To import custom templates that were previously saved to <b>batch XML</b> files (that is, through the <b>Export Reference Data</b> option), click on this button. The Import Reference Data dialog displays.</p> <p>Click on the <b>Select File</b> button and browse for and select the required source XML file.</p> <p>In the Select Datasets to Import panel, click on the required datasets:</p> <ul style="list-style-type: none"> <li>• RTF Document Templates</li> <li>• Templates - RTF Style Detail</li> <li>• Templates - RTF Tag &amp; Language Options</li> </ul> <p>Click on the <b>Import</b> button to import the templates. The names of the imported templates are added to the list on the Templates tab.</p> <p>If the template you want to import is a single, external file (not in a <b>batch file</b>) you can instead <b>open</b> a template and use a different option to import the file into that template.</p>	<a href="#">Export Reference Data</a>  <sup>376</sup> <a href="#">Import Reference Data</a>  <sup>380</sup>  <a href="#">Import a Document Template</a>  <sup>2707</sup>
<b>Delete</b>	<p>To delete a custom template, click on the template name and click on the <b>Delete</b> button.</p> <p>A prompt displays to confirm the deletion.</p> <p>You can select multiple custom templates for deletion; press <b>Ctrl</b> or <b>Shift</b> as you click on each template name.</p>	

### Notes

- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is switched on, you need **Configure Resources** access permission to create document templates

- Whatever template you initially select on the Template tab or New Document Template dialog, of whichever template type and/or group, you can easily switch to a different template and template type within the Document Template Editor whenever you need to

#### Learn more

- [System Document Templates](#)  2648
- [List of Available Permissions](#)  329
- [Notes on Creating Stylesheets](#)  2705
- [Notes on Creating Tables of Contents](#)  2706
- [Notes on Creating Cover Pages](#)  2707
- [Template Fragments](#)  2708

#### Learning Center topics

- **Alt+F1 | Enterprise Architect | Reporting | RTF Templates**

### 19.1.3.1 Design Custom Document Templates

If you want to record and document details of your model, you can do so using a range of document reports, each generated using a document **template**. To meet your specific requirements, it is possible to **customize** the reports by developing **your own** templates; during this process you also have the option of generating **test reports** to review the output from your design as you develop it. The custom templates you can develop include full-document templates and independent Stylesheets, Cover Pages and Tables of Contents.

You **create** and **edit** report templates using the **Document Template Designer** to define:

- The model components to report on
- The information to extract on each component and
- The styles and layout of the report

It is possible to create a template from scratch or by copying and editing either another custom template or a system template. If you want to review the system templates you can list and display them using the Document Template Designer, and establish what templates already exist to either extract the information you require, or act as a starting point for your own design.

#### Access   **Tools | Document Template Designer**








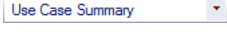
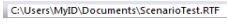
Also, through the Generate Documentation dialog or the *Document Generation* folders in the project Resources window




#### The Document Template Designer

The Document Template Designer consists of:

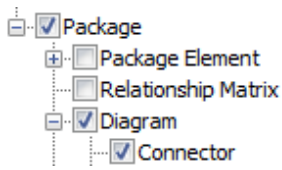
- A Toolbar, through which you create or select a template to work on, save your work, and generate and view the test reports
- A Sections panel, on which you select the components of the model to report on
- A Content panel, in which you develop the content and layout of the report, using an extensive range of context menu options

Toolbar Options

Icon	Action	See also
	Creates a new template. A prompt displays for you to enter the name of the new template and, optionally, the name of an existing template to use as a base.	<a href="#">Custom Document Templates</a> <small>[2682]</small>
	Saves changes to the current template.	
	Saves the current template as a new template. A prompt displays for the new template name.	
	Displays a range of document options that you can set to apply to all reports generated using the template. These options filter and organize the information presented in the report.	<a href="#">Document Options</a> <small>[2654]</small>
	Opens the Resources window and/or folder containing the currently-displayed document template, and highlights the template name.	<a href="#">Resources</a> <small>[1173]</small>
	Toggles the Document Generator view between showing the Sections hierarchy panel and hiding it.	<a href="#">Setting Sections for Reporting</a> <small>[2688]</small>
	Displays the Template Group name in which to locate or save the template; defaults to <b>Model Templates</b> .  Click on the drop-down arrow and select the appropriate Template Group name to search in. This filters the options in the next field to templates within that group only. For example, Cover Pages, or 'DavidG Reports'.	
	Displays a list of the templates in the Template Group you specified in the previous field. To view or edit an existing template, click on the drop-down arrow and select the template name.  <ul style="list-style-type: none"> <li>The system templates are protected; when you select one all editing options are grayed out</li> <li>Custom templates are available to edit</li> </ul>	<a href="#">Custom Document Templates</a> <small>[2682]</small>
	Displays the path and name of the file into which to generate the test report from the template.	

Icon	Action	See also
	If no file path is shown, you can either type one in or use the browser button (below).	
	Displays the browser dialog, in which you can locate and select the file to hold the generated test report.	
	Generates and displays the test report based on the template.  To generate a test report, you need to specify a target file (above) and select a Package, diagram or element in the Project Browser to run the report on. You could develop a special test Package for this purpose.	
	Displays the report held in the file specified in the file path field.  You can return to a generated report many times whilst you review the effects of different sections of the template you are editing.	

### The Sections and Content Panels

Facility	Description	See also
<b>Sections Panel</b>	Consists of a hierarchy of model components and their properties, each with a checkbox. To include information on a type of component in the report, you select the checkbox against it.  As you select checkboxes, corresponding pairs of open/close markers display in the Content panel; for example:	<a href="#">Setting Sections for Reporting</a> <sup>[2688]</sup>
	<div> <div> <b>Sections panel</b>  </div> <div> <b>Content Panel</b> <pre>package &gt; diagram &gt; connector &gt; &lt; connector &lt; diagram &lt; package</pre> </div> </div>	
<b>Content Panel</b>	The editing area, in which you develop the structure and content of the template.  As you add section markers to the template (as above), various pointer texts display between them such as:  <code>element&gt;</code>	<a href="#">Add Section Content</a> <sup>[2702]</sup>

Facility	Description	See also
	<p>[right-click-to-insert-Element-field(s)]  <b>&lt;element</b></p> <p>You replace this text with a range of fields that extract information from your model to display in the report.</p> <p>You can also add your own text.</p>	
<b>Content Panel Context Menu</b>	<p>Lists a wide range of options to:</p> <ul style="list-style-type: none"> <li>• Apply styles and formatting to the template</li> <li>• Insert objects such as tables, images and an internal contents list, and</li> <li>• Save, import and export the template file</li> </ul>	<a href="#">Custom Template Design Options</a> <sup>[2704]</sup>
<b>Content Panel Toolbars</b>	<p>Two text editing toolbars (based on the standard Windows toolbars) that you can use in conjunction with the context menu options.</p> <p>The lower toolbar provides an additional facility, for toggling the format of <b>sequenced</b> lists in the document. This is the drop-down arrow to the left of the bullet-list icon; it provides options for switching the list format of:</p> <ul style="list-style-type: none"> <li>• Single-level lists, with the sequence represented by arabic or roman numerals or upper case or lower case letters, and</li> <li>• Multiple-level lists, with the sequencing represented by arabic or roman numerals or symbols (whichever bullet symbol you are using)</li> </ul>	<a href="#">Format Paragraphs</a> <sup>[1064]</sup>

### Notes

- A standard file of global document styles, called Normal.rtf, is **automatically** applied to all new templates you create from scratch; it is possible to edit Normal.rtf to tailor the styles to your requirements, and to override these styles within the template
- You can also create special customized Tables of Contents and Cover Pages to be used optionally on any document report in place of any Cover Pages or Tables of Contents built into the report templates, and Stylesheets that can be chosen to override the Normal.rtf styles or any edited styles in a report template
- You can transport templates between models, using the **Export Reference Data** and **Import Reference Data** options

### Learn more

- [Generate Documents](#) <sup>[2642]</sup>
- [Resources](#) <sup>[1173]</sup>
- [Notes on Creating Stylesheets](#) <sup>[2705]</sup>
- [Notes on Creating Tables of Contents](#) <sup>[2706]</sup>
- [Notes on Creating Cover Pages](#) <sup>[2707]</sup>
- [Styles, Special Text & Table of Contents](#) <sup>[1054]</sup>

- [Export Reference Data](#)<sup>[376]</sup>
- [Import Reference Data](#)<sup>[380]</sup>

#### Learning Center topics

- **Alt+F1 | Enterprise Architect | Reporting | RTF Templates | Create a new template**

### 19.1.3.2 Setting Sections for Reporting

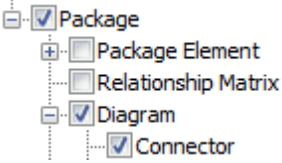


As part of designing or editing a template for generating a document report, you define the content to deliver in your report, specifically:

- Which **model features** to report on (such as elements, attributes, linked documents and Tagged Values), using the Sections panel of the Document Template Designer
- What **information** to provide on the instances of each feature (such as the name, type, phase, priority or author), using the Content panel of the Document Template Designer

The information you define here determines the structure of the report, what sections it contains and the order of those sections.

**Access**    **Tools | Document Template Designer > Sections**  
**Tools | Document Template Designer > Content**

#### Select model components to be documented in the report

Step	Action	See also
1	<p>Expand the some or all of the hierarchy in the Sections panel.</p> 	
2	<p>Select the checkbox next to the feature name; the feature name is then displayed as a pair of section tags in the Content panel of the Document Template Designer.</p> <p><b>element &gt;</b> <b>&lt; element</b></p> <p>For features in the lower levels of the hierarchy, selecting a child feature automatically selects the parent level(s) and adds the parent tags to the Contents panel.</p>	
3	<p>If you want to change the sequence of sections, click on the appropriate feature name in the Sections hierarchy and click on  or  to move the feature name up or down.</p>	



Step	Action	See also
	The position of the feature name within the Sections hierarchy determines the position of the section documenting that feature in the Content panel, and hence in the report. You can change the sequence either before or after you have selected the checkboxes, it makes no difference.	
4	For each selected feature, click between the section tags in the Contents panel and add text, or right-click and select the <b>Insert Fields</b> context menu option to add fields.	<a href="#">Add Section Content</a> <sup>[2702]</sup>

### Notes

- More specific guidance is provided for selecting the following features:
  - Child objects
  - Embedded elements
  - Profiled Relationship Matrices
  - Constraints and Scenarios
  - Linked Documents and Document Artifact contents
  - Elements held in external packages
  - Tagged Values
  - Sections formatted as tables
  - Using the Custom section for Custom Query Fragments

### Learn more

- [Design Custom Document Templates](#)<sup>[2684]</sup>
- [Child Objects](#)<sup>[2690]</sup>
- [Report on Embedded Elements](#)<sup>[2691]</sup>
- [Reporting Profiled Relationship Matrices](#)<sup>[2692]</sup>
- [Report on Constraints and Scenarios](#)<sup>[2693]</sup>
- [Reporting Linked Documents](#)<sup>[2694]</sup>
- [Report Elements From External Packages](#)<sup>[2695]</sup>
- [Report on Tagged Values](#)<sup>[2698]</sup>
- [Create Sections as Tables](#)<sup>[2700]</sup>
- [Generate Documentation](#)<sup>[2644]</sup>
- [Custom Query Fragments](#)<sup>[2711]</sup>

### Learning Center topics

- **Alt+F1 | Enterprise Architect | Reporting | RTF Templates | Add a Section**

### 19.1.3.2.1 Child Objects

An object in your model can have subordinate objects: for example, elements have child diagrams and other elements; packages contain other packages, elements and diagrams; and diagrams contain elements and connectors. When you set up your document report template to report on these subordinate objects, you can:

- Generate just the information you specify in selected fields, such as the object name only; for example, details of elements could be extracted in a number of places - Package elements, Embedded elements, source and target elements on connectors, child elements - and you might not require full details in all sections in your report, so you would just set a field for the element name in some of the sections

Alternatively

- Generate the same type of information as is provided for the parent-level objects of that type, by not defining any content (text or fields) between the section tags; for example, you could generate recursive documentation of child packages, getting the same level of detail as for the parent package

**Access** [Tools | Document Template Designer > Sections](#)  
[Tools | Documentation | Document Template Designer > Content](#)

#### Examples of reporting on child objects

Example	Detail	See also
<b>Specific object details</b>	<p>This template has valid content (text and the <b>Element.Name</b> field) between the <i>child element</i> tags. Therefore, in the generated report, the child element section will show only the element name and none of the detail shown for the parent element.</p> <pre> element&gt;{   {Element.Name}   Type: → {Element.Type} -- {Element.BaseClasses}   Status: → {Element.Status} .. Version {Element.Version} .. Phase {Element.Phase}   Package: → {Element.ParentPackage} ... Keywords: {Element.Tag}   Detail: → Created on {Element.DateCreatedShort}   {Element.Notes}   child element&gt;     → Child: {Element.Name}   &lt;child element &lt;/element </pre>	
<b>Default to parent-level detail</b>	<p>This template has no valid content between the <i>child element</i> tags. Therefore, in the generated report the child element section will default to the detail shown for the parent element.</p>	

Example	Detail	See also												
	<pre>element"&gt;{    {Element.Name}    Type: → → {Element.Type}..{Element.BaseClasses}   Status: → → {Element.Status}..Version: {Element.Version}..Phase: {Element.Phase}   Package: → → {Element.ParentPackage}..Keywords: {Element.Tag}   Detail: → → Created on: {Element.DateCreatedShort}    {Element.Notes}    child*element"&gt;   &lt;child*element   &lt;element</pre>													
Default parent detail for child objects	<table><tr><th>Child Object (Checkboxes)</th><th>Section Content Applied</th></tr><tr><td>Package &gt; Child Package</td><td>Package</td></tr><tr><td>Package &gt; Element &gt; Child Element</td><td>Package &gt; Element</td></tr><tr><td>Package &gt; Element &gt; Diagram</td><td>Package &gt; Diagram</td></tr><tr><td>Package &gt; Diagram &gt; Element</td><td>Package &gt; Element</td></tr><tr><td>Package &gt; Diagram &gt; Connector</td><td>Package &gt; Element &gt; Connec</td></tr></table>	Child Object (Checkboxes)	Section Content Applied	Package > Child Package	Package	Package > Element > Child Element	Package > Element	Package > Element > Diagram	Package > Diagram	Package > Diagram > Element	Package > Element	Package > Diagram > Connector	Package > Element > Connec	
Child Object (Checkboxes)	Section Content Applied													
Package > Child Package	Package													
Package > Element > Child Element	Package > Element													
Package > Element > Diagram	Package > Diagram													
Package > Diagram > Element	Package > Element													
Package > Diagram > Connector	Package > Element > Connec													

### Notes

- In principle, it is better to leave Child sections blank to replicate their parent sections

Child sections do not contain the same sub-section detail as their parents; **Element::Child Element** does not contain sub-sections such as Scenario or Attribute, so where Child Element sections are populated, these sub-sections are not rendered

An exception to this is cases where sub-sections are not required, but different formatting of the section fields is preferred

### Learn more

- [Setting Sections for Reporting](#)<sup>[2688]</sup>
- [Add Section Content](#)<sup>[2702]</sup>

#### 19.1.3.2.2 Report on Embedded Elements

As part of your model, you might have created embedded (structural) elements on an element, such as:

- ActivityParameter
- RequiredInterface
- ActionPin
- Port
- EntryPoint
- ExpansionNode
- ObjectNode

- ProvidedInterface
- ExitPoint
- Part

When you design a document report template, you can add a section to specifically identify and report on these embedded elements.

**Access** [Tools | Document Template Designer > Sections](#)

#### Add section to report on embedded elements on an element

Action	Detail	See also
<b>Add section</b>	<p>Select the <b>Package &gt; Element &gt; Embedded Elements</b> checkbox. The embedded elements section markers are added to the template:</p> <pre>embedded elements &gt; &lt;embedded elements</pre> <p>When the report is generated from the template, any embedded elements are listed with their parent element.</p>	

#### Notes

- The Embedded Elements section does not contain the same sub-sections as the parent Element section, such as Scenario or Attribute; if you do not define fields in the Embedded Elements section, it defaults to displaying the same detail as is in the Element section, whereas if you do add fields, only those field values are shown

#### Learn more

- [Setting Sections for Reporting](#) <sup>[2688]</sup>
- [Child Objects](#) <sup>[2690]</sup>
- [Add Section Content](#) <sup>[2702]</sup>
- [Manage Structural Elements](#) <sup>[935]</sup>

#### **19.1.3.2.3 Reporting Profiled Relationship Matrices**

When you are using the Relationship Matrix, you search on combinations of source and target packages, element types, connector type and connector direction. You can save your frequently-used combinations as profiles, and you can include the results of using these profiles in your document report by setting up a section in the report template.

You do not directly specify a matrix profile; the Document Generator returns the Relationship Matrix for all profiles that include the documented package as the source **or** target package.

**Access** [Tools | Document Template Designer > Sections](#)

**Add section to report on Relationship Matrix contents**

Action	Detail	See also
<b>Add section</b>	<p>Select the <b>Package &gt; Relationship Matrix</b> checkbox; the relationship matrix section markers are added to the template.</p> <p>Right-clicking between the markers and add the Image field to return the relationship matrix in the report.</p> <pre>relationship matrix &gt; {Matrix.Image} &lt; relationship matrix</pre> <p>You can also insert a number of fields to identify specific aspects of the profile, such as relationship name, direction, type, target element and element type, and source element and element type.</p>	<a href="#">Add Section Content</a> <sup>[2702]</sup>

**Notes**

- You can view a list of the Relationship Matrix profiles in your project in the Resources window, in the *Matrix Profiles* folder

**Learn more**

- [Setting Sections for Reporting](#)<sup>[2688]</sup>
- [Relationship Matrix](#)<sup>[727]</sup>
- [Matrix Profiles](#)<sup>[735]</sup>
- [Resources](#)<sup>[1173]</sup>

**19.1.3.2.4 Report on Constraints and Scenarios**

In the Document Template Designer, you can create sections to include information from your model on:

- Constraints* on Package Elements, Elements, Connectors and Attributes, and
- Scenarios* for Package Elements and Elements

You can set additional options in the Constraints and Scenarios sections to define what **types** of constraint or scenario are included in your reports.

**Access**   **Tools | Document Template Designer > Sections**

**Set sections to extract Constraint and Scenario details**

Object	Detail	See also
<b>Constraints</b>	<p>In the Sections panel, you can choose any or all of the following constraint option checkboxes:</p> <ul style="list-style-type: none"> <li><b>Constraint-Pre</b> - to include all constraints of the type 'pre-condition' in</li> </ul>	<a href="#">Constraints</a> <sup>[963]</sup>

Object	Detail	See also
	<p>this section of the report</p> <ul style="list-style-type: none"> <li>• <b>Constraint-Post</b> - to include all constraints of the type 'post-condition' in this section of the report</li> <li>• <b>Constraint</b> - to include all constraints that have not been generated in the Pre-Constraint and Post-Constraint sections of the report</li> </ul> <p>Set any fields you need, between the section markers.</p>	<p><a href="#">Add Section Content</a><sup>[2702]</sup></p>
<b>Scenarios</b>	<p>In the Sections panel, you can choose any or all of the following scenario option checkboxes:</p> <ul style="list-style-type: none"> <li>• <b>Element &gt; Scenario</b> - to include all scenarios in this section of the report; if any of the following sections are also selected, the report includes all scenarios that are not exception paths</li> <li>• <b>Element &gt; Scenario &gt; Exception</b> - to include all exceptions for each scenario</li> <li>• <b>Element &gt; Scenario &gt; Structured Scenarios</b> - to include all scenario steps in the scenario sections of the report</li> <li>• <b>Element &gt; Scenario &gt; Structured Scenarios &gt; Exception</b> - to include the steps for each exception path in the scenario sections of the report</li> </ul> <p>Set any fields you need, between the section markers.</p>	<p><a href="#">Scenarios</a><sup>[965]</sup></p> <p><a href="#">Add Section Content</a><sup>[2702]</sup></p>

#### Learn more

- [Setting Sections for Reporting](#)<sup>[2688]</sup>

#### 19.1.3.2.5 Reporting Linked Documents

If you have large quantities of information to provide on an element or package, you can attach it as a formatted document either as the content of a Document Artifact element associated with your subject element, or by attaching the document file directly to the element (depending on the edition of Enterprise Architect you are using). In either case, you can include the linked document contents **in your document report** by selecting one or more of the **Linked Document** checkboxes in the report template definition.

**Access**   **Tools | Document Template Designer > Sections**

#### Add sections to report on Linked Documents

Action	Detail	See also
<b>Add sections</b>	<p>Select the checkboxes for the levels of Linked Document you want to include in the report:</p> <ul style="list-style-type: none"> <li>• <b>Package &gt; Package Element &gt; Linked Document</b></li> </ul>	<p><a href="#">Linked Documents</a><sup>[1044]</sup></p> <p><a href="#">Creating Document Artifacts</a><sup>[1046]</sup></p>

Action	Detail	See also
	<ul style="list-style-type: none"> <li>Package &gt; Package Element &gt; External Requirements &gt; Linked Document</li> <li>Package &gt; Element &gt; Linked Document or</li> <li>Package &gt; Element &gt; External Requirements &gt; Linked Document</li> </ul> <p>The linked document is rendered into the document report at:</p> <pre>linked document &gt; &lt;linked document</pre>	

### Notes

- In some system templates that you might copy, the **Linked Document** checkbox is only available as a child of the **External Requirements** checkbox
- You do not need to add any fields between the linked document section markers

### Learn more

- [Setting Sections for Reporting](#) 

### Learning Center topics

- Alt+F1 | Enterprise Architect | Reporting | RTF Templates | Include Linked Documents**

#### 19.1.3.2.6 Report Elements From External Packages

Elements can be re-used in different diagrams across a model, which can often mean that a diagram contains 'external' elements from packages **other than** the diagram's parent package. Using the Document Template Designer, you can create **custom** templates to generate reports showing all elements used in a package with 'internal' and 'external' elements grouped separately, or only the elements actually held within a package. You can also select options to expose the external elements from all diagrams in a package, or only from selected diagrams in the package.

The separation of 'external' and 'internal' elements avoids duplication of information and identifies the elements that impact the package but are not part of the package structure. Normally, you would report the same level of detail for each internal and external element, but you can also define a smaller group of details for the external elements, including the name of the parent package.

**Access**   **Tools | Document Template Designer > Sections**

**Add sections to include elements in diagrams**

Step	Action	See also
1	<p>In the Sections panel on the left-hand side of the editor window, select the <b>Package &gt; Element</b> checkbox.</p> <p>Selecting the checkbox adds the following set of sections to the Content panel of the template:</p> <pre>package &gt; [right-click-to-insert-Package-field(s)] element &gt; [right-click-to-insert-Element-field(s)] &lt; element &lt; package</pre>	<a href="#">Setting Sections for Reporting</a> <sup>[2688]</sup>
2	<p>Delete the two [right-click-to...] texts and after the:</p> <ul style="list-style-type: none"> <li><code>package &gt;</code> section marker, type <b>Package Name:</b> then right-click and select the <b>Insert Field   Name</b> option</li> <li><code>element &gt;</code> section marker, type <b>Element Name:</b> then right-click and select the <b>Insert Field   Name</b> option</li> </ul> <p>The template now resembles the following illustration:</p> <pre>package &gt; Package Name: {Pkg.Name} element &gt; Element Name: {Element.Name} &lt; element &lt; package</pre>	<a href="#">Add Section Content</a> <sup>[2702]</sup>
3	<p>In the Sections panel, select the <b>Package &gt; Diagram &gt; Element</b> checkbox.</p> <p>Selecting the checkbox extends the set of sections in the Content panel of the template:</p> <pre>package &gt; Package Name: {Pkg.Name} element &gt; Element Name: {Element.Name} &lt; element diagram &gt; [right-click-to-insert-Diagram-field(s)] element &gt; [right-click-to-insert-Element-field(s)] &lt; element &lt; diagram &lt; package</pre>	
4	<p>Delete the two [right-click-to...] texts and after the:</p> <ul style="list-style-type: none"> <li><code>diagram &gt;</code> section marker type <b>Diagram:</b> then right-click and select the</li> </ul>	



Step	Action	See also
	<p><b>Insert Field   Name</b> option</p> <p>Delete the space between the <b>element &gt;</b> section markers. The template now resembles the following illustration:</p> <pre> package &gt; Package Name: {Pkg.Name} element &gt; Element Name: {Element.Name} &lt; element diagram &gt; Diagram: {Diagram.Name} element &gt; &lt; element &lt; diagram &lt; package </pre> <p>Because the <b>Diagram &gt; Element</b> section does not contain any fields, the type of content of this section in the generated report is determined by the <b>Package &gt; Element</b> section.</p>	

### Specify diagrams & generate report

Step	Action	See also
1	<p>Select the appropriate option:</p> <ul style="list-style-type: none"> <li>In the Diagram Properties dialog for <b>selected</b> diagrams, the <b>Document each contained element in RTF</b> checkbox, to identify <b>all</b> the elements in those diagrams in your report, or</li> <li>On the Generate RTF Documentation dialog, the <b>Include all diagram elements in report</b> checkbox to include <b>all</b> elements in <b>each and every</b> diagram covered by the report</li> </ul>	<a href="#">Diagram Options</a> <sup>[2667]</sup> <a href="#">Generate Documentation</a> <sup>[2644]</sup>
2	<p>In the Project Browser, select the package to report on, and execute the report through either:</p> <ul style="list-style-type: none"> <li>The Document Template Designer, or</li> <li>The Generate Documentation Dialog</li> </ul>	<a href="#">Design Custom Document Templates</a> <sup>[2684]</sup> <a href="#">Generate Documentation</a> <sup>[2644]</sup>
3	The generated report for this example resembles the following output:	

Step	Action	See also
	<p> <a href="#">Package Name: BlockFrame</a>  <a href="#">Element Name: Activity1</a>  <a href="#">Element Name: ActivityInitial</a>  <a href="#">Element Name: Block1</a>  <a href="#">Element Name: Block2</a>  <a href="#">Element Name: Block3</a>  <a href="#">Element Name: Block4</a>  <a href="#">Element Name: Block8</a>  <a href="#">Element Name: ExceptionHandler1</a>  <a href="#">Diagram: BlockFrame</a>  <a href="#">Element Name: Class-who</a>  <a href="#">Element Name: Class2</a>  <a href="#">Element Name: Class3</a> </p> <p>The report shows that the <i>BlockFrame</i> package contains 8 elements, but has links to three elements from other packages because those three elements have been used in the <i>BlockFrame</i> diagram.</p>	

### Notes

- The illustrations in this description show the very simplest set-up, to achieve the result of identifying external elements in a report; you can, if you wish, populate the report with other sections and fields, leaving the **Diagram > Element** section empty to duplicate the structure of the **Package > Element** section
- If you do add fields to the **Diagram > Element** section, the section will have its **own** structure and format and can provide different element details; for example, selecting the **Insert Field | Name (Full)** menu option adds the `{Element.FullName}` field, which identifies each external element's parent package
- If you insert the **Package > Diagram > Element** section but *not* the **Package > Element** section, the subsequent report will show just the elements used in each diagram, without distinguishing between external and internal elements
- If you insert the **Package > Element** section but not the **Package > Diagram > Element** section, the subsequent report will show just the elements held in the package

### Learning Center topics

- [Alt+F1 | Enterprise Architect | Reporting | RTF Templates | Reporting Elements from External Packages](#)

#### 19.1.3.2.7 Report on Tagged Values

Tagged Values are used extensively across the objects of your model, including:

- Package Elements
- Elements
- Connectors and their Source and Target elements
- Attributes
- Operations (Methods) and

- External Requirements

When designing a template to report on these objects, you can include information on their Tagged Values either:

- As a group, using the Tagged Value subsection to identify all Tagged Values used in the object, or
- Individually, using the **valueOf()** field to extract a specific value of a specific tag

Access   **Tools | Document Template Designer > Sections**

#### Extract information on Tagged Values

Action	Detail	See also
<b>Insert a Tagged Value section</b>	<p>The Tagged Value section reports all Tagged Values for the object.</p> <p>In the Sections panel, select the checkbox for each object to report on, and for the Tagged Value subsection for that object. For example:</p> <ul style="list-style-type: none"> <li>• <b>Package &gt; Package Element &gt; Tagged Value</b></li> <li>• <b>Package &gt; Connector &gt; Source &gt; Tagged Value</b></li> <li>• <b>Package &gt; Element &gt; External Requirements &gt; Tagged Value</b>, or</li> <li>• <b>Package &gt; Element &gt; Method &gt; Tagged Value</b></li> </ul> <p>The Tagged Value section markers display in the Contents panel.</p> <p><b>tagged value &gt;</b>  <b>[right-click-to-insert-Tagged-Value-field(s)]</b>  <b>&lt; tagged value</b></p> <p>Delete the <i>[right-click-to-...]</i> text, right-click in the space and select the <b>Insert Fields</b> option to add Tagged Value <b>Name</b>, <b>Notes</b> and/or <b>Value</b> fields.</p>	<p><a href="#">Setting Sections for Reporting</a> <sup>[2688]</sup></p> <p><a href="#">Add Section Content</a> <sup>[2702]</sup></p>
<b>Insert a valueOf() field</b>	<p>For any object having a Tagged Value, you can add a <b>valueOf()</b> field to your report template to extract the value of a specific tag. This should be one of the tags normally associated with the object, such as <i>ConnectorAltName</i> for a connector.</p> <p>Obtain the name of the tag before starting this procedure.</p> <p>To insert the field, right-click between the section tags for the <b>object</b> and select the <b>Insert Field   valueOf</b> option.</p> <p>The Document Template Designer prompts you to type in the name of the tag from which to extract the value. Type the name and click on the <b>OK</b> button. The Document Template Designer adds the field at the cursor position, in the format:</p> <p style="text-align: center;">{Objecttype.valueOf(tagname)}</p>	<p><a href="#">Add Section Content</a> <sup>[2702]</sup></p>

Action	Detail	See also
	<p>For example:</p> <pre>{Connector.valueOf(ConnectorAltName)}</pre> <p>The <b>valueOf</b> field extracts just the <b>value of the tag</b>, so for clarity you could type some lead-in text or the meaning of the tag, immediately preceding the value field. For example:</p> <p><b>Alternative Name:</b> {Connector.valueOf(ConnectorAltName)}</p>	

#### 19.1.3.2.8 Create Sections as Tables

In a **document report template**, you can render a model section as a table, defined with any number of columns but with only *two* rows:

- The first row is used to describe the **headings** of the columns, which you define and format yourself
- The second row defines the output, which you specify by right-clicking in each cell and selecting the output type from the **field list**; the output is then generated iteratively for every occurrence of the section in question

For example, you might set up the **Model > Glossary >** section as a section table, using the table definition:

model >		
<b>Model Glossary</b>		
glossary >		
<b>Term</b>	<b>Type</b>	<b>Meaning</b>
{ModelGlossary.Term}	{ModelGlossary.Type}	{ModelGlossary.Meaning}
< glossary		
< model		


This table definition renders the document output:

<b>Model Glossary</b>		
<b>Term</b>	<b>Type</b>	<b>Meaning</b>
Accounting Periods	Business	A defined period of time whereby performance reports may be extracted. (normally 4 week periods)
Association	Technical	A relationship between two or more entities. Implies a connection of some type - for example one entity uses the services of another, or one entity is connected to another over a network link.

**Access** Select **Table** from the menu bar at the top of the Document Editor window, or right-click and select the **Table** option within either:

- The Content panel of the Document Template Designer or
- The text area of the document

### Create and format a table

To	Action	See also
<b>Insert a table in the document</b>	<p>Position the cursor on the point in the text to create the table, and select the <b>Table   Insert Table</b> menu option.</p> <p>The New Table Parameters dialog displays, in which you specify the number of table rows and columns. For a model section table in a report template you can accept the default values of two rows and three columns. Click on the <b>OK</b> button.</p> <p>Your table might be invisible; if so, select the <b>Table   Show Gridlines</b> menu option to reveal the table and cell borders in dotted lines. These lines are for guidance in creating the document, and do not display on the printed document. You can add printable borders if you wish, using other context menu options.</p> <p>The editor initially creates a table with cells of equal width across the page; you can change the cell width by dragging the cell borders using the mouse, or using other context menu options.</p>	<a href="#">Create Tables</a> 
<b>Add a header row</b>	<p>Select the top row of the table, and then select the <b>Table   Header Row</b> menu option. Apply any heading text, settings and formatting to the highlighted row.</p> <p>In the document or compiled report, the heading row is repeated at the top of the columns each time the table flows on to a new page. In a report, if the heading row is populated with the values from field names, the heading rows on subsequent pages reflect the values from that first row.</p> <p>This option operates on the single row at the top of the table, and not on multiple rows.</p>	
<b>Populate data row with fields</b>	<p>Right-click on the table cell and select the <b>Insert Field   &lt;field name&gt;</b> option to add one or more fields to the cell.</p> <p>When the report is generated, the field values are extracted and displayed in the table for each instance of the object, such as an element or attribute.</p>	

### Notes

- In a document report template, if you type a carriage return between the end of the table and the section terminator, the table you generate has a line space between the rows; for example:

```

package*>@
?
@
element*>@
*      ElementName*      Author*
*      {ElementName}*      {ElementAuthor}*
@
<~element@
<~package@

```

← Carriage Return

This generates the table:

	Element Name	Author
	Choose Recipient	John Redfern

Line Space

	Manage Contacts	Walter Frederick
--	-----------------	------------------

To avoid this, remove any carriage return between the end of the table and the section terminator, as follows:

```

package*>@
?
@
element*>@
*      ElementName*      Author*
*      {ElementName}*      {ElementAuthor}*
<~element@
<~package@

```

← No Carriage Return

This generates a table with no space between the rows:

	Element Name	Author
	Choose Recipient	John Redfern
	Manage Contacts	Walter Frederick

#### Learning Center topics

- **Alt+F1 | Enterprise Architect | Reporting | RTF Templates | Add a Table** (to a report)

#### 19.1.3.2.9 Add Section Content

In the Document Template Designer, as you select the check boxes in the Sections panel to report on the different objects and features of the model, pairs of highlighted tags are added to the Contents panel. Many of these pairs of tags have a short instruction between them, as indicated below:

```

sectionname >
[right-click-to-insert-<objectname>-fields]
< sectionname



```

Where the text message is shown, you can delete it and use the space between the tags to provide the **content** of the report, by:

- Typing or pasting boiler-plate text that you want to include in your reports, and
- Inserting fields that extract the values of specific characteristics and properties of the object

**Access** [Tools | Document Template Designer > Content](#)

#### Add information content to template

Action	Detail	See also
<b>Add text</b>	<p>Click in the space between the tags (or anywhere else in the template) and type or paste <b>Ctrl+V</b> the text you want to display in the report.</p> <p>Highlight and right-click on text to format it, using the Document Editor commands.</p>	<a href="#">Custom Template Design Options</a>  2704
<b>Insert fields</b>	<p>Right-click in the area between the opening and closing tags, and select the <b>Insert Field</b> context menu option; this displays a context-sensitive list of fields to add to this section of the template.</p> <p>Click on the name of the field you want to add. The Document Template Editor displays the field tag in the text space. For example:</p> <pre>{Element.ParentPackage}</pre> <p>When you generate a report from the template, the report generator replaces the field name with the actual value. You might need to type lead-in text and add spaces to support the field value, or use the fields in a table with row and column headings.</p> <p>You can insert as many fields as you want from the list; you usually apply each field once, as the report generator applies the template instructions to each instance of the object.</p>	<a href="#">Create Tables</a>  1075

#### Notes

- As the **model** and **linked document** tags do not require fields, you cannot display the editor context menu between these tags
- The **child element**, **embedded element** and **child package** tags represent sections that, **can** have defined fields, but these fields prevent the sections from replicating the structure of their parent **element** and **package** sections; as it is preferable to use the parent structures, these tags do not **prompt** you to add fields with the *[right-click-to-insert-<objectname>-fields]* message, although the message is still applicable
- If you select a field with short date format (such as *Pkg.DateCreatedShort*, *Diagram.DateModifiedShort* or *Element.DateCreatedShort*) the format is actually drawn from the MS Windows settings

To use a different short date format, click on the **Start** icon on the Windows desktop and select the **Control Panel | Regional and Language Options | Customize** option

#### Learn more

- [Design Custom Document Templates](#) 
- [Setting Sections for Reporting](#) 

#### Learning Center topics

- **Alt+F1 | Enterprise Architect | Reporting | RTF Templates | Add a Field**

### 19.1.3.3 Custom Template Design Options











The Document Template Designer, through the common document Editor, provides a wide range of options for defining the format and styles of the document reports generated from your custom templates. You access the Document Editor options through a context menu and, for some of the functions, through the toolbars at the top of the Content panel.

**Access**   **Tools | Document Template Designer: <template>**

When you have opened the template in the Designer, you display the context menu by right-clicking in the **Content** panel.

#### Option Descriptions

The menu option descriptions are grouped according to the actions you are performing.

Actions	Link
<b>Creating and importing templates</b>	<a href="#">File Control</a> 
<b>Configuring the editor page display and formatting tools shown</b>	<a href="#">Editor Tool Display Options</a> 
<b>Incorporating stylesheets, special texts and Tables of Contents</b> <b>Managing the base styles in the Normal.rtf style template file</b>	<a href="#">Styles, Special Texts &amp; Table of Contents</a>  <a href="#">The Normal.rtf Style Template</a> 
<b>Moving through, searching and selecting text</b>	<a href="#">Scroll, Search and Select Text</a> 
<b>Formatting characters and text strings</b>	<a href="#">Format Text</a> 
<b>Formatting paragraphs and text blocks</b>	<a href="#">Format Paragraphs</a> 
<b>Inserting tab points</b>	<a href="#">Set Tabs</a> 
<b>Inserting sections, columns and page breaks, and repaginating</b>	<a href="#">Define Document Sections</a> 
<b>Inserting and editing page headers and footers, and footnotes and endnotes</b>	<a href="#">Insert Headers, Footers, Footnotes and Endnotes</a> 



Actions	Link
Inserting tables	<a href="#">Create Tables</a> <sup>[1075]</sup>
Setting up User-Defined Section Numbering (in document report templates)	<a href="#">Apply User-Defined Section Numbering</a> <sup>[1081]</sup>
Inserting hyperlinks and bookmarks	<a href="#">Insert Reference Links</a> <sup>[1083]</sup>
Inserting images, OLE objects, frames and drawing objects	<a href="#">Insert Images, Objects and Frames</a> <sup>[1085]</sup>
Checking the spelling of text, and word use	<a href="#">Checking Text</a> <sup>[1089]</sup>
Tracking changes and incorporating or rejecting them	<a href="#">Track Changes</a> <sup>[1091]</sup>
Protecting document and template text from accidental change	<a href="#">Protect Document Contents</a> <sup>[1093]</sup>
Printer setup and printing documents	<a href="#">Print Report Documents</a> <sup>[1094]</sup>

### Notes

- Throughout your template editing:
  - To undo one or more immediately previous edits, press **Ctrl+Z**, or select the **Edit | Undo** menu option; you can still undo a change even after you have saved the change
  - To redo one or more immediately previous undone edits, press **Ctrl+Y**, or select the **Edit | Redo** menu option

#### 19.1.3.4 Notes on Creating Stylesheets

**Stylesheets** define a range of formatting parameters that control the appearance and layout of a document report. You create them to provide an alternative set of styles to the Normal.rtf file and to any special styles redefined within a document report template. A user can **choose** to apply a Stylesheet to **any** kind of document report they are generating, in which case the styles in the Stylesheet override the styles with matching names in the template or Normal.rtf. You can, therefore, create a Stylesheet that:

- Redefines and overrides certain styles
- Does not include other named styles, so use of those styles defaults to the template or Normal.rtf file definition, and
- Provides additional styles that can be applied by editing the generated report

Each custom Stylesheet can define a totally different page layout and document styles to other Stylesheets and to the Normal.rtf Stylesheet.

The simplest way to create a new Stylesheet is to copy one of the system-provided files (for print output or for

online output) or another custom Stylesheet, and edit the style definitions in the copy. However, you can create a Stylesheet from scratch if you wish; this would be initially derived from the Normal.rtf file.

**Access** **Project | Resources > Document Generation > User Templates | right-click on Style Sheets | Create Document Template**  
**Project | Resources > Document Generation > System Templates > Style Sheets | right-click on <stylesheet name> | Copy Document Template**  
**Tools | Document Template Designer | **

You can also create Stylesheets within the Templates tab of the Generate Documentation dialog

In each of these cases, you create the new Stylesheet file through the New Document Template dialog, and edit the styles in the Document Template Designer using the Styles, Special Text and Table of Contents facilities.


#### Learn more

- [Selecting a Stylesheet](#)<sup>[2653]</sup>
- [Custom Document Templates](#)<sup>[2681]</sup> (New Document Template dialog)
- [Design Custom Document Templates](#)<sup>[2684]</sup>
- [Custom Template Design Options](#)<sup>[2704]</sup>
- [Styles, Special Text & Table of Contents](#)<sup>[1054]</sup>

### 19.1.3.5 Notes on Creating Tables of Contents

A **Table of Contents** template defines the style, levels and numbering formats of a document report Table of Contents that a user can **choose** to apply to **any** kind of document report they are generating, in which case the Table of Contents template overrides any Table of Contents styles defined in the template or Normal.rtf file.

The simplest way to create a new Table of Contents is to copy one of the system-provided files (for portrait output or for landscape output) or another custom Tables of Contents, and edit the style definitions in the copy. However, you can create a Table of Contents from scratch if you wish; this would be initially derived from the Normal.rtf file.

**Access** **Project | Resources > Document Generation > User Templates | right-click on Table of Contents | Create Document Template**  
**Project | Resources > Document Generation > System Templates > Table of Contents | right-click on <table of contents name> | Copy Document Template**  
**Tools | Document Template Designer | **

You can also create Table of Contents within the Templates tab of the Generate Documentation dialog

In each of these cases, you create the new Table of Contents template file through the New Document Template dialog, and edit the styles in the Document Template Designer using the Styles, Special Text and Table of Contents facilities.

#### Learn more


- [Selecting a Table of Contents](#)<sup>[2652]</sup>
- [Custom Document Templates](#)<sup>[2681]</sup> (New Document Template dialog)

- [Design Custom Document Templates](#) <sup>[2684]</sup>
- [Custom Template Design Options](#) <sup>[2704]</sup>
- [Styles, Special Text & Table of Contents](#) <sup>[1054]</sup>

### 19.1.3.6 Notes on Creating Cover Pages

A **Cover Page** template defines the style, fields and format of a document report front cover that a user can **choose** to apply to **any** kind of document report they are generating, in which case the Cover Page styles override any styles of the same name defined in the report template or Normal.rtf file.

The simplest way to create a new Cover Page is to copy one of the system-provided files (for portrait output or for landscape output) or another custom Cover Page Contents, and edit the style definitions in the copy. However, you can create a Cover Page from scratch if you wish; this would be initially derived from the Normal.rtf file.

**Access** **Project | Resources > Document Generation > User Templates | right-click on Cover Pages | Create Document Template**  
**Project | Resources > Document Generation > System Templates > Cover Pages | right-click on <cover page name> | Copy Document Template**  
**Tools | Document Template Designer | **

You can also create a Cover Page within the Templates tab of the Generate Documentation dialog

In each of these cases, you create the new Cover Page template file through the New Document Template dialog, and edit the styles in the Document Template Designer using the Styles, Special Text and Table of Contents facilities.

#### Learn more

- [Selecting a Cover Page](#) <sup>[2654]</sup>
- [Custom Document Templates](#) <sup>[2681]</sup> (New Document Template dialog)
- [Design Custom Document Templates](#) <sup>[2684]</sup>
- [Custom Template Design Options](#) <sup>[2704]</sup>
- [Styles, Special Text & Table of Contents](#) <sup>[1054]</sup>




### 19.1.3.7 Import a Document Template

Many organizations have established corporate formats and templates, or might design templates outside Enterprise Architect. If you want to use such 'external' templates from your business to apply formats and standards to your document reports, you can import them into your document template library.

**Access** **File | Import**

#### Import a template from an external directory




Step	Action	See also
1	Save the external template as a document.	

Step	Action	See also
2	In Enterprise Architect, create a new blank template; name the template but do not specify an existing template to copy from.	<a href="#">Custom Document Templates</a>  <sup>[2682]</sup>
3	When the template is listed on the Templates tab of the Generate Documentation dialog, click on the name and click on the <b>Edit</b> button.  The Document Template Designer displays.	<a href="#">Design Custom Document Templates</a>  <sup>[2684]</sup>
4	Right-click in the Content panel and select the <b>File   Import</b> context menu option.  The File Open dialog displays.	
5	Locate your template file, and click on the <b>Open</b> button.  The Open dialog closes, returning you to the Document Template Designer; this now contains your imported template.	
6	Right-click and select the <b>File   Save</b> menu option.  If necessary, make any changes to the template and select <b>File   Save</b> again before selecting <b>File   Close</b> to exit the dialog.	<a href="#">Custom Template Design Options</a>  <sup>[2704]</sup>

### Notes

- Standard graphical images (such as a logo in the header, main text or footer) are imported; however, any **meta-file** graphics are not imported
- You can select the new template to use in generating a document report, either on the Generate Documentation dialog or in a Master Document or Model Document element

### Learn more

- [Generate Documentation](#)  <sup>[2642]</sup>
- [Master Document](#)  <sup>[2672]</sup>
- [Model Document](#)  <sup>[2673]</sup>

### Learning Center topics

- Alt+F1 | Enterprise Architect | Reporting | RTF Template | Importing a Company Template**




## 19.1.3.8 Template Fragments

Within a document template it is possible to include one or more sub-templates, or **Template Fragments**. Using Template Fragments, you can include in your document reports information from **SQL Queries** and **Model Scripts**, as well as Project Browser data, providing a greater level of customization of the document output. You can also reuse the same Template Fragment in several different templates, so you can develop a library of common Template Fragments that can be used to build larger, more complex document reports.

A Template Fragment is created through the same process as a normal, full-document template. Each Template Fragment can have its own Document Options, such as the filtering and sorting applied.

You can also use, copy and edit a number of system-provided Template Fragments, which you can access from the *Document Generation > System Templates > Fragments* folder in the Resources window.




### Template Fragments - Process Overview

Step	Action	See also
1	Create the Template Fragments to be included in your template.	<a href="#">Creating a Template Fragment</a> 
2	Create a normal, full-document template that is to include your Template Fragments.	<a href="#">Design Custom Document Templates</a> 
3	Within this full-document template add references to your Template Fragments in the appropriate locations.	<a href="#">Adding Fragments to a Template</a> 

### Notes

- You cannot generate documentation directly from a Fragment; to use a Fragment it must be included in a template

### Learn more

- [Generate Documents](#) 
- [Document Templates](#) 
- [Resources](#) 
- [Enhanced Reporting Webinar](#) (Online Resource; speakers/headphones required, allow approx. 20 minutes for viewing)

### Learning Center topics








- Alt+F1 | Enterprise Architect | Reporting | RTF Template Fragments | Overview**

#### 19.1.3.8.1 Creating a Template Fragment

A Template Fragment can return data from either an SQL Query, Model Scripts or Project Browser data. The process of defining the content of a Template Fragment depends on which of these types of data you want your Template Fragment to return, although in all cases the process is very similar to creating a normal document template.

**Access** **Project Browser package context menu | Documentation | Rich Text Format (RTF) Report (F8) > Templates**

### Create a new Template Fragment

Step	Action	See also
1	Click on the <b>New</b> button. The New Document Template dialog displays.	<a href="#">Document Templates</a>  <sup>[2681]</sup>
2	In the <b>New Template</b> field, type the name for the Template Fragment.	
3	In the <b>Copy Template</b> field, click on the drop down arrow and select either: <ul style="list-style-type: none"> <li>• <b>None</b>, to create an empty Template Fragment to develop from scratch, or</li> <li>• The name of an existing Template Fragment to act as a base</li> </ul>	
4	In the <b>Template Group</b> field either: <ul style="list-style-type: none"> <li>• Click on the drop-down arrow and select an existing group from the list, or</li> <li>• Type a name to generate a new template group, or</li> <li>• Leave this field blank, to <b>not</b> place the Template Fragment in a group</li> </ul> <p>The Template Group determines the location of the Template Fragment in the Resources window.</p>	<a href="#">Resources</a>  <sup>[1173]</sup>
5	Select the <b>Mark Template as Fragment</b> checkbox.	
6	Click on the <b>OK</b> button. The Document Template Designer displays in the Diagram View.	<a href="#">Design Custom Document Templates</a>  <sup>[2684]</sup>
7	Define the contents of your Template Fragment; the actual steps you follow depend on the data type you want the Template Fragment to return: <ul style="list-style-type: none"> <li>• Normal Document Template model data - the Template Fragment operates in the same way as a normal template</li> <li>• Custom SQL - returns data from a custom SQL query</li> <li>• Custom Scripts - returns data from a Model Script</li> </ul>	<a href="#">Design Custom Document Templates</a>  <sup>[2684]</sup> <a href="#">Custom SQL Fragments</a>  <sup>[2712]</sup> <a href="#">Custom Script Fragments</a>  <sup>[2714]</sup>
8	Once you have created the Template Fragment, you incorporate it into a normal template (or into an existing Template Fragment, if you wish).	<a href="#">Adding Fragments to a Document Template</a>  <sup>[2718]</sup>

#### Notes

- As with normal document templates, Template Fragments can also be created and opened via the **Resources window**
- Template Fragments in the Resources window are held in a *Fragments* folder in each of the

*Document Generation > System Templates* and *Document Generation > User Templates* folders, and are indicated by a red **T** symbol

- If you have an existing template that you want to use as a Template Fragment, open the template in the document editor and select the **Template Fragment** checkbox in the Document Options dialog
- If you embed a Template Fragment within another Fragment, report generation from the parent template will be slower

#### Learn more

- [Template Fragments](#) <sup>[2708]</sup>
- [Resources](#) <sup>[1173]</sup>
- [Document Options](#) <sup>[2654]</sup>
- [Custom Query Fragments](#) <sup>[2711]</sup>

#### Learning Center topics

- **Alt+F1 | Enterprise Architect | Reporting | RTF Template Fragments |**
  - **Create a SQL based Fragment 1**
  - **Create a SQL based Fragment 2**
  - **Create a Script Fragment 1**
  - **Create a Script Fragment 2**
  - **Create a Filtered Fragment 1**
  - **Create a Filtered Fragment 2**
  - **Combine Filtered Fragments**

#### **19.1.3.8.1.1 Custom Query Fragments**

You can create Template Fragments to return data from SQL Queries, or from Model Scripts. The first step in defining such Fragments is to use the **Custom Query** settings to enter either your SQL Search or a starting point for your Model Script.

**Access** **Tools | Document Template Designer: <template> > Content | File | Document Options > Custom Query**

#### Custom Query Options

Option	Action	See also
<b>Custom SQL</b>	Click on this radio button to enter an SQL Query.  You can then add fields within the <i>Custom</i> section of your Template Fragment, to refer to the data returned by the SQL Query.	<a href="#">Custom SQL Fragments</a> <sup>[2712]</sup>
<b>Custom Script</b>	Click on this radio button to display a field in which you specify a Model Script; click on the drop-down arrow and select the name of the Script that will be used by your Template Fragment.  You can then add fields within the <i>Custom</i> section of your Template	<a href="#">Custom Script Fragments</a> <sup>[2714]</sup>

Option	Action	See also
	Fragment that refer to the data returned by the Script.	

### Notes

- Custom Queries and the *Custom* template section can only be used in Template Fragments; they cannot be used in normal, full-document templates
- Custom Queries can reference the ID of the element or Package currently being processed by the document template, using the **#OBJECTID#** or **#PACKAGEID#** macros
- **#Branch#** gets the IDs of the child Packages of the current Package being processed, working recursively down to the lowest level of sub-Package; this is only valid when processing a Package - elements return a **0** value
- **#UserName#** gets the name of the user logged into version control
- **#Author#** takes the user name from the **Author** field in the General page of the Options dialog, so the defined search can be performed on objects created by that user (this value can be manually re-set in the Options dialog)
- **#Package#** gets the **package\_ID** of the currently-selected Package

### Learn more



- [Document Options](#)  <sup>2654</sup>
- [Template Fragments](#)  <sup>2708</sup>
- [Creating a Template Fragment](#)  <sup>2709</sup>

### Learning Center topics

- **Alt+F1 | Enterprise Architect | Reporting | RTF Template Fragments |**
  - **Create a SQL based Fragment 1**
  - **Create a SQL based Fragment 2**
  - **Create a Script Fragment 1**
  - **Create a Script Fragment 2**

You can create a Template Fragment to return data from an SQL Query. You do this by defining an SQL Query in the Document Options of the Template Fragment, then adding fields within the *Custom* section of your Template Fragment that refer to the columns returned by the SQL Query.

### Create an SQL Query Fragment

Step	Action	See also
1	Open or create a Template Fragment in the Document Template Editor.	<a href="#">Document Templates</a>  <sup>2687</sup> <a href="#">Creating a Template Fragment</a>  <sup>2709</sup>



Step	Action	See also
2	In the Document Template Editor, right-click on the Template background and select <b>File   Document Options</b> . The Document Options dialog displays.	
3	Select the Custom Query tab.	
4	In the <b>Template Fragment type</b> panel, select the <b>Custom SQL</b> radio button.	
5	In the main text field on the tab, type the SQL Query that is to be run on your model. This Query must return one or more columns; for example:  <pre>SELECT DocName AS TemplateName, Author AS TemplateLocation FROM t_document WHERE DocType = 'SSDOCSTYLE' and (IsActive = 1)</pre> <p>The SQL Query can also reference the ID of the Element or Package currently being processed by the document template, using the #OBJECTID# or #PACKAGEID# macros; for example:</p> <pre>SELECT Count ( Object_Type) AS CountOfActors FROM t_object WHERE Object_Type = "Actor" and Package_ID = #PACKAGEID#</pre>	
6	Click on the <b>OK</b> button to close the Document Options dialog.	
7	In the Document Template Editor, in the Sections panel, select the <b>Custom</b> checkbox to generate the <i>Custom</i> section.	<a href="#">Setting Sections for Reporting</a> <sup>[2688]</sup>
8	Within the <i>Custom</i> section, right-click and select the <b>Insert Custom Field</b> context menu option. A prompt displays for the name of the field to create.	
9	In response to the prompt, type the name of the column that is being returned by your Query; for example, <b>CountOfActors</b> . Click on the <b>OK</b> button.	
10	Repeat steps 8 and 9 for each column that you want to include in your report. Add any other formatting and content you need, to the Template Fragment.	
11	Save the Template Fragment, and add it to a normal document template.	<a href="#">Adding</a>

Step	Action	See also
		<a href="#">Fragments to a Document Template</a> <sup>[2718]</sup>

### Notes

- An Element Filter will not apply to the *Custom* section in a Template Fragment
- You can test your SQL Query using the SQL Editor in the Model Search window

### Learn more

- [Template Fragments](#)<sup>[2708]</sup>
- [Custom Query Fragments](#)<sup>[2711]</sup>
- [Create Search Definitions](#)<sup>[711]</sup>

### Learning Center topics

- **Alt+F1 | Enterprise Architect | Reporting | RTF Template Fragments |**
  - **Create a SQL based Fragment 1**
  - **Create a SQL based Fragment 2**

You can create a Template Fragment to return data from a Model Script. You do this by creating a Script that returns an XML string in a specific format. You add a reference to this Script in the Template Fragment Document Options, then add fields within the *Custom* section of your Template Fragment that refer to the data returned by the Script.

### Creating a Script Fragment

Step	Action	See also
1	Using the Scripting window, create a Model Script that returns a string in XML format.  Examples are available of: <ul style="list-style-type: none"> <li>• A Script that returns data in the appropriate XML format</li> <li>• The XML format that the Script must return so that it can be used by a Template Fragment</li> </ul>	<a href="#">Scripting</a> <sup>[2791]</sup> <a href="#">Example Template Fragment Script</a> <sup>[2718]</sup> <a href="#">Example Output of a Template Fragment Script</a> <sup>[2711]</sup>
2	Open or create a Template Fragment in the Document Template Editor. Right click on the Template background and select <b>File   Document Options</b> . The Document Options dialog displays.	<a href="#">Document Templates</a> <sup>[2681]</sup> <a href="#">Creating a Template Fragment</a> <sup>[2709]</sup>

Step	Action	See also
3	<p>Select the Custom Query tab and, in the Template Fragment type panel, select the <b>Custom Script</b> option.</p> <p>A blank field displays to the right of the options.</p>	
4	Click on the drop-down arrow and select your Script from the list.	
5	<p>In the main text field on the tab, type the entry point of your Script that returns the XML data.</p> <p>For example, if you were using the example script mentioned in step 1, then the entry point would be:</p> <pre>MyRtfData( #OBJECTID# );</pre> <p>To convey <b>Package</b> information, you can replace #OBJECTID# with #PACKAGEID#.</p>	
6	Click on the <b>OK</b> button to close the Document Options dialog.	
7	In the Document Template Editor, in the Sections panel, select the <b>Custom</b> checkbox to generate the <i>Custom</i> section.	<a href="#">Setting Sections for Reporting</a> <sup>[2688]</sup>
8	<p>Within the <i>Custom</i> section, right-click and select the <b>Insert Custom Field</b> context menu option.</p> <p>A prompt displays for the name of the field to create.</p>	
9	<p>In response to the prompt, type the name of the data field that is being returned by your script.</p> <p>Referring to the example from step 1, you would add <b>DateGen</b>.</p> <p>Click on the <b>OK</b> button.</p>	
10	<p>Repeat steps 8 and 9 for each data field that you want to include in your report.</p> <p>Add any other formatting and content you need, to the Template Fragment.</p>	
11	Save the Template Fragment, and add it to a normal Document Template.	<a href="#">Adding Fragments to a Document Template</a> <sup>[2718]</sup>

**Notes**

- Scripts have a limit of 60 seconds per call before control is returned to the Document Generator

Learn more

- [Template Fragments](#)<sup>[2708]</sup>
- [Custom Query Fragments](#)<sup>[2711]</sup>

Learning Center topics

- **Alt+F1 | Enterprise Architect | Reporting | RTF Template Fragments |**
  - **Create a Script Fragment 1**
  - **Create a Script Fragment 2**

The following JavaScript returns a row of data in XML format. This XML data can be returned in a document report using a **Custom Script** Template Fragment.

This script will also display the XML data in the System Output window, so you can easily check the output without running a document report.

```
! I N C Local Scripts.EAConstants-JScript

/*
 * Script Name: Document Properties
 * Author: Tom O'Reilly
 * Purpose: To display the properties of a document
 * Date: 02/02/2013
 */

function MyRtfData(objectID)
{
    var i;
    var xml DOM = new ActiveXObject( "MSXML2.DOMDocument.4.0" );
    xml DOM.validateOnParse = false;
    xml DOM.async = false;

    var node = xml DOM.createProcessingInstruction("xml", "version='1.0'
encoding='ISO-8859-1'");
    xml DOM.appendChild(node);

    var xml Root = xml DOM.createElement( "EADATA" );
    xml DOM.appendChild(xml Root);

    var xml DataSet = xml DOM.createElement( "Dataset_0" );
    xml Root.appendChild(xml DataSet);

    var xml Data = xml DOM.createElement( "Data" );
    xml DataSet.appendChild(xml Data);

    var xml Row = xml DOM.createElement( "Row" );
    xml Data.appendChild(xml Row);

    var xml Name = xml DOM.createElement( "DateGen" );
    var today = new Date();
    var dd = today.getDate();
    var mm = today.getMonth()+1; //January is 0!

    var yyyy = today.getFullYear();
    if(dd<10){dd='0'+dd} if(mm<10){mm='0'+mm} today = mm+'/' +dd+'/' +yyyy;

    xml Name.text = today;
    xml Row.appendChild(xml Name);
}
```

```

var xml Name = xml DOM.createElement( "TimeGen" );
var currentTime = new Date();
var hours = currentTime.getHours();
var minutes = currentTime.getMinutes();
if ( minutes < 10 )
{
    minutes = "0" + minutes;
}
xml Name.text = hours + ":" + minutes + " ";
xml Row.appendChild( xml Name );

var xml Name = xml DOM.createElement( "Author" );
xml Name.text = Session.UserName;
xml Row.appendChild( xml Name );

var xml Name = xml DOM.createElement( "EA_REPOS" );
xml Name.text = Repository.ConnectionString;
xml Row.appendChild( xml Name );

return xml DOM.xml;
};

Session.Output( MyRtfData(439) );

```

### Learn more

- [Scripting](#)<sup>[2791]</sup>
- [Template Fragments](#)<sup>[2708]</sup>
- [Creating a Template Fragment](#)<sup>[2709]</sup>
- [Custom Script Fragments](#)<sup>[2714]</sup>
- [The System Output Window](#)<sup>[169]</sup>

You can create a Template Fragment to return data from a Model Script into a document report. The Script must return one or more rows of data in the following XML format:

```

<EADATA>
  <Dataset_0>
    <Data>
      <Row>
        <DateGen>09/01/2013</DateGen>
        <TimeGen>9:31</TimeGen>
        <Author>John</Author>
        <EA_REPOS>C:\Users\John\Documents\Sample Model.eap</EA_REPOS>
      </Row>
    </Data>
  </Dataset_0>
</EADATA>

```

Each of these data fields in the XML (such as **DateGen**, **TimeGen**) refers to fields that can be included in the *Custom* section of a Template Fragment.

### Notes

- If the text is a formatted Note, add `formatted="1"` to the field row; for example: `<Author formatted="1"><i>John</i></Author>`

### Learn more

- [Scripting](#)  <sup>[2791]</sup>
- [Template Fragments](#)  <sup>[2708]</sup>
- [Creating a Template Fragment](#)  <sup>[2709]</sup>
- [Custom Script Fragments](#)  <sup>[2714]</sup>

#### 19.1.3.8.2 Adding Fragments to a Document Template

A **Template Fragment** is a template that can be used as a smaller sub-section of a normal document template. You do not generate a report from a Template Fragment on its own; instead you use a normal template that includes **references** to one or more Template Fragments.




Fragments can be called into any of the following sections in the template:

- Package
- Element
- Attribute
- Method

When generating a document report, as the report generator processes each section in the template, if it encounters a reference to a Template Fragment it will process that Fragment, inserting its output in the appropriate location of the generated report.

**Access** [Project Browser package context menu | Documentation | Rich Text Format \(RTF\) Report \(F8\) > Templates](#)

#### Incorporating Template Fragments into a Document Template

Step	Action	See also
1	Create the Template Fragment(s) to be included in your document template, or identify system Template Fragments to use.	<a href="#">Creating a Template Fragment</a>  <sup>[2709]</sup>
2	Create a new document template, or open an existing template in the document editor.	<a href="#">Document Templates</a>  <sup>[2681]</sup>
3	In the Sections panel, select the checkbox against each section that will refer to the Template Fragment(s).	<a href="#">Setting Sections for Reporting</a>  <sup>[2688]</sup>
4	In the Content panel, at the appropriate point within one of these sections, right-click and select the <b>Insert Template</b> context menu option.  The Select Template dialog displays.	
5	Click on the drop-down arrow and select the Template Fragment to insert.	
6	Click on the <b>OK</b> button.	

Step	Action	See also
	The tag <b>{Template - &lt;template name&gt;}</b> is inserted at the current cursor position.	
<b>7</b>	Repeat steps 4 to 6 for each Fragment you want to insert into the document template.  Complete any other work you need to do on the template, then save it and use it to run reports.	

### Editing or Switching Template Fragments

**Access** Right-click on the red **Template** tag > <required option>

Option	Action	See also
<b>View / Edit Fragment</b>	Opens the selected Template Fragment in another instance of the Document Template editor.  If you have selected a System Fragment, you can view the Fragment but not edit it. If you have selected a Model (user-created) Fragment, you can edit the Fragment.	<a href="#">Design Custom Document Templates</a> <sup>[2684]</sup>
<b>Override Fragment</b>	(Option available for a selected System Fragment)  Opens another instance of the Document Template Editor, where the System Fragment is duplicated as a Model (user created) Fragment with the same name. You can edit this Fragment and save it, and the original template will apply that new Model fragment instead.	
<b>Switch Fragment</b>	Displays the <b>Select Template</b> dialog; click on the drop-down arrow and select a different Template Fragment.	

### Notes

- You can also add a Template Fragment to a document template by locating the Fragment in the *Document Generation* folder in the **Resources window** and dragging the Fragment into the Document Editor; Template Fragments are held in the *Fragments* folders within the *System Templates* and *User Templates* folders
- It is possible to embed a Template Fragment within another Fragment; however, this will slow down report generation

### Learn more

- [Template Fragments](#) <sup>[2708]</sup>
- [Resources](#) <sup>[1173]</sup>

Learning Center topics

- **Alt+F1 | Enterprise Architect | Reporting | RTF Template Fragments |**
  - **Add a SQL Fragment to Template**
  - **Add Script Fragment to Template**

**19.1.4 The Legacy RTF Report Generator**

Creating a Rich Text Format (RTF) document is a simple and flexible process. An RTF document is based on a Package or an element in your project (more usually a Package). To produce a document, you select the Package or element to report on in the Project Browser, Package Browser, Diagram List or Model Search, then press **F8** to display the Generate Documentation dialog. On the Options tab, click on the **Switch Generator** button to access the legacy Rich Text Format Report dialog.

The Rich Text Format Report dialog enables you to set the exact contents and look and feel of your report. You enter the file name of the report, a heading, additional notes, template name (for saving the set-up) and other options. You can also select the style of the report; either plain or formal.

Optionally, you can set up a filter, the details to include, element types to exclude, whether to process child packages, whether to show diagrams and the diagram format.

You can switch back to the Generate Documentation dialog by clicking on the **Switch RTF Generator** button.

Notes

- The Legacy Generator is available if you have RTF templates created in releases of Enterprise Architect prior to 4.1, and you prefer to generate RTF reports using the original generator; however, as you can generate reports from these templates using the post-Enterprise Architect 4.1 RTF Generator, the Legacy Generator and instructions for its use are no longer updated
- Reports produced using the Legacy RTF Generator do reflect the Notes formatting feature in any text associated with elements
- The Rich Text Format Report dialog panels are individually described in the subsequent topics of this section (listed below); the dialog has a lot of options - you can use them selectively to produce output at the level of detail suited to your project

Learn more

- [Document a Single Element](#)<sup>[2721]</sup>
- [Set the Main RTF Properties](#)<sup>[2721]</sup>
- [Apply a Filter](#)<sup>[2722]</sup>
- [Exclude Elements](#)<sup>[2722]</sup>
- [RTF Diagram Format](#)<sup>[2723]</sup>
- [Model Include](#)<sup>[2723]</sup>
- [RTF Report Options](#)<sup>[2723]</sup>
- [RTF Report Selections](#)<sup>[2724]</sup>
- [Generate the Report](#)<sup>[2726]</sup>
- [Diagrams Only Report](#)<sup>[2741]</sup>
- [Report Templates](#)<sup>[2726]</sup>
- [Include or Exclude a Package from Report](#)<sup>[2761]</sup>
- [Save as Document](#)<sup>[2728]</sup>



- [Notes Toolbar](#)<sup>[1143]</sup>

#### 19.1.4.1 Document a Single Element

RTF documentation can also be generated for a single element.

**Access** Click on **element** | **Element** | **Advanced** | **Rich Text Format (RTF) Report**

Click on the Options tab and the **Switch Generator** button to display the Rich Text Format Report dialog.

#### Learn more

- [Generate Documentation](#)<sup>[2644]</sup>
- [The Legacy RTF Report Generator](#)<sup>[2720]</sup>

#### 19.1.4.2 Set the Main RTF Properties

The main section of the Rich Text Format Report dialog enables you to set the output location and appearance of the final RTF document.

#### Set Output Location and Appearance

Step	Action	See also
1	Open the Rich Text Format Report dialog.	<a href="#">The Legacy RTF Report Generator</a> <sup>[2720]</sup>
2	Supply an <b>Output Filename</b> to save the report into; always include the extension . RTF as part of the filename.	
3	In the <b>Template Name</b> field, provide the name of the template to generate the report from.	
4	Select a report <b>Style</b> : Formal or Basic.	
5	Type a <b>Heading</b> for your report; this appears as the first heading item in your output.	
6	Select your required <b>Heading Style</b> and <b>Initial Heading Level Indent</b> from the drop-down lists.	

#### Notes

- It is recommended that you enter a full path name for your report; the images in your report are saved externally in an images directory, and supplying the full directory path avoids confusion over the location of these images

- If you move your report you must also move the images directory

### 19.1.4.3 Apply a Filter

You can apply a filter on the Rich Text Format Report dialog to include or exclude elements by date modified, phase or status. This helps to track changes and break a document into multiple delivery phases.

#### Apply Filter

Topic	Detail	See also
<b>Usage</b>	<p>Open the Rich Text Format Report dialog.</p> <ul style="list-style-type: none"> <li>• To enable the date filter, select the checkbox in the <b>Date</b> field</li> <li>• In the first two <b>Only include objects</b> fields, click on the drop-down arrows and select the appropriate criteria (<b>Modified/Created, Before/After</b>)</li> <li>• The package phase filter applies at the package level (not the element level) and ignores the phase of the root package that you are documenting; to enable the phase filter, in the <b>Where Package Phase</b> field click on the drop-down arrow and select an operator</li> </ul> <p>Enterprise Architect filters out all packages that do not meet the selection criteria; All elements in the package are ignored, regardless of their individual phase</p> <ul style="list-style-type: none"> <li>• The element status filter enables you to limit the output by element status; unlike the package phase filter, this filter applies to every element</li> </ul> <p>You can filter against a status of like or <i>unlike</i> a criterion, for example, <i>like proposed</i>, or against the <i>in</i> and <i>not in</i> operators, such as <i>in approved</i>, <i>not in validated</i>; when using the <i>in</i> and <i>not in</i> operators, enter a comma-separated list of status types as your criteria expression</p>	<a href="#">The Legacy RTF Report Generator</a> <small>[2720]</small>

### 19.1.4.4 Exclude Elements

Using the Rich Text Format Report dialog, you can exclude elements of any type from your final output; this is useful when you want to highlight particular items and not clutter up a report with too much detail.

#### Exclude Elements

Topic	Detail	See also
<b>Usage</b>	<p>Open the Rich Text Format Report dialog.</p> <p>Click on each element to exclude, or click on the <b>All</b> button to exclude all elements.</p> <p>Click on the <b>None</b> button to clear your selections.</p>	<a href="#">The Legacy RTF Report Generator</a> <small>[2720]</small>

#### 19.1.4.5 RTF Diagram Format

It is possible to output diagrams to Bitmap files, GIF files or Windows Metafiles.

##### Set Format

Topic	Detail	See also
<b>Usage</b>	<p>Open the Rich Text Format Report dialog.</p> <p>In the Diagram format panel (bottom center of the dialog) select the required format for the report.</p> <p>Generally the two metafile options (EMF and WMF) are recommended; however, there are times that the others might be suitable.</p>	<a href="#">The Legacy RTF Report Generator</a> <sup>[2720]</sup>

#### 19.1.4.6 Model Include

##### Include model items in report

Topic	Detail	See also
<b>Usage</b>	<p>The Model Include panel of the Rich Text Format Report dialog has the following options:</p> <ul style="list-style-type: none"> <li>• <b>Glossary</b> to include the project glossary</li> <li>• <b>Tasks</b> to include project tasks</li> <li>• <b>Issues</b> to include project issues</li> </ul> <p>Select the appropriate checkbox to include the items in the generated RTF documentation.</p>	<a href="#">Project Glossary</a> <sup>[533]</sup> <a href="#">Project Tasks</a> <sup>[526]</sup> <a href="#">Project Issues</a> <sup>[528]</sup>

#### 19.1.4.7 RTF Report Options

Additional RTF report options you can select from the Options panel on the Rich Text Format Report dialog are shown below.

##### Options

Topic	Detail	See also
<b>Checkboxes</b>	<p>You can select whether or not to recursively document packages, show diagrams or add a page break before each new package. Select the:</p> <ul style="list-style-type: none"> <li>• <b>Process all Children</b> checkbox to recursively process all child packages within the main package</li> <li>• <b>Show Diagrams</b> checkbox to include diagrams in your document; clear this item for no diagrams</li> <li>• <b>New page per package</b> checkbox to force a page break on each new package (excepting empty packages)</li> </ul>	<a href="#">Setting the Main RTF Properties</a> <sup>[2721]</sup>

Topic	Detail	See also
	<ul style="list-style-type: none"> <li>• <b>Document all elements</b> checkbox to include all elements included in the project</li> <li>• <b>Document Packages</b> checkbox to document the package as an element in addition to the documentation that would normally be produced for package documentation</li> <li>• <b>Hide 'note-less' elements</b> checkbox to exclude all elements without notes from the documentation</li> <li>• <b>Embed Diagrams in Document</b> checkbox to ensure that the diagram images are contained within the RTF document rather than stored in a linked external file</li> <li>• <b>Skip root package</b> checkbox to exclude the parent package from the documentation and include only the child packages</li> <li>• <b>Document Linked Elements</b> checkbox to include the object details for linked elements that do not originate from the selected package</li> <li>• <b>Use Heading styles for Details</b> checkbox to ensure that the details are formatted as heading styles rather than formatted text; this option is only available when the <b>Heading Style</b> field in the Main section of the Rich Text Format Report dialog is set to <b>Max 9 levels</b> - elements are package + 1.</li> </ul>	

#### 19.1.4.8 RTF Report Selections

Using the **For each Object Include** section of the Rich Text Format Report dialog you can select the documentation sections to include in your report.

##### Select Document Sections

Topic	Detail	See also
<b>Usage</b>	<p>What you include or exclude governs how simple or detailed your report is. You can create multiple reports at different levels of detail for different audiences. Experiment with these options to see what effect inclusion or exclusion has. Most items are self-explanatory.</p> <p>Selecting the checkbox against a category item in the list selects all of the options that are contained in the category. To expand a category, click on the + symbol next to the category name. To exercise greater control over a category of options expand the top level item and then select the required individual items from the list.</p> <p>Sometimes an item applies only to a certain type of element; for example, <i>Attributes</i> only applies to Class elements and a few other element types. The <i>Child Diagrams</i> option shows or hides any diagrams that are attached under a model element; for example, a Use Case might have a Scenario diagram attached.</p>	

##### Notes

- Use this feature to produce the right level of detail for your audience; technical readers might want to see everything, whilst management might require only the general outline

### 19.1.4.9 Custom Language Settings

If you export RTF-format documents from Enterprise Architect in languages other than English, you can customize the codepage, default language ID and character set that Enterprise Architect uses when generating RTF. This makes it much easier to generate documentation appropriate to your country or locale.

You can also set up a list of word substitutions. For instance, where Enterprise Architect would include the word *Figure*, you can specify another word to replace it that is either in your language or more meaningful to your readers.

#### Set up Substitutions

Step	Action	See also
1	Open the <b>Rich Text Format Report</b> dialog.	<a href="#">The Legacy RTF Report Generator</a> [2720]
2	In the Language panel (bottom left of dialog) click on the <b>Adjust</b> button. The <b>Customize RTF Language</b> dialog displays.	
3	Double-click on an item to set or clear its <b>Substitute</b> word.	
4	When you have finished, click on the <b>OK</b> button.	

#### Set Up Codepage and Character Set

Step	Action	See also
1	From the drop-down lists in the <b>Language</b> , <b>Codepage</b> and <b>Charset</b> fields, select the language, codepage and character set that most closely match your location.	
2	If required, modify the <b>Substitute Tags</b> by double-clicking on each and manually setting the value (for advanced use only).	
3	To clear the substitution list, double-click on each item in turn and delete the substitute value.	
4	When you have completed the settings, click on the <b>OK</b> button to save them. Now when you generate RTF documents, the substitute tags are used in the output.	

**Notes**

- If you want to generate reports in a language that requires Unicode support, please note that Enterprise Architect .EAP files default to using **JET 3.5** as the database engine, which does not support Unicode character sets (see the *Getting Started > Basics* topic) ; to provide Unicode support in your project, either upsize to a **DBMS repository** or set **JET 4.0** as the database engine, and download a copy of the **Jet 4.0 EABase** model from the Sparx Systems website and do an **EAP to EAP transfer** of your model into the Jet 4.0 file
- You can transport these language and tag definitions between models, using the **Export Reference Data** and **Import Reference Data** options on the **Project | Model Import/Export** menu

**Learn more**

- [Basics](#) <sup>[43]</sup>
- [Server Based Repositories](#) <sup>[214]</sup>
- [Perform a Project Data Transfer](#) <sup>[504]</sup>
- [Export Reference Data](#) <sup>[376]</sup>
- [Import Reference Data](#) <sup>[380]</sup>

**19.1.4.10 Generate the Report****Generate**

Topic	Detail	See also
<b>Usage</b>	<p>Once you have set up the document properties as required, click on the <b>Generate</b> button to generate the report.</p> <p>When you have generated the document, click on the <b>View</b> button to open the report in MS Word.</p>	

**19.1.4.11 Legacy Report Style Templates**

Using the **Legacy RTF Style Editor** you can edit the RTF associated with various sections of the RTF Report facility in Enterprise Architect. You would typically use this functionality to customize a report's look and feel for your company or client.

If you have previously defined and saved a template, click on the **Load** button on the Rich Text Format Report dialog to open the list of defined templates. Select one in order to load it as the current template; all the features saved become the current features. This enables you to define a set of standard report types that streamline document production.


**Access**   **Project | Resources | Document Generation | Legacy Templates**

**Create or Edit RTF Style Templates**

Step	Action	See also
<b>1</b>	Expand the <i>Legacy Templates</i> folder.	

Step	Action	See also
2	To edit an <b>existing</b> Legacy template, expand the <i>Legacy Templates</i> tree and double-click on the template name, or right-click and select the <b>Modify Document Template</b> context menu option. The RTF Style Editor displays.	
3	Alternatively, to create a <b>new</b> Legacy template, right-click on <i>Legacy Templates</i> and select the <b>Create RTF Style Template (Legacy)</b> context menu option.  A prompt displays for the new template name.	
4	Type the name of the new template and click on the <b>OK</b> button.  The RTF Style Editor displays.	

### RTF Fragments

Topic	Detail	See also
<b>RTF Style Editor</b>	<p>The RTF Style Editor contains a list of all available RTF fragments for modification and customization.</p> <p>Each fragment typically contains RTF plus one or more special tag names that Enterprise Architect replaces with information during generation. Currently you cannot alter the content within the tag names, but you can omit a complete tag by removing it, or alter its basic display properties in the surrounding RTF.</p> <p>Special tag names are delimited by # characters; for example, <b>#NOTES#</b></p> <p>Click on the:</p> <ul style="list-style-type: none"> <li>• <b>Get Default</b> button to retrieve the default Enterprise Architect template for the currently-selected template item in the left hand list</li> <li>• <b>Save</b> button to save the version of the template for this style only</li> <li>• <b>Delete</b> button to remove your modified version of the template, which causes Enterprise Architect to use the default template during report generation</li> </ul> <p>To select a template during report generation, click on the <b>Style</b> drop-down arrow on the Rich Text Format Report dialog. Once a style is selected, Enterprise Architect applies that to the current report. Select &lt;Basic&gt; for the inbuilt style.</p>	<a href="#">The Legacy RTF Report Generator</a>  <sup>2720</sup>

### Notes

- The RTF Style Editor discussed here automatically displays when you modify or create a Legacy RTF template. If you select a template created in the more recent Document Template Editor, that editor opens automatically instead
- You can transport these RTF templates between models, using the **Export Reference Data** and **Import Reference Data** options on the **Project | Model Import/Export** menu
- To delete a template, right-click on it and select the **Delete Document Template** context menu option
- You can also alter the custom language settings

#### Learn more

- [Design Custom Document Templates](#) <sup>[2684]</sup>
- [Export Reference Data](#) <sup>[376]</sup>
- [Import Reference Data](#) <sup>[380]</sup>
- [Custom Language Settings](#) <sup>[2725]</sup>

### 19.1.4.12 Save as Document

The *Document* feature enables a particular documentation configuration to be 'remembered', linking the loaded template within the Report dialog to the current highlighted package. If a particular template is always used with a specific package, and multiple cases of documentation exist to be propagated, saving these as Documents can ease document generation later.

#### Create and use Documents

Step	Action	See also
1	Open the Rich Text Format Report dialog.	<a href="#">The Legacy Report Generator</a> <sup>[2720]</sup>
2	Click on the <b>Save as Document</b> button. The Save current as document definition dialog displays:	
3	In the <b>Enter Value</b> field, type a name for the document and click on the <b>OK</b> button. The document is added to the Resources window for easy future access.	
4	To generate documentation from the Resources window, right-click on the required document. The context menu displays.	
5	Select the required context menu option. <ul style="list-style-type: none"> <li>• <b>Open Document</b> - Opens the corresponding document file, as specified by the template Filename property</li> <li>• <b>Generate Document</b> - Opens the Report dialog, loaded with the specified template</li> <li>• <b>Auto Generate Document</b> - Generates documentation, with the document located at the path specified by the template's Filename property</li> </ul>	



Step	Action	See also
	<ul style="list-style-type: none"> <li>• <b>Delete Document</b> - Removes the specified document</li> </ul>	

#### 19.1.4.13 Use Microsoft Word

When generating reports with the 'legacy' Report Generator, you can use Microsoft Word™ to easily incorporate additional features in the reports. You can:

- Create a custom master document combining separate reports, to add a common table of contents, table of figures, headers and footers
- Create documents with sustainable links to generated 'pieces' of Enterprise Architect output, pre-divided using bookmarks, and updated automatically when the document is opened
- Open a report in Word
- Change linked images to embedded images

#### Notes

- With the more recent, 'enhanced' Document Report Generator, many of the facilities provided by Word have been incorporated into the Generator or otherwise rendered unnecessary; it is therefore likely that you would have no need to use Word in generating your reports - you can use Virtual Documents, for example, instead of Word master documents and Bookmarks
- These topics on the use of Word are therefore included only to support users who are still working with the 'Legacy' Report Generator, or who are using Word for personal preference
- When you develop a report using Word with Enterprise Architect, leave definition of the section styles in the Editor to the final stages just prior to report generation; Word truncates the section bookmarks, as it uses a smaller field length for sections
- In Word, you can review and edit reports generated by Enterprise Architect, but you cannot import them back into Enterprise Architect without damaging the section style definition

#### Learn more

- [The Legacy Report Generator](#)<sup>[2720]</sup>
- [Bookmarks](#)<sup>[2730]</sup>
- [Virtual Documents](#)<sup>[2669]</sup>
- [Open a Report in Microsoft Word](#)<sup>[2729]</sup>
- [Change Linked Images to Embedded](#)<sup>[2730]</sup>
- [Features of Word](#)<sup>[2732]</sup>

##### 19.1.4.13.1 Open a Report in Microsoft Word

If you want to use Microsoft Word™ to open a document file generated in Enterprise Architect, simply load Word and open the file as a normal document. Word converts the file for display.

If Word is your default handler of document files, you can also load up and view the report by either:

- Double-clicking on the output file, or
- Clicking on the **View** button on the Generate Documentation dialog

### 19.1.4.13.2 Change Linked Images to Embedded

One of the options available when generating document reports is to embed images in the document, instead of incorporating links to the image files. You would embed images if you want the document to be portable (separated from the directory containing the image files), but use links to the image files if you want to update the images without regenerating the document. You would be more likely to link to image files when developing the model and documentation, and more likely to embed images when the document was ready for review or use.

If you have generated the document report with linked images and imported the document into Word™, and you want to distribute that document, you can convert the file links to embedded images without having to go back to Enterprise Architect to regenerate the document with embedded images.

#### Break image links in Word

Step	Action	See also
1	Open the required document in Word.	
2	Select the <b>Edit   Links</b> menu option.	
3	Highlight all links in the <b>Links</b> list.	
4	Select the <b>Save Picture in Document</b> checkbox.	
5	Click on the <b>Break Link</b> button.	
6	When prompted, click on the <b>Yes</b> button to break the links. Word breaks the links and saves copies of the images inside the document.	

### 19.1.4.13.3 Document Bookmarks

If you want to create a complex document that consists of **sections** of one or more document reports, rather than one **complete** report, you can create a master document in Microsoft Word™ and within it create links to the report sections using their **bookmarks**. Bookmarks are GUID-based numbers that are automatically created for packages, diagrams and elements, and placed in a document when it is generated. Every object is bookmarked in the document according to the following rules:

- All alphabetic and numeric characters remain the same
- All other characters (including spaces) are converted to underscores

For example *UC01: Use Case Model* becomes *UC01\_\_Use\_Case\_Model*. A Package bookmark applies from the beginning of a Package to the end, and includes all child Packages and elements underneath. You can also manually assign bookmarks if you want to have references to additional subdivisions of a model or document. If you change your model and regenerate your document report, you can simply refresh the links in the master document in Word to automatically update the Word document.

When you use bookmarks to add sections of a generated report to a Word master document, you work with Word and the Enterprise Architect Project Browser open at the same time.

#### Add a Bookmarked Section of an RTF Document to a Word Document

Step	Action	See also
1	Within the Word file, position the cursor at the point at which to insert the section of the document.	
2	Select the Word <b>Insert   File</b> menu option. The Insert File dialog displays.	
3	Browse for and select the document file to insert, then click on the <b>Range</b> button.	
4	In the Enterprise Architect Project Browser, right-click on the Package, diagram or element to include in the documentation. The context menu displays.	
5	To paste the object's bookmark into the clipboard, select the appropriate menu option: <ul style="list-style-type: none"> <li>• <b>Documentation   Copy Documentation Bookmark</b> (Package)</li> <li>• <b>Copy/Paste   Copy Documentation Bookmark</b> (element)</li> <li>• <b>Copy Documentation Bookmark</b> (diagram)</li> </ul>	
6	In Word, in the <b>Range</b> cell of the Insert File dialog, press <b>Ctrl+V</b> to paste the information from the clipboard.	
7	Click on the <b>OK</b> button.	
8	Click on the drop-down arrow next to the <b>Insert</b> button. A short menu displays containing two options: <ul style="list-style-type: none"> <li>• <b>Insert</b> embeds a permanent copy of the text</li> <li>• <b>Insert as Link</b> creates a link that can be updated if you alter the source document</li> </ul> Select the <b>Insert as Link</b> option.	
9	Repeat steps 1 to 8 for each section of each document you want to include in the Word document.	
10	Select the Word <b>Tools   Options</b> menu option and, on the Options dialog, select the General tab and select the <b>Update automatic links at Open</b> checkbox.	

Step	Action	See also
	Whenever the Word document is opened, the links to the document sections will be automatically updated with any changes made to the generated documents.	

### Notes

- You cannot use Bookmarking in **Master Document** elements, which effectively replace Bookmarking in Word

Bookmarking requires each bookmark to be unique; when you generate a report with a standard template (including in a single Model Document element), each bookmark is unique and there is a 1:1 association between the element details being generated and the elements in the repository

As Master Documents are intended to contain multiple sub-documents, the association ceases to be 1:1; there is no simple method that enables the generated data to be uniquely identified directly in association with the original element

### Learn more

- [Insert Reference Links](#)<sup>[1083]</sup> (manually insert bookmarks)
- [Create Master Document](#)<sup>[2672]</sup>

#### 19.1.4.13.4 Features of Word

When you have generated reports in the Enterprise Architect **'legacy'** report generator, you can open them in Microsoft Word™ to enhance the output of your project documentation. Some of the things you might do include:

- Add a table of contents
- Add a table of figures
- Add headers and footers
- Manipulate the report tables
- Set an option to refresh linked files automatically

### Learn more

- [Add Table of Contents](#)<sup>[2733]</sup>
- [Add Table of Figures](#)<sup>[2733]</sup>
- [Add Headers and Footers](#)<sup>[2734]</sup>
- [Manipulate Tables](#)<sup>[2734]</sup>
- [Refresh Links](#)<sup>[2735]</sup>

### Learning Center topics

- Alt+F1 | **Enterprise Architect | Reporting | Rich Text Reports | Report Baseline Difference**

#### 19.1.4.13.4.1 Add Table of Contents

Among the features of Microsoft Word™ that can be incorporated into generated Enterprise Architect document reports is a Table of Contents, which can be used to aid navigation of the documentation and enhance readability. The Table of Contents provides hyperlinks to the diagrams in the electronic version of the documentation, and page numbers in both the printed and electronic versions of the documentation.

##### Include Table of Contents in document

Step	Action	See also
1	In Word, open the report to which to add a Table of Contents.	
2	Select the <b>Insert   Reference   Index and Tables</b> menu option.	
3	Click on the Table of Contents tab to set the options that are available for setting up the Table of Contents.	

##### Notes

- The format of the Table of Contents is dependant on the heading levels created when the document is generated; to set the heading styles in the document, see the *RTF Report Options* topic

##### Learn more

- [RTF Report Options](#)  2723

#### 19.1.4.13.4.2 Add Table of Figures

Among the features of Microsoft Word™ that can be incorporated into generated Enterprise Architect reports is a Table of Figures, which can be used to aid navigation of the documentation and enhance readability. The Table of Figures provides hyperlinks to the diagrams in the electronic version of the documentation, and page numbers in both the printed and electronic versions of the documentation.

##### Include Table of Figures in document

Step	Action	See also
1	In Word, open the report to which to add a table of figures.	
2	Select the <b>Insert   Reference   Index and Tables</b> menu option.	
3	Click on the Table of Figures tab to set the options that are available for setting up the table of figures.	

#### 19.1.4.13.4.3 Add Headers and Footers

Among the features of Microsoft Word™ that can be incorporated into generated Enterprise Architect document reports are headers and footers.

##### Include headers and footers in the document report

Step	Action	See also
1	In Word, open the report to which to add headers and footers.	
2	Select the <b>View   Header and Footer</b> menu option, and enter the appropriate information into the header section and the footer section of the document.	

#### 19.1.4.13.4.4 Manipulate Tables in Word

In the document reports generated from the **Legacy** report generator in Enterprise Architect, tables are included when items such as attributes and methods are selected in the **For each Object** section of the Rich Text Format Report dialog. Microsoft Word™ offers several levels of customization for tables and can be used to enhance formatting the tables in situations where the margins of the table exceed the dimensions of the page size selected for printing in Word.

##### Manually resize the table

When the amount of detail for a documented item, such as an attribute or operation, exceeds the margins of the page in Word, you can manually resize the table in order to view all of the details.

Step	Action	See also												
1	Select the table that exceeds the margin size.													
2	<p>Mouse over the border of the table until the mouse pointer changes into the 'drag' icon, as shown below.</p> <div><p><i>Message Attributes</i></p><table><tr><th>Attribute</th><th>Type</th><th>Notes</th></tr><tr><td><u>sentTime</u></td><td>protected : <i>Date</i></td><td></td></tr><tr><td><u>receivedTime</u></td><td>protected : <i>Date</i></td><td></td></tr><tr><td>body</td><td>protected : <i>String</i></td><td></td></tr></table></div>	Attribute	Type	Notes	<u>sentTime</u>	protected : <i>Date</i>		<u>receivedTime</u>	protected : <i>Date</i>		body	protected : <i>String</i>		
Attribute	Type	Notes												
<u>sentTime</u>	protected : <i>Date</i>													
<u>receivedTime</u>	protected : <i>Date</i>													
body	protected : <i>String</i>													
3	Drag the cursor to the left to reduce the width of the table, and then select the <b>File   Print Preview</b> menu option to confirm that the table borders are within the page margins.													

Step	Action	See also
4	Resize all of the tables that overhang the margins of the page, using steps 1 to 3.	

### **Apply styles to tables**

One of the customizable properties of Word when working with tables is the ability to apply a style to a table, so that you can rapidly change the appearance of the table.

Step	Action	See also
1	In Word, open the report in which to change the table styles.	
2	Locate and select the table for which to adjust the appearance.	
3	<p>Select the <b>Table   Table Auto Format</b> menu option.</p> <p>The Table Autoformat dialog displays.</p> <p>From here you can specify a predefined table style from the <b>Table styles</b> list, or create a new style by clicking on the <b>New</b> button. The table styles defined in the Table Autoformat dialog only apply to one table at a time so you must apply the style to each table individually.</p>	

#### **19.1.4.13.4.5 Refresh Links**

If you link into reports generated in Enterprise Architect, you can set an option in the Microsoft™ Word master document so that each time the document is opened, the links are refreshed and - if the reports have changed - the linked text is automatically updated.

**Access** **In Word, Tools | Options > General**

### **Refresh linked report text**

Step	Action	See also
1	Select the <b>Update automatic links at Open</b> checkbox.	
2	Click on the <b>OK</b> button.	

### 19.1.5 System Documents

As well as tailoring the output generated from standard or custom report templates, you can generate a number of **pre-set reports** that are each provided to meet a specific modeling requirement.

#### Pre-set Reports

Report	See also
<b>Testing Details Report</b> - extracts the test details for all elements in the selected Package hierarchy that have test items lodged against them.	<a href="#">Testing Details Report</a> <sup>[2736]</sup>
<b>Implementation Details Report</b> - lists, for a specified Package, the elements that require implementation, together with any source elements in Realize (Implements) relationships with those elements.	<a href="#">Implementation Details Report</a> <sup>[2737]</sup>
<b>Dependency Details Report</b> - shows a list of any elements that are dependent on (having a Dependency connector to) another element for their specification.	<a href="#">Dependency Details Report</a> <sup>[2738]</sup>
<b>Maintenance Report</b> - extracts the maintenance details for all elements in the selected Package hierarchy that have maintenance items lodged against them.	<a href="#">Maintenance Report</a> <sup>[2740]</sup>
<b>Diagrams Only Report</b> - lists only the diagrams from the target Package.	<a href="#">Diagrams Only Report</a> <sup>[2741]</sup>
<b>Resource Report</b> - shows how your resources are deployed in your project, displaying a list of all elements that have resources allocated to them.	<a href="#">Resource Report</a> <sup>[546]</sup>
<b>Testing Report</b> - outputs the test scripts and results you have entered against elements in the model.	<a href="#">Testing Report</a> <sup>[2742]</sup>

#### 19.1.5.1 Testing Details Report


As you manage the test program for your model, you can monitor the status of tests run on the elements within a Package hierarchy by generating a Testing Details report on the Package. You can filter the report to show tests of a certain type, having a specific status, run by a specific person and/or checked by a specific person.

Access **Project | QA Reports & Metrics > Testing Details**

#### Generate a Testing Details report

Field/Button	Action	See also
<b>Root Package</b>	Displays the name of the Package selected from the Project Browser. All elements and Packages under this package are included in the report.	



Field/Button	Action	See also
	If you want to change the Package, click on the new Package in the Project Browser and then click on the  icon at the end of the field. The replacement Package name displays in the field.	
<b>Run By</b>	Click on the drop-down arrow and select a name to filter for tests run by that person. Click on the <b>x</b> button to clear the field.	
<b>Checked By</b>	Click on the drop-down arrow and select a name to filter for tests checked by that person. Click on the <b>x</b> button to clear the field.	
<b>Test Type</b>	Select the radio button for the required test type - <b>Unit, Integration, System, Scenario, Acceptance</b> , or <b>All</b> types.	
<b>Status</b>	Select the radio button for the required test status - <b>Passed, Failed, Not Run</b> , or <b>All</b> statuses.	
<b>Refresh</b>	Click on this button to re-run the report query on the Package currently selected in the <b>Root Package</b> field.	
<b>Print</b>	Click on this button to print a summary of the test results.	

### Notes

- The **Testing Details** report provides a **summary** of the test status and who ran and checked the test; the **Testing** report is an alternative option, providing a **detailed** description of how each test executed, what the input and acceptance criteria were, and what the results were
- You can restructure the output of the report using the List Header facilities for reported information

### Learn more



- [Customizing the Search View](#)<sup>[709]</sup>
- [Working on Test Records](#)<sup>[2605]</sup>
- [Testing Report](#)<sup>[2742]</sup>
- [List Header](#)<sup>[677]</sup>

### 19.1.5.2 Implementation Details Report

Using the **Implementation report**, you can list all elements in a package (selected from the Project Browser) that **require implementors**, together with any source elements that are in **Realization** (or **Implements**) relationships with those elements.

**Access** [Project | QA Reports & Metrics > Implementation Details](#)

**Generate Implementation report**

Field/Button	Action	See also
<b>Root Package</b>	Displays the name of the root Package; all elements and packages under this Package are included in the report.  If you want to change the Package, click on the replacement in the Project Browser and click on the <b>Refresh List</b> button.	
<b>Set Target Types</b>	By default the Implementation report includes only a limited number of element types, such as Use Cases and Requirements.  If you want to include other element types, click on this button and select the additional types on the Implementation Targets dialog.	<a href="#">Implementation Targets Dialog</a>  <sup>[2739]</sup>
<b>Refresh List</b>	Click on this button to run the report and display the results in the Details panel.	
<b>Locate Object</b>	Click on this button to locate the selected <b>implemented</b> (or <b>realized</b> ) element in the Project Browser.	
<b>Show Unimplemented</b>	Select this checkbox to list all elements in the Package that can be implemented but currently don't have any other element to realize them (for example, a Use Case that has no Component or Class to implement the Use Case behavior).	
<b>Show Implemented</b>	Select this checkbox to list all elements in the Package that <b>are</b> implemented; that is, they have another element associated with them in a Realize relationship (for example, a Use Case that is implemented by a Component).	<a href="#">Realize</a>  <sup>[1440]</sup>
<b>Details</b>	Lists the implementation details, consisting of: <ul style="list-style-type: none"> <li>• The names of the elements in the current Package that can be implemented</li> <li>• The type of the elements that can be implemented</li> <li>• The type of the relationship on the elements (Realization)</li> <li>• The name of any element that the element is implemented by</li> <li>• The type of the implementor element</li> </ul>	
<b>Print</b>	Click on this button to print the Implementation report.	
<b>Save Report</b>	Click on this button to specify the file location into which to save the Implementation report. A prompt displays for the file path and name.	

### 19.1.5.2.1 Implementation Targets Dialog

When you generate an Implementation report, by default it includes only certain element types, such as Use Cases and Requirements. If you want to include other element types, you can add them to the report parameters using the Implementation Targets dialog.

**Access** [Project | Documentation | Implementation Details: Set Target Types](#)

#### Include other element types in report

Field/Button	Action	See also
<b>Targets - Active</b>	Lists the types of element that the Implementation report includes.  If you do not want to include a type of element in your report, double-click on it to remove it from the list. The element type is then displayed in the <b>Targets - Available</b> list.	
<b>Targets - Available</b>	Lists the types of element that <b>could</b> be included in the report but are currently not.  If you want to include any of these types of element in your report, double-click on each one to transfer it to the <b>Targets - Active</b> list.	
<b>Close</b>	Click on this button to close the Implementation Targets dialog and immediately refresh the Implementation report with details on the additional element types.	

### 19.1.5.3 Dependency Details Report

Using the **Dependency report**, you can list all elements in a Package (selected from the Project Browser) that are **dependent on another element** for their specification. For example, a Use Case derives its specification from the Requirement that it realizes. On the report, each of the elements in the first column is the source or dependent in a Dependency relationship with the corresponding target element in the **Dependent on** column.

**Access** [Project | QA Reports & Metrics > Dependency Details](#)

#### Run the Dependency report

Field	Action	See also
<b>Root Package</b>	Displays the name of the root Package; all elements and Packages under this Package are included in the report.  If you want to change the Package, click on the replacement Package in the Project Browser and click on the <b>Refresh</b> button.	
<b>Refresh</b>	Click on this button to run the report.	


Field	Action	See also
<b>Locate Object</b>	Click on an element name in the report and click on this button to locate the element in the Project Browser.	
<b>Print</b>	Click on this button to print the dependency details.	
<b>Save Report</b>	Click on this button to save the report as a file. A small dialog displays in which you specify the file location to save the report to.	
<b>Details</b>	<p>Lists the dependency details, consisting of:</p> <ul style="list-style-type: none"> <li>• The names of the elements in the current Package that have a dependency</li> <li>• The type of the elements that have a dependency</li> <li>• The relationship type</li> <li>• The names of the elements that they are dependent on</li> <li>• The type of the elements that they are dependent on</li> </ul>	<a href="#">Dependency</a> <small>1404</small>

#### 19.1.5.4 Maintenance Report

As you manage the maintenance items against elements in your model, you can monitor the status of reported defects, changes, issues and tasks by generating a Maintenance report on the Package. You can filter the report to show maintenance items of a certain type, having a specific status, reported by a specific person and/or resolved by a specific person.

**Access** [Project | QA Reports & Metrics > Maintenance Details](#)

##### Generate a Maintenance report

Field/Button	Action	See also
<b>Root Package</b>	<p>Displays the name of the Package selected from the Project Browser. All elements and packages under this Package are included in the report.</p> <p>If you want to change the Package, click on the new package in the Project Browser and then click on the  icon at the end of the field. The replacement Package name displays in the field.</p>	
<b>Reported By</b>	Click on the drop-down arrow and select a name to filter for maintenance items reported by that person. Click on the <b>x</b> button to clear the field.	
<b>Resolved By</b>	Click on the drop-down arrow and select a name to filter for maintenance items resolved by that person. Click on the <b>x</b> button to clear the field.	

Field/Button	Action	See also
<b>Maintenance Type</b>	Select the radio button for the required item type - <b>Defects, Issues, Changes, Tasks</b> or <b>All</b> types.	
<b>Status</b>	Select the radio button for the required item status - <b>New, Verified, Complete</b> or <b>All</b> statuses.	
<b>Locate Object</b>	(After clicking on an item in the Item Details list.) Click on this button to highlight the element containing the maintenance item, in the Project Browser.	
<b>Refresh</b>	Click on this button to re-run the report query on the Package currently selected in the <b>Root Package</b> field.	
<b>Print</b>	Click on this button to print a summary of the item maintenance list.	

#### Notes

- If you want to edit an item from the report, double-click on the item line; the Maintenance window displays, on which you can update the details
- You can re-organize the display of the listed items using the List Header facilities for reported information

#### Learn more

- [Maintenance](#)<sup>[2619]</sup>
- [Create Maintenance Items](#)<sup>[2625]</sup>
- [List Header](#)<sup>[677]</sup>

### 19.1.5.5 Diagrams Only Report

If you want to report on the diagrams in a Package, and **only** the diagrams, you can generate a **Diagrams Only report**. This is convenient for printing or handling a lot of diagrams in batch, rather than exporting or printing each diagram at a time.

**Access** **Project Browser package context menu | Documentation | Diagrams Only Report**

#### Produce a Diagrams Only report

Field/Button	Action	See also
<b>Output Path</b>	Type in or browse for the output location to create the report in.	

Field/Button	Action	See also
<b>Embed Diagrams in Document</b>	<p>Leave the checkbox selected to create the diagrams within the document file.</p> <p>Deselect the checkbox to represent the diagrams with linked image files.</p>	
<b>Include all child packages</b>	<p>Leave the checkbox selected to include the diagrams in any child Package under the selected Package.</p> <p>Deselect the checkbox to report on the diagrams in the selected Package only.</p>	
<b>Include Diagram Name</b>	<p>Leave the checkbox selected to include the diagram names in the generated documentation.</p> <p>Deselect the checkbox to omit the diagram names.</p>	
<b>Order Diagrams Alphabetically</b>	<p>Leave the checkbox selected to present the diagrams in alphabetical order.</p> <p>Deselect the checkbox to present the diagrams in the order in which they are listed in the Project Browser.</p>	
<b>Diagram Format</b>	<p>Click on the appropriate radio button for the graphics format in which to capture the diagrams.</p> <p>The <b>GIF</b> radio button defaults to selected.</p>	
<b>Page Setup</b>	Click on this button to define the page dimensions and layout for the report.	<a href="#">Set Up Diagram Page</a> <small>870</small>
<b>Generate</b>	Click on this button to run the report.	
<b>View</b>	When the report is generated, click on this button to display the output document.	

#### 19.1.5.6 Testing Report

As you develop the testing environment for your model, you create and execute test scripts for the elements in the model. To help you track the status and results of running the scripts, you can generate a **Testing report** on a Package selected from the Project Browser. This report can cover just the selected Package, or the complete hierarchy of the Package. You can also report on scripts for specific types of test, or for all types of test.

**Access** [Project | Documentation | Testing Report, or Project Browser package context menu | Documentation | Testing Report](#)

**Generate a Testing Report**

Field/Button	Action	See also
<b>Root Package</b>	Displays the selected Package name.  If you want to run the report on a different Package, close the Generate Test Documentation dialog, select the replacement Package in the Project Browser, and open the dialog again.	
<b>Report on</b>	<b>Deselect</b> the check box against each type of test to <b>omit</b> from the report. All checkboxes default to selected.	
<b>Include Child Packages</b>	Leave the checkbox selected to report on tests on elements in the subordinate Packages.  Deselect the checkbox to report just on the root Package.	
<b>Output file</b>	Type in or browse for the file name and path into which to generate the Testing report.	
<b>Generate</b>	Click on this button to generate the report.  A message displays when the report has been generated. You can then open the file in your preferred file viewer.	

**Notes**

- In your file viewer settings, set the page layout to Landscape to display the columns of the report clearly
- The **Testing** report provides a **detailed** description of how each test executed, what the input and acceptance criteria were, and what the results were; the **Testing Details** report is an alternative option, showing a **summary** of the test status and who ran and checked the test

**Learn more**

- [Testing](#)<sup>[2604]</sup>
- [Testing Details Report](#)<sup>[2736]</sup>

## 19.2 Web Reports

As you develop your model you can publish it on the web, where the outline structure closely mirrors the model hierarchy and makes it very simple to explore models on-line. With a great look and the ability to explore very large models efficiently on-line, the web-publishing capability is a significant enhancement. You can export either an entire model or a single branch of the model to the web pages. The report provides an easy to use, highly detailed, Javascript-based model tree. In addition, hyperlinked elements make it very simple to browse to related information.

### Notes

- The HTML output is compatible with any standard web server, on either Unix or Windows platforms - simply bundle up the entire output directory and place it within the context of your web server; all path names should be relative and case sensitive
- You can create **Web Style Templates** to customize your web output
- You can use Master Documents and Model Documents to select specific areas of your model for reporting
- In the Corporate, Business and Software Engineering, System Engineering and Ultimate editions of Enterprise Architect, if security is enabled you must have **Generate Documents** permission to generate HTML documents

### Learn more

- [Create a Web Page Report](#)<sup>[2744]</sup>
- [Create Web Style Templates](#)<sup>[2747]</sup>
- [Master Documents and Model Documents](#)<sup>[2759]</sup>
- [Permission List](#)<sup>[329]</sup>

### Learning Center topics

- **Alt+F1 | Enterprise Architect | Reporting | HTML Reports | Generating an HTML Report**

### 19.2.1 Create a Web Page Report


If you want to create a report on a Package to display as a **web page**, you can select the Package in the Project Browser and perform a simple procedure using the Publish as HTML dialog. All child Packages of the selected Package are automatically included in the report.

**Access** **Project Browser package context menu | Documentation | HTML Report (Shift+F8)**

#### Create a Web Page report

Field/Button	Action	See also
<b>Package</b>	<p>Confirm the name of the selected package, for which you are creating the web documentation.</p> <p>If you want to change the Package, close the dialog, select the new Package and select the <b>HTML Report</b> option again.</p>	



Field/Button	Action	See also
<b>Title</b>	Defaults to the Package name. If you want the report to have a different title, overtype the field contents with your preferred text.	
<b>Output to</b>	Type in or browse for the directory path your report is to be saved to.	
<b>Style</b>	<p>(Optional.) Click on the drop-down arrow and select a custom web style template to apply to your documentation, in place of the default template.</p> <p>Use the <b>&lt;default&gt;</b> value for the standard, system-provided template.</p>	<a href="#">Create Web Style Templates</a> <sup>[2747]</sup>
<b>File</b>	Type the file extension for your web documentation files; the default is <b>.htm</b> .	
<b>Header Image</b>	<p>If you want the report to have an identifying image at the top, type in or browse for the image file directory path and filename.</p> <p>If you do not specify a path, the image defaults to the Enterprise Architect logo.</p>	
<b>Preserve White space in Notes</b>	Select the checkbox to reproduce the spacing of any element notes. Deselect the checkbox to remove white space.	
<b>No Page for Note and Text Items</b>	Select the text box to <b>exclude</b> Note elements and Text elements from the report.	
<b>Default Diagram</b>	<p>Select the appropriate radio button to begin the web report with:</p> <ul style="list-style-type: none"> <li>• The default Model Diagram</li> <li>• The currently-selected diagram</li> <li>• Another diagram, for which you can browse by clicking on the  button</li> <li>• No diagrams (the <b>None</b> radio button defaults to selected)</li> </ul>	
<b>Image Format</b>	Select the appropriate radio button for the file format in which to store images - PNG or GIF.	
<b>Include</b>	<p>Select the appropriate checkboxes to include, in your web document:</p> <ul style="list-style-type: none"> <li>• Test Case elements</li> <li>• Maintenance Items</li> <li>• Resource allocations</li> </ul>	

Field/Button	Action	See also
	<ul style="list-style-type: none"> <li>Hyperlinked files</li> </ul>	
<b>System</b>	Select the appropriate checkboxes to include, in your web document, the System components: <ul style="list-style-type: none"> <li>Glossary</li> <li>Model Tasks</li> <li>Model Issues</li> </ul>	
<b>Generate</b>	Click on this button to generate the report.	
<b>Progress</b>	Shows the accumulating percentage completion of the report generation process.	
<b>View</b>	Once the report is complete, click on this button to launch your default web browser and view the web pages.	

#### View page for specific diagram or element

When you **view** the web report in your web browser, you can switch directly to a page for a specific diagram or element by specifying the appropriate GUID after the report web address. That is:

```
ht t p : / / pat h / pat h / pat h / I ndex . ht m ? gui d = x x x x x x x x x x x
```

Type the word *guid* in lower case, and do not include braces { } in the value; for example:

```
ht t p : / / . . . / pat h / I ndex . ht m ? gui d = DC62B0DA- 0D60- 4447- 85E6- B9BBAE7FC90F
```

To obtain the GUID, right-click on the Package or element in the Project Browser and select the **Copy/Paste** menu option, or click on the diagram and select the **Copy Reference** context menu option.

#### Browser behaviour

The display of your web report can be affected by the browser that you use. Specifically:

- Firefox** automatically converts the diagram or element page locator path (above) to a **file:///C:/path** protocol and actions it
- Opera** automatically adds **file://localhost/** to the start of the page locator path, in the same way as Firefox adds the **file:///C:/path** protocol
- Opera** by default disables **XMLHttpRequest** when loading a file, and the web report detects this and prompts the user to change settings; there are no problems when the page is hosted on a web server
- Microsoft Internet Explorer** does not operate on the page locator path directly, but the protocol in Firefox also works in Internet Explorer; therefore, to use the absolute references without a web server, you must access the path using the **file:///** protocol
- If you are using **Internet Explorer 7.0** or later, and you do not have it open, its security profile might block the report display; click on the explanation banner at the top of the screen and select the **Allow**

**Blocked Content** context menu option

- **Chrome** completely disables **XMLHttpRequest** when loading a file, so will not display web reports directly from the file system; there are no problems when the page is hosted on a webserver

### Notes

- To generate a demonstration web report right now, open the *EAExample* project, select the *System Model* package and follow the steps above
- You can also select Packages in your model to exclude them (and their subordinate Packages) from the web reports you generate

### Learn more

- [Diagram Menu](#)<sup>[665]</sup>
- [Element Options in the Project Browser](#)<sup>[660]</sup>
- [Package Menu](#)<sup>[651]</sup>
- [Exclude Package from Report](#)<sup>[2761]</sup>

### Learning Center topics

- **Alt+F1 | Enterprise Architect | Reporting | HTML Reports**

## 19.2.2 Create Web Style Templates

If you want to customize the look and feel of a web report for your company or client, you can use the **HTML and CSS Style Editor** to change the HTML associated with the various sections of the report. This editor is derived from, and provides the facilities of, the common Code Editor. The HTML and CSS Style Editor contains a set of HTML **fragments** for modification and customization, listed in the left hand Templates panel of the dialog.

Each fragment typically contains HTML plus one or more special tag names that are replaced with information during generation. The special tag names are delimited by two # characters; for example, **#NOTES#**, as you can see if you click on a template fragment name to display its contents in the right hand Current Modified Template panel. The HTML report generator produces output based on the fragments, either in their default state or as you have customized them. Currently you cannot alter the content within the tag names, but you can omit a complete tag by removing it, or alter its basic display properties in the surrounding HTML.

**Access** **Project | Resources | Document Generation | Web Style Templates**

### Create or edit web style templates

Task	Action	See also
<b>Create a new template</b>	<p>Right-click on the <i>Web Style Templates</i> folder and select the <b>Create HTML Template</b> context menu option.</p> <p>Type the name of the new template when prompted to do so. The HTML and CSS Style Editor displays.</p>	

Task	Action	See also
<b>Edit an existing template</b>	<p>Expand the <i>Web Style Templates</i> folder and either:</p> <ul style="list-style-type: none"> <li>• Double-click on the template name or</li> <li>• Right-click on the template name and select the <b>Modify HTML Style Template</b> context menu option</li> </ul> <p>The HTML and CSS Style Editor displays.</p>	
<b>Delete a customized template</b>	<p>Expand the <i>Web Style Templates</i> folder, right-click on the template name and select the <b>Delete HTML Template</b> context menu option.</p>	

#### Within the HTML and CSS Style Editor

Task	Action	See also
<b>Edit a template fragment</b>	<p>Click on the required fragment in the <b>Templates</b> list; the fields in that fragment display in the Current Modified Template panel.</p> <p>Change the field values, or delete fields, as you wish.</p>	<a href="#">HTML Template Fragments</a> <small>[2749]</small>
<b>Save your style changes</b>	<p>Click on the <b>Save</b> button to save your version of this template fragment only.</p> <p>Save each fragment before you move on to another fragment.</p>	
<b>Delete a modified template fragment</b>	<p>Click on the template fragment in the left-hand list and click on the <b>Delete</b> button. This removes the modified version of the fragment, which causes the report generator to use the default (system-provided) fragment during report generation.</p>	
<b>Retrieve the original, default template fragment</b>	<p>Click on the template fragment in the left hand list and click on the <b>Get Default</b> button.</p>	

#### Notes

- Each time you generate the web report it overwrites the templated files, so you must back up your modified versions and copy them back in after every update

#### Learn more

- [Code Editor Functions](#)  
[2157]

### Learning Center topics

- [Alt+F1 | Enterprise Architect | Reporting | HTML Templates | Create a New Template](#)

#### 19.2.2.1 HTML Template Fragments

When you create a **web report template** in the **HTML and CSS Style Editor**, it automatically contains a wide range of system-provided **HTML Template fragments**. You can edit the fields of each separate fragment, or use the unchanged default.

**Access** [Project | Resources | Document Generation | Web Style Templates | right-click | Create HTML Template](#), or  
[Project | Resources | Document Generation | Web Style Templates | <template name> | right-click | Modify HTML Style Template](#)

### Fragment Fields

Fragment	Fields
<b>Body - Diagram</b>	<ul style="list-style-type: none"> <li>• #AUTHOR# - Diagram author</li> <li>• #CREATEDATE# - Diagram created date</li> <li>• #CSS# - Style Sheet to use</li> <li>• #GUID# - Diagram GUID</li> <li>• #IMAGE# - Image of the diagram</li> <li>• #IMAGES# - Image path</li> <li>• #LOCKED# - Is the diagram locked (<b>true</b> or <b>false</b>)</li> <li>• #MODDATE# - Diagram modified date</li> <li>• #NAME# - Diagram name</li> <li>• #NOTE# - Diagram notes</li> <li>• #STEREOTYPE# - Diagram stereotype</li> <li>• #TITLE# - Diagram title</li> <li>• #TYPE# - Diagram type</li> <li>• #VERSION# - Diagram version</li> </ul>
<b>Body – Object</b>	<ul style="list-style-type: none"> <li>• #ABSTRACT# - Element abstract (if true, = <b>abstract</b>)</li> <li>• #ACTIVE# - Element isActive (<b>true</b> or <b>false</b>)</li> <li>• #ALIAS# - Element alias ("<b>#ALIAS#</b>")</li> <li>• #AUTHOR# - Element author</li> <li>• #BEHAVIOR# - Returns the object behavior.</li> <li>• #CLASSIFIER# - Element classifier</li> <li>• #CLASSIFIERREF# - Returns the HREF of the classifier</li> <li>• #COMPLEXITY# - Element complexity</li> <li>• #CREATEDATE# - Element created date</li> <li>• #CSS# - Style Sheet to use</li> <li>• #DIAGRAMS# - List of diagrams the element is on</li> </ul>

Fragment	Fields
	<ul style="list-style-type: none"> <li>• #DIFF# - Element difficulty</li> <li>• #GUID# - Element GUID</li> <li>• #IMAGES# - Image path</li> <li>• #KEYWORDS# - Element keywords</li> <li>• #LANGUAGE# - Element language</li> <li>• #LEAF# - Element isLeaf (<b>true</b> or <b>false</b>)</li> <li>• #LEVELNUMBER# - Element level number</li> <li>• #MODDATE# - Element modified date</li> <li>• #MULTIPLICITY# - Element multiplicity ("<b>Multiplicity:</b> <b>#MULTIPLICITY#</b>")</li> <li>• #NAME# - Element name</li> <li>• #NOTE# - Element notes</li> <li>• #PHASE# - Element phase</li> <li>• #PRIORITY# - Element priority</li> <li>• #ROOT# - Element isRoot (<b>true</b> or <b>false</b>)</li> <li>• #SCOPE# - Element scope</li> <li>• #STATUS# - Element status</li> <li>• #STEREOTYPE# - Element stereotype(s)</li> <li>• #TYPE# - Element type; for example, Class, Object</li> <li>• #VERSION# - Element version</li> </ul>
<b>Body – Project Glossary</b>	<ul style="list-style-type: none"> <li>• #CONTENT# - Loops <b>Body – Project Glossary Item</b> for each Project Glossary item</li> </ul>
<b>Body - Project Glossary Item</b>	<ul style="list-style-type: none"> <li>• #MEANING# - Glossary Meaning</li> <li>• #TERM# - Glossary Term</li> <li>• #TYPE# - Glossary Type</li> </ul>
<b>Body – Project Issue</b>	<ul style="list-style-type: none"> <li>• #CONTENT# - Loops <b>Body – Project Issue Item</b> for each Project Issue item</li> </ul>
<b>Body - Project Issue Item</b>	<ul style="list-style-type: none"> <li>• #DATERESOLVED# - Project Issue resolved date (blank if no date entered)</li> <li>• #ISSUE# - Project Issue name</li> <li>• #ISSUEDATE# - Project Issue issue date</li> <li>• #NOTES# - Project Issue notes</li> <li>• #OWNER# - Project Issue owner</li> <li>• #RESOLUTION# - Project Issue resolution</li> <li>• #RESOLVER# - Project Issue resolver</li> <li>• #STATUS# - Project Issue status</li> </ul>

Fragment	Fields
<b>Body – Project Task</b>	<ul style="list-style-type: none"> <li>• #CONTENT# - Loops <b>Body – Project Task Item</b> for each Project Task item</li> </ul>
<b>Body - Project Task Item</b>	<ul style="list-style-type: none"> <li>• #ENDDATE# - Project Task end date</li> <li>• #NAME# - Project Task name</li> <li>• #NOTES# - Project Task notes</li> <li>• #OWNER# - Project Task owner</li> <li>• #PHASE# - Project Task phase</li> <li>• #PRIORITY# - Project Task priority</li> <li>• #STARTDATE# - Project Task start date</li> <li>• #STATUS# - Project Task status</li> <li>• #TYPE# - Project Task type</li> </ul>
<b>Content – Attributes</b>	<ul style="list-style-type: none"> <li>• #CONTENT# - Loops <b>Content – Attributes Item</b> for each attribute on an element</li> </ul>
<b>Content – Attributes Item</b>	<ul style="list-style-type: none"> <li>• #ALIAS# - Attribute alias ("<i>&lt;i&gt;Alias:&lt;/i&gt; #ALIAS#&lt;br&gt;</i>")</li> <li>• #ALLOWDUPLICATES# - For Non-Table elements, returns True or False</li> <li>• #COLLECTION# - For Non-Table elements, returns True or False</li> <li>• #CONST# - Attribute is constant value ("<b>const</b> " &lt;-- Note Space)</li> <li>• #CONSTRAINT# - Attribute constraint</li> <li>• #DEFAULT# - Attribute default value ("<i>&lt;i&gt;Initial:&lt;/i&gt; #DEFAULT&lt;br&gt;</i>")</li> <li>• #DERIVED# - Attribute is Derived</li> <li>• #FK# - Foreign Key, for Table elements returns True or False</li> <li>• #ISID# - Attribute is ID</li> <li>• #ISORDERED# - For Non-Table elements, returns True or False</li> <li>• #NAME# - Attribute name</li> <li>• #NOTE# - Attribute notes</li> <li>• #NOTNULL# - For Table elements, returns True or False</li> <li>• #ORDERED# - Attribute Is ordered value ("<b>Ordered &lt;br /&gt;</b>")</li> <li>• #PK# - Primary Key, for Table elements returns True or False</li> <li>• #RANGE# - If lower != 1 ("<b>Range:&lt;lower&gt; to &lt;upper&gt;</b>")</li> <li>• #SCOPE# - Attribute scope ("<b>SCOPE#</b> " &lt;-- Note space)</li> <li>• #STATIC# - Attribute is static value ("<b>static</b> " &lt;-- Note Space)</li> <li>• #STEREOTYPE# - Attribute stereotype(s)</li> <li>• #TAGS# - Attribute tags ("<b>Property Name=Property Value&lt;br&gt;</b>")</li> <li>• #TRANSIENT# - Attribute is Transient</li> <li>• #TYPE# - Attribute type (if <b>Column</b>, "<b>#TYPE(Column Precision, Scale)</b>")</li> </ul>

Fragment	Fields
	<ul style="list-style-type: none"> <li>• #UNIQUE# - For Table elements, returns True or False</li> </ul>
<b>Content – Constraints</b>	<ul style="list-style-type: none"> <li>• #CONTENT# - Loops <b>Content – Constraints Item</b> for each constraint on an element</li> </ul>
<b>Content – Constraints Item</b>	<ul style="list-style-type: none"> <li>• #CONSTRAINT# - Constraint name</li> <li>• #NOTES# - Constraint notes</li> <li>• #STATUS# - Constraint status</li> <li>• #TYPE# - Constraint type</li> </ul>
<b>Content – Custom Properties</b>	<ul style="list-style-type: none"> <li>• #CONTENT# - List of all <b>Content – Custom Properties - Item</b></li> </ul>
<b>Content – Custom Properties - Item</b>	<ul style="list-style-type: none"> <li>• #PROPERTY# - Custom property name</li> <li>• #VALUE# - Custom property value            ( if type equal boolean                true or false                else                #VALUE#)</li> </ul>
<b>Content - Element Effort</b>	<ul style="list-style-type: none"> <li>• #CONTENT# - List of <b>Content - Element Effort Item</b></li> </ul>
<b>Content - Element Effort Item</b>	<ul style="list-style-type: none"> <li>• #EFFORT# - Effort name</li> <li>• #TYPE# - Effort Type</li> <li>• #TIME# - Effort Time</li> <li>• #NOTES# - Effort Notes</li> </ul>
<b>Content - Element Risk</b>	<ul style="list-style-type: none"> <li>• #CONTENT# - List of <b>Content - Element Risk Item</b></li> </ul>
<b>Content - Element Risk Item</b>	<ul style="list-style-type: none"> <li>• #RISK# - Risk name</li> <li>• #TYPE# - Risk Type</li> <li>• #WEIGHT# - Risk Weight</li> <li>• #NOTES# - Risk Notes</li> </ul>
<b>Content - Element Metric</b>	<ul style="list-style-type: none"> <li>• #CONTENT# - List of <b>Content - Element Metric Item</b></li> </ul>
<b>Content - Element Metric Item</b>	<ul style="list-style-type: none"> <li>• #METRIC# - Metric name</li> <li>• #TYPE# - Metric type</li> <li>• #WEIGHT# - Metric weight</li> <li>• #NOTES# - Metric notes</li> </ul>



Fragment	Fields
<b>Content – Files</b>	<ul style="list-style-type: none"> <li>• #CONTENT# - List of <b>Content - Files Item</b></li> </ul>
<b>Content – Files Item</b>	<ul style="list-style-type: none"> <li>• #DATE# - File date</li> <li>• #NAME# - File filename (&lt;a href="#FILENAME#" &lt;/a&gt;)</li> <li>• #NOTES - File notes</li> <li>• #SIZE# - File size</li> <li>• #TYPE# - File type</li> </ul>
<b>Content - Inherited Attributes</b>	<ul style="list-style-type: none"> <li>• #CONTENT# - Loops <b>Content – Inherited Attributes Item</b> for each attribute on an element</li> </ul>
<b>Content - Inherited Attributes Item</b>	<ul style="list-style-type: none"> <li>• #ALIAS# - Attribute alias ("<i>&lt;i&gt;Alias:&lt;/i&gt; #ALIAS#&lt;br&gt;</i>")</li> <li>• #CONST# - Attribute is constant value ("<b>const</b> " &lt;-- Note Space)</li> <li>• #CONSTRAINT# - Attribute constraint</li> <li>• #DEFAULT# - Attribute default value ("<i>&lt;i&gt;Initial:&lt;/i&gt; #DEFAULT&lt;br&gt;</i>")</li> <li>• #NAME# - Attribute name</li> <li>• #NOTE# - Attribute notes</li> <li>• #ORDERED# - Attribute Is ordered value ("<b>Ordered &lt;br /&gt;</b>")</li> <li>• #RANGE# - If lower != 1 ("<b>Range:&lt;lower&gt; to &lt;upper&gt;</b>")</li> <li>• #SCOPE# - Attribute scope ("<b>SCOPE#</b> " &lt;-- Note space)</li> <li>• #STATIC# - Attribute is static value ("<b>static</b> " &lt;-- Note Space)</li> <li>• #STEREOTYPE# - Attribute stereotype(s)</li> <li>• #TAGS# - Attribute tags ("<b>Property Name=Property Value&lt;br&gt;</b>")</li> <li>• #TYPE# - Attribute type (if <b>Column</b>, "<b>#TYPE(Column Precision, Scale)</b>")</li> </ul>
<b>Content - Inherited Operations</b>	<ul style="list-style-type: none"> <li>• #CONTENT# - List of <b>Content - Inherited Operations Item</b></li> </ul>
<b>Content - Inherited Operations Item</b>	<ul style="list-style-type: none"> <li>• #ABSTRACT# - Operation abstract (<b>abstract</b>)</li> <li>• #ALIAS# - Operation alias ("<i>&lt;i&gt;Alias:&lt;/i&gt; #ALIAS#&lt;br&gt;</i>")</li> <li>• #CONCURRENCY# - Operation concurrency (<b>blank</b> if not set)</li> <li>• #CONST# - Operation constant (<b>const</b>)</li> <li>• #CONSTRAINTS# - List of <b>Method Constraint</b></li> <li>• #ISQUERY# - Operation IsQuery (<b>isQuery</b>)</li> <li>• #NAME# - Operation name</li> <li>• #NOTE# - Operation notes.</li> <li>• #PARAMS# - List of <b>Content – Operations Item Parameters</b></li> <li>• #SCOPE# - Operation Scope</li> <li>• #STATIC# - Operation IsStatic (<b>static</b>)</li> </ul>

Fragment	Fields
	<ul style="list-style-type: none"> <li>• #STEREOTYPE# - Operation stereotype</li> <li>• #TAGLABEL# - static text (<b>Tags</b>)</li> <li>• #TAGS# - Attribute tags ("<b>Property Name=Property Value&lt;br&gt;</b>")</li> <li>• #TYPE# - Operation type</li> </ul>
<b>Content – Notes</b>	<ul style="list-style-type: none"> <li>• #VALUE#&gt; - Notes text</li> </ul>
<b>Content – Operations</b>	<ul style="list-style-type: none"> <li>• #CONTENT# - List of <b>Content - Operations Item</b></li> </ul>
<b>Content – Operations Item</b>	<ul style="list-style-type: none"> <li>• #ABSTRACT# - Operation abstract (<b>abstract</b>)</li> <li>• #ALIAS# - Operation alias ("<b>&lt;i&gt;Alias:&lt;/i&gt; #ALIAS#&lt;br&gt;</b>")</li> <li>• #CONCURRENCY# - Operation concurrency (<b>blank</b> if not set)</li> <li>• #CONST# - Operation constant (<b>const</b>)</li> <li>• #CONSTRAINTS# - List of <b>Method Constraint</b></li> <li>• #ISQUERY# - Operation IsQuery (<b>isQuery</b>)</li> <li>• #NAME# - Operation name</li> <li>• #NOTE# - Operation notes.</li> <li>• #PARAMS# - List of <b>Content – Operations Item Parameters</b></li> <li>• #SCOPE# - Operation Scope</li> <li>• #STATIC# - Operation IsStatic (<b>static</b>)</li> <li>• #STEREOTYPE# - Operation stereotype</li> <li>• #TAGLABEL# - static text (<b>Tags</b>)</li> <li>• #TAGS# - Attribute tags ("<b>Property Name=Property Value&lt;br&gt;</b>")</li> <li>• #TYPE# - Operation type</li> </ul>
<b>Content – Operations Item Parameters</b>	<ul style="list-style-type: none"> <li>• #DEFAULT# - Op Parameter default</li> <li>• #GUID# - Op Parameter GUID</li> <li>• #KIND# - Op Parameter kind</li> <li>• #NAME# - Op Parameter name</li> <li>• #NOTES# - Op Parameter notes</li> <li>• #STEREOTYPE# - Op Parameter stereotype</li> <li>• #TYPE# - Op Parameter type</li> </ul>
<b>Content – Resource Allocation</b>	<ul style="list-style-type: none"> <li>• #CONTENT# - List of <b>Content – Resource Allocation Item</b></li> </ul>
<b>Content – Resource Allocation Item</b>	<ul style="list-style-type: none"> <li>• #ACTUAL# - Resource actual time</li> <li>• #ENDDATE# - Resource end date</li> <li>• #EXPECTED# - Resource expected date</li> <li>• #NOTES# - Resource notes</li> <li>• #PERCENT# - Resource percent complete</li> </ul>

Fragment	Fields
	<ul style="list-style-type: none"> <li>• #RESOURCE# - Resource name</li> <li>• #ROLE# - Resource role</li> <li>• #STARTDATE# - Resource start date</li> <li>• #TIME# - Resource time</li> </ul>
<b>CSS – Main</b>	<ul style="list-style-type: none"> <li>• None</li> </ul>
<b>CSS – ToC</b>	<ul style="list-style-type: none"> <li>• None</li> </ul>
<b>Feature Notes</b>	<ul style="list-style-type: none"> <li>• #FIELD#</li> <li>• #VALUE#</li> </ul>
<b>Javascript</b>	<ul style="list-style-type: none"> <li>• None</li> </ul>
<b>Link (Association)</b>	<ul style="list-style-type: none"> <li>• #CONTENT#</li> </ul>
<b>Link (Association) Item</b>	<ul style="list-style-type: none"> <li>• #CONNECTION# - Connector type</li> <li>• #IMAGE# - The file path of the images</li> <li>• #LINK# - (&lt;a href= "path to element"&gt; "Connection Name" &lt;/a&gt;)</li> <li>• #NOTES# - The connector notes</li> <li>• #NUMBER# - A unique number used to identify div elements</li> <li>• #SOURCEROLE# - Source role</li> <li>• #SOURCEROLENOTES# - Source role notes</li> <li>• #STEREOTYPE# - Connector stereotype</li> <li>• #TARGETROLE# - Target role</li> <li>• #TARGETROLENOTES# Target role notes</li> <li>• #TYPE# - Connector type</li> </ul>
<b>Link (Flow)</b>	<ul style="list-style-type: none"> <li>• #CONTENT#</li> </ul>
<b>Link (Flow) Item</b>	<ul style="list-style-type: none"> <li>• #DIRECTION# - Connector direction</li> <li>• #ELEMNAME# - Name of the element at the source/destination of the connector</li> <li>• #IMAGE# - The file path of the images</li> <li>• #LINK# - (&lt;a href= "path to element"&gt; "Connection Name" &lt;/a&gt;)</li> <li>• #LINKREF# - Page name of the element at the source/destination of the connector</li> <li>• #NAME# - Connector name</li> <li>• #NOTES# - The connector notes</li> <li>• #NUMBER# - A unique number used to identify div elements</li> <li>• #STEREOTYPE# - Connector stereotype</li> </ul>

Fragment	Fields
	<ul style="list-style-type: none"> <li>• #TYPE# - Connector type</li> </ul>
<b>Link (Other)</b>	<ul style="list-style-type: none"> <li>• #CONTENT#</li> </ul>
<b>Link (Other) Item</b>	<ul style="list-style-type: none"> <li>• #CONNECTION# - Connector type</li> <li>• #IMAGE# - The file path of the images</li> <li>• #LINK# - (&lt;a href= "path to element"&gt; "Connection Name" &lt;/a&gt;)</li> <li>• #NOTES# - The connector notes</li> <li>• #NUMBER# - A unique number used to identify div elements</li> <li>• #SOURCEROLE# - Source role</li> <li>• #SOURCEROLENOTES# - Source role notes</li> <li>• #STEREOTYPE# - Connector stereotype</li> <li>• #TARGETROLE# - Target role</li> <li>• #TARGETROLENOTES# Target role notes</li> <li>• #TYPE# - Connector type</li> </ul>
<b>Linked Document</b>	<ul style="list-style-type: none"> <li>• #LINKDOC# - Linked Document.</li> </ul>
<b>Linked Requirement</b>	<ul style="list-style-type: none"> <li>• #CONTENT# - List of <b>Linked Requirement Item</b></li> </ul>
<b>Linked Requirement Item</b>	<ul style="list-style-type: none"> <li>• #DIFF# - Linked Requirement difficulty</li> <li>• #NAME# - Linked Requirement name</li> <li>• #PRIORITY# - Linked Requirement priority</li> <li>• #STATUS# - Linked Requirement status</li> </ul>
<b>Linked Section</b>	<ul style="list-style-type: none"> <li>• #ITEMS#</li> <li>• #TITLE#</li> </ul>
<b>Maintenance</b>	<ul style="list-style-type: none"> <li>• #CONTENT# - List of <b>Maintenance Line Item</b></li> </ul>
<b>Maintenance Line Item</b>	<ul style="list-style-type: none"> <li>• #DATEREPORTED# - Maintenance date reported</li> <li>• #DATERESOLVED# - Maintenance date resolved</li> <li>• #IMAGE# - The file path of the images</li> <li>• #NOTES# - Maintenance notes</li> <li>• #NUMBER# - A unique number used to identify div elements</li> <li>• #PRIORITY# - Maintenance priority</li> <li>• #PROBLEM# - Maintenance name</li> <li>• #REPORTEDBY# - Maintenance reported by</li> <li>• #RESOLVEDBY# - Maintenance resolved by</li> <li>• #RESOLVERNOTES# - Maintenance resolved notes</li> </ul>

Fragment	Fields
	<ul style="list-style-type: none"> <li>• #STATUS# - Maintenance status</li> <li>• #TYPE# - Maintenance type</li> <li>• #VERSION# - Maintenance version</li> </ul>
<b>Message</b>	<p>(Applies only to Sequence messages.)</p> <ul style="list-style-type: none"> <li>• #CONTENT# - Loops the <b>Message Item</b> for each attribute on an element</li> <li>• #DIRECTION# - Contains the value <b>To</b> or <b>From</b></li> </ul>
<b>Message Item</b>	<ul style="list-style-type: none"> <li>• #KIND# - The Message <b>Kind</b> field</li> <li>• #MESSAGE# - Connector Message</li> <li>• #NAME# - Name of the Message (&lt;a href="#&lt;path&gt;"&gt;#NAME#&lt;/a&gt;) If Message has a classifier: #NAME#="#NAME# :Classifier"</li> <li>• #NOTES# - The Message notes (&lt;strong&gt;Type:&lt;/strong&gt; #Item Type#&lt;br /&gt; #NOTES#)</li> <li>• #SYNCH# - The Message <b>Synch</b> field</li> <li>• #TYPE# - The type of Message</li> </ul>
<b>Method Constraint</b>	<ul style="list-style-type: none"> <li>• #NAME# - Method Constraint name</li> <li>• #NOTES# - Method Constraint notes</li> <li>• #TYPE# - Method Constraint type</li> </ul>
<b>Object Requirement</b>	<ul style="list-style-type: none"> <li>• #CONTENT# - List of <b>Object Requirement Item</b></li> </ul>
<b>Object Requirement Item</b>	<ul style="list-style-type: none"> <li>• #DIFF# - Requirement difficulty</li> <li>• #NAME# - Requirement name</li> <li>• #NOTES# - Requirement notes</li> <li>• #PRIORITY# - Requirement priority</li> <li>• #STABILITY# - Requirement stability</li> <li>• #STATUS# - Requirement status</li> <li>• #TYPE# - Requirement type</li> </ul>
<b>Package Content</b>	<ul style="list-style-type: none"> <li>• #CONTENT# - List of <b>Package Content Row</b></li> </ul>
<b>Package Content Row</b>	<ul style="list-style-type: none"> <li>• #NAME# - Link to Package (&lt;a href="#Link to file#"&gt;#Package name#&lt;/a&gt;)</li> <li>• #TYPE# - Link to Image (&lt;img src="#path to image#"&gt;)</li> </ul>
<b>Page - Basic template</b>	<ul style="list-style-type: none"> <li>• #CONTENT# - Contains <b>Body - Diagram</b> through to <b>Body - Object</b></li> <li>• #TITLE# - Current package name</li> </ul>
<b>Page - Index</b>	<ul style="list-style-type: none"> <li>• #CSS# - Style Sheet to use</li> </ul>

Fragment	Fields
	<ul style="list-style-type: none"> <li>• #HOME# - A link to the Start page</li> <li>• #JS# - Javascript to use</li> <li>• #TITLE# - Current package name</li> <li>• #TOC# - Table of Contents</li> </ul>
<b>Page - ToC</b>	<ul style="list-style-type: none"> <li>• None</li> </ul>
<b>Scenario</b>	<ul style="list-style-type: none"> <li>• #CONTENT# - List of <b>Scenario Item</b></li> <li>• #EXCEPTIONS# - List of Structured Scenario exceptions</li> <li>• #STRUCTURED# - List of Structured Scenarios</li> </ul>
<b>Scenario Item</b>	<ul style="list-style-type: none"> <li>• #IMAGE# - The file path of the images</li> <li>• #NOTES# - Scenario notes</li> <li>• #NUMBER# - A unique number used to identify div elements</li> <li>• #SCENARIO# - Scenario name</li> <li>• #TYPE# - Scenario type</li> </ul>
<b>Scenario Exception</b>	<ul style="list-style-type: none"> <li>• #CONTENT# - Loops <b>Scenario Exception Item</b> for each exception</li> </ul>
<b>Scenario Exception Item</b>	<ul style="list-style-type: none"> <li>• #NAME# - Exception name</li> <li>• #STEPNO# - Exception step number</li> <li>• #TYPE# - Exception Type</li> </ul>
<b>Scenario Structured</b>	<ul style="list-style-type: none"> <li>• #CONTENT# - Loops <b>Scenario Structured Items</b> for each Structured Scenario item</li> </ul>
<b>Scenario Structured Items</b>	<ul style="list-style-type: none"> <li>• #ACTION# - Name of the scenario</li> <li>• #STEPNO# - Scenario step number</li> <li>• #RESULT# - Step result value</li> <li>• #USES# - Step uses value</li> <li>• #STATE# - Step state value</li> <li>• #TYPE# - Step type (System or User)</li> </ul>
<b>Tagged Value</b>	<ul style="list-style-type: none"> <li>• #CONTENT# - List of <b>Tagged Value Line Item</b></li> </ul>
<b>Tagged Value Line Item</b>	<ul style="list-style-type: none"> <li>• #IMAGE# - The file path of the images</li> <li>• #NOTES# - Tagged Value notes</li> <li>• #NUMBER# - A unique number used to identify div elements</li> <li>• #PROPERTY# - Tagged Value name</li> <li>• #VALUE# - Tagged Value if type is boolean (value is <b>true</b> or <b>false</b>)</li> </ul>

Fragment	Fields
<b>Test Cases</b>	<ul style="list-style-type: none"> <li>• #CONTENT# - List of <b>Test Cases Line Item</b></li> </ul>
<b>Test Cases Line Item</b>	<ul style="list-style-type: none"> <li>• #ACCEPTANCE# - Test case acceptance notes</li> <li>• #CHECKEDBY# - Test case checked by</li> <li>• #CLASS# - Test case class (<b>Unit, Integration, System, Acceptance, Scenario</b>)</li> <li>• #IMAGE# - The file path of the images</li> <li>• #INPUT# - Test case input notes</li> <li>• #NOTES# - Test case notes</li> <li>• #NUMBER# - A unique number used to identify div elements</li> <li>• #RESULTS# - Test case result notes</li> <li>• #RUNBY# - Test case run by</li> <li>• #RUNDATE# - Test case last run</li> <li>• #STATUS# - Test case status</li> <li>• #TEST# - Test case name</li> <li>• #TYPE# - Test case type</li> </ul>

#### Learn more

- [Create Web Style Templates](#)<sup>[2747]</sup>

### 19.2.3 Master Documents and Model Documents

You can develop virtual documentation for output in both printable document and web page formats, using Master Document and Model Document elements. The process for generating web virtual documentation differs from that for printable documentation in two ways:

- You use a standard or customized HTML template, which you select in the **Style** field on the Publish as HTML dialog; you do **not** specify or use any templates defined in the element **RTFTemplate** Tagged Value
- You create a **Package list** to select the elements for the report, and not a Model Search (which is defined in the **SearchName** and **SearchValue** Tagged Values in the Model Document elements)

**Access** Click **Document element**, then **Project | Documentation | Publish as HTML** (Shift+F8) or **Project Browser Document element context menu | Documentation | HTML Report**

#### Generate web documentation from a Master Document and/or Model Documents

Step	Action	See also
1	<p>On the Publish as HTML dialog, set the options for your web document as you wish.</p> <p>Select your preferred template - such as a customized Master Document</p>	<a href="#">Create a Web Page Report</a> <sup>[2748]</sup>

Step	Action	See also
	template - in the <b>Style</b> field.	
2	Click on the <b>Generate</b> button to generate the documentation.  The HTML Report Generator works through the defined content of the Master Document element and/or the Model Document elements, and pulls in the information from the listed Packages, formatted according to the template identified in the <b>Style</b> field.	
3	Click on the <b>View</b> button to view the documentation.	

#### Learn more

- [Virtual Documents](#) 

#### Learning Center topics

- **Alt+F1 | Enterprise Architect | Reporting | Virtual Documents**



## 19.3 Exclude Package from Report

When you generate a document or web report on a Package, all child Packages of that Package are, by default, included in the report. If you do not want to include a particular Package in any reports, you can mark it for exclusion. All of the excluded Package's child Packages are also excluded from the reports.

**Access**   **Project Browser Package context menu | Documentation | Generated Report Options**

### Mark a Package to be excluded from reporting

Step	Action	See also
1	On the Generated Report Options dialog, select the <b>Exclude Package from Generated Reports</b> radio button.  (Select the <b>Include Package in Generated Reports</b> radio button to add the Packages to the reports again.)	
2	Click on the <b>OK</b> button.	

## 19.4 Charts

Enterprise Architect can provide you with a **wealth of data** that is crucial to business planning, organizational strategy, decision making and Project Management. One way of **summarizing** this data in a format suitable for swift and easy appraisal is to present it in the form of **charts and graphs**, which are ideal for including in reports and distributing via the internet. Within Enterprise Architect you can create **Chart** elements that define the type, source, content and appearance of a chart, either on their own **Dashboard** diagram (one of the **Extended** diagram types) or on other types of diagram as best suits your requirements. This provides a simple and fast mechanism for collating and presenting a lot of information automatically, such as summaries of Requirement Status or Test Case Status values across the current project.

### Types of Chart Available

Using Enterprise Architect, you can create a number of different charts including:

- Pie - 2-dimensional and 3-dimensional
- Doughnut - 2D and 3D
- Torus
- Line Graph (Time Series)
- Table (Model Views)
- Horizontal bar - 2D and 3D
- Vertical column - 2D and 3D

You can also generate some of these charts filtered according to **another** data quantity, presenting this as table columns, segments of a bar or separate bars in a cluster.

### Creating a Chart

You create a Chart element by dragging a Chart icon onto a diagram from the **Dashboard** pages of the Diagram Toolbox onto a diagram (standard Chart icons are also available on the common **Artifacts** page of the Toolbox). You then define the type of Chart, the data it presents, the source of that data and the appearance of the Chart, within the **element Properties**.

Depending on the type of Chart you are creating, the data can be sourced from across the **whole model**, from **specific Packages** within the model, or from a custom **SQL Query** that extracts the information from the model. You can also paste the data from an **external CSV file** into the element, to be presented as a Chart.

Each Chart is **dynamic**, and is automatically updated whenever you edit it or open its parent diagram. You can also manually refresh it using an option on the element context menu. Time Series Charts can also be automatically updated by the Cloud Server according to a schedule that you define in the element Properties.

### Key Features

A major benefit of generating a Chart as and from a model element is that many of the facilities for working with elements - and the Packages and diagrams that contain them - are available on the Chart, such as:

- Including charts and graphs in **document reports**
- Hosting Charts on a **website** to facilitate communication
- **Automatically updating** and processing the element content (and hence refreshing the Chart)
- Performing **XMI imports and exports** on Chart Packages
- Saving a Chart as a diagram **image**

- Setting the Chart diagram as the **model** or **user default diagram**
- Defining a Chart as a **model Pattern**
- Changing the element properties and getting an **immediate change** to the content and/or appearance of the Chart; useful in, for example, testing a search and visualizing the results

#### Learn more

- [Chart Elements](#) <sup>[2763]</sup>
- [Define a Model View Chart](#) <sup>[2765]</sup>
- [Define a Time Series Chart](#) <sup>[2767]</sup>
- [Standard Chart Data](#) <sup>[2770]</sup>
- [Chart Appearance](#) <sup>[2774]</sup>
- [Including Charts in Reports](#) <sup>[2788]</sup>

### 19.4.1 Chart Elements

The starting point for defining your own data charts is to drag one of the **Chart** element icons from the Diagram Toolbox onto your diagram. The Chart icons are divided into two groups:

- **Patterned** Pie Charts, Bar Charts, Series Charts and Model Views, which create Charts ready-defined for specific purposes and just require small adjustments to tailor them for your project data
- **Basic templates** for Standard Charts (Pie and Bar Charts), Time Series graphs and Model View tables, which create Charts that you develop to meet your own requirements

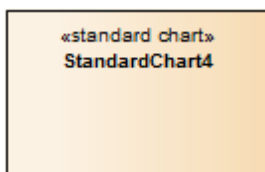
Both groups of icons are available from the **Dashboard** pages of the Diagram Toolbox. Basic template icons are also available from the common **Artifacts** page of the Diagram Toolbox.

Additionally, you can obtain a selection of sample Charts (reporting on Issues and Defects) together on a diagram by selecting the **Dashboard: Issues and Defects** checkbox in the Model Wizard. This Pattern is in the **Core Extensions Technology** group, and provides:

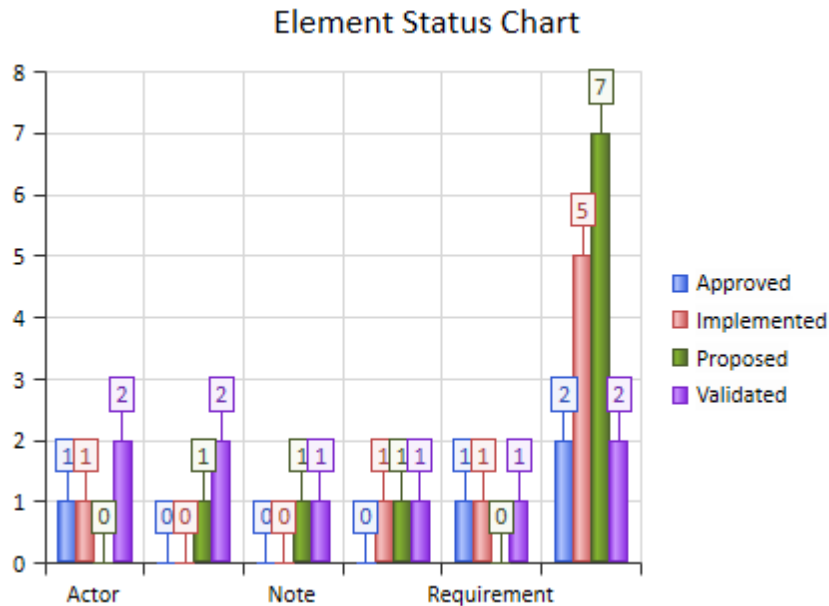
- Two Model View Charts
- A Pie Chart
- A 2D Bar Chart and
- A Time Series Chart

#### Initial Appearance

The Chart element initially displays as a 'normal' element, as shown:



However, depending on which type of Chart icon you select, the element either immediately transforms to the Chart and its Properties dialog automatically displays, or you select to display the Properties dialog and, once you identify the Chart type and contents, the element becomes the Chart you have defined; for example:



This Chart image is dynamically generated - it shows the selected properties of the model elements at the moment:

- The definition is saved, or
- The element's parent diagram is opened, or
- You right-click on the element in the open diagram and select the **Refresh Chart** or **Refresh List** menu option, or
- For TimeSeries Charts, either when the Cloud server is triggered by the update timetable you have defined in the Chart properties, or when you click on the **Manual Record** button in the Properties dialog

You can add a Chart element to any existing diagram, and any number of these elements to one diagram. To make the Charts clear, easy to locate and simple to report on, you might store your Charts:

- On dedicated Dashboard diagrams, the diagram having the same name as a Chart it contains
- Where Charts are not related or for comparison, with just one Chart element per diagram (the Chart depictions are, however, very clear within quite small Chart elements)
- Within a child Package of the Packages that they report on

### Define Chart

You define the Chart represented by an element using the element's Properties dialog. On the General page of the dialog, notice that the element has a stereotype: *Chart*, *TimeSeriesChart* or *ModelView*.

You define a Time Series Chart and a Model View Chart each in a specific way. You define a Standard (Pie or Bar) Chart in three stages:

1. Double-click on the element to open the Properties dialog, and (if the Chart does not already have a specific name) on the **General** page type in the function of the Chart as its name; for example, **Element Status Chart**.
2. Select the **Chart Details | Source** page and define the type of Chart, the type of data you want the Chart to show, and the source of that data.

3. Select the **Chart Details | Appearance** page and define exactly how you want the Chart to display on the screen and in reports.

#### Learn more

- [Define a Model View Chart](#)<sup>[2765]</sup>
- [Define a Time Series Chart](#)<sup>[2767]</sup>
- [Standard Chart Data](#)<sup>[2770]</sup> (Details)
- [Chart Appearance](#)<sup>[2774]</sup>
- [Including Charts in Reports](#)<sup>[2788]</sup>

### 19.4.2 Define a Model View Chart

Your model contains a great deal of information on both project design and project management, and a major benefit of the Chart feature is providing the facility to capture, summarize and present that information in the form of a Model View table that you can operate on both as a single element and as a list of selectable items. On one diagram you can put a number of Model View elements reporting on different aspects of the model, to create a tailored automatic status review of your project.

For a Model View Chart, the type is defined in the template and the content is defined in the SQL Query that you create as the data source. After you generate the Chart, you can perform a number of operations on the data it displays.



**Access**    **Double-click on Chart element (opens directly to Chart Details | Source > Custom SQL)**  
**Right-click on Chart element | Edit List View**

#### Specify Content

Field/ Button	Action	See also
data panel	Type in the SQL Query to define what type of data to extract from the model, and what structures in the model to extract it from.  For example: <code>SELECT t_object.Name, t_object.Status, t_object.Author FROM t_object WHERE t_object.Object_Type = 'Change'</code>  This Query returns a list of Change elements, showing the element name, status and author.	<a href="#">Create Search Definitions</a> <sup>[711]</sup>
<b>OK</b>	Click on this button to close the dialog and generate the Chart.	
<b>Cancel</b>	Click on this button to abort the changes and close the dialog.	

#### Edit Chart Entries

Having created the Model View Chart, you can drag the borders of the element to a suitable display size, manipulate the display as for other Search views, and work on individual entries through the context menu.

Context Menu Option	Action	See also
<b>Exit Edit Mode</b>	Close the Chart as a list of accessible entries and revert to the Chart as a single element.	
<b>Properties</b>	Open the Properties dialog for the selected item in the Model View.	
<b>Copy Selected to Clipboard</b>	Copy the selected item to the clipboard.	
<b>Documentation</b>	<p>Display options to generate a report on the selected item or items, either as:</p> <ul style="list-style-type: none"> <li>• One report on all selected items, or</li> <li>• A separate report for each of the selected items</li> </ul> <p>In either case, the Generate Documentation dialog displays. If you have selected to generate separate reports, the Generate Documentation dialog displays separately for each report in turn.</p>	<a href="#">Generate Documentation</a> 
<b>Create (Edit/Delete) Linked Document</b>	<p>Open the Linked Document Editor to create (or edit) a linked document on the selected item.</p> <p>OR</p> <p>Delete the existing linked document on the item.</p>	
<b>Find in Diagrams</b>	<p>Locate the selected item in any diagrams in which it is used.</p> <p>If the item is only used in one diagram, the display switches to that diagram with the item highlighted on that diagram.</p> <p>If the item is not used in any other diagram, or is used in more than one diagram, the Element Usage dialog displays.</p>	<a href="#">Show Element Use</a> 
<b>Find in Project Browser</b>	Locate and highlight the selected item in the Project Browser.	

#### Notes

- Provided that the SQL Query contains the parameter *ea\_guid AS CLASSGUID*, you can also drag items out of the list and drop them onto the diagram containing the Chart element, in the same way as you would from the Project Browser

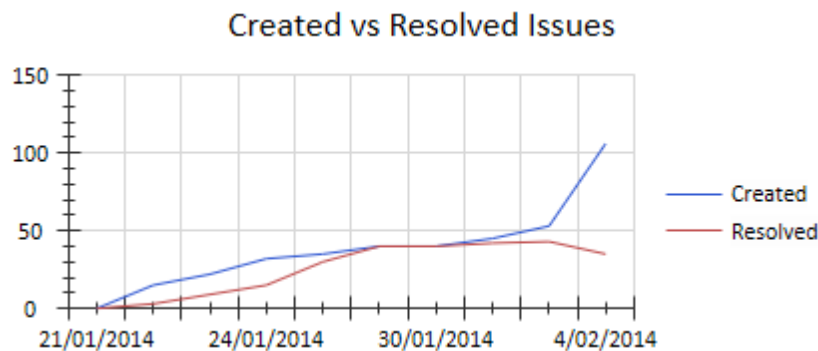
Learn more

- [Customizing the Search View](#)<sup>[708]</sup>
- [Paste from Project Browser](#)<sup>[833]</sup>

**19.4.3 Define a Time Series Chart**

Your model contains a great deal of information on both project design and project management, and a major benefit of the Chart feature is providing the facility to capture, summarize and present that information. Using the **Time Series** Chart you can record changes in **multiple series** of data over time. Each series corresponds to an SQL Query that you can either run **manually** or update **automatically** on a schedule run by the Cloud service. The value of each series is plotted on the line graph on the day or at the time it was collected.

As an example, you might generate the Chart manually on different days, running two Queries showing Issues created and Issues resolved, for comparison. The Chart might resemble this:



After defining the **content** of the Time Series Chart, you can modify its **appearance** on the Appearance page of the element's Properties dialog.

**Access**    **Double-click on Chart element | Chart Details | Source**

Define Content


Field/Button	Action	See also
<b>Package</b>	To generate the Chart on the entire project, leave this field blank or click on the ( ... ) <b>Browse</b> button and select the <b>&lt;none&gt;</b> entry.  Otherwise, click on the <b>Browse</b> button and select the specific Package to generate the Chart on.	<a href="#">Select &lt;item&gt; dialog</a> <sup>[994]</sup>
<b>Include Child Packages</b>	If you have specified a Package, select this checkbox to extract information from the child Packages of that Package.  Clear the checkbox to ignore any child Packages when extracting information from the parent Package.	
<b>Data Point Interval</b>	Click on the down-arrow and select either:	

Field/Button	Action	See also
	<ul style="list-style-type: none"> <li>• <b>Manual</b>, to generate the Chart manually at any time or date you need; the <b>Manual Record</b> button is enabled</li> <li>• A time interval at which the Cloud Server will automatically regenerate the Chart</li> </ul>	
<b>Exclude Weekends</b>	<p>Enabled when the <b>Data Point Interval</b> field is set to <b>Daily</b>.</p> <ul style="list-style-type: none"> <li>• Select the checkbox to refresh the Chart on each week day except for Saturday and Sunday</li> <li>• Clear the checkbox to refresh the Chart on each week day including Saturday and Sunday</li> </ul>	
<b>Run On</b>	<p>Enabled when the <b>Data Point Interval</b> field is set to <b>Weekly</b> or <b>Monthly</b>. If that field is set to:</p> <ul style="list-style-type: none"> <li>• <b>Weekly</b> - click on the down arrow and select the day of the week on which to refresh the Chart</li> <li>• <b>Monthly</b> - click on the down arrow and select the date in the month on which to refresh the Chart</li> </ul>	
<b>Series</b>	Lists the SQL Queries that will be run whenever the Chart is regenerated. This panel also shows the date on which the Query was last executed, and the number of records displayed as a result.	
<b>Edit</b>	<p>Click on an SQL Query name in the Series list and click on this button to edit the Query.</p> <p>The Edit Series dialog displays; see the <i>Edit Series Dialog</i> table below.</p>	
<b>Add</b>	<p>Click on this button to add a new SQL Query to the Series list, to be executed when the Chart is regenerated.</p> <p>The Edit Series dialog displays; see the <i>Edit Series Dialog</i> table below.</p>	
<b>Remove</b>	<p>Click on a SQL Query in the Series list and click on this button to delete the Query as a source of information for the Chart.</p> <p>A confirmation prompt displays. Click on the <b>Yes</b> button to delete the Query.</p>	
<b>Manual Record</b>	<p>If you have set the <b>Data Point Interval</b> field to manual, click on this button to regenerate the Chart.</p> <p>The <b>Results</b> column in the Series list updates (if changes have occurred); the Chart itself does not change until you click on the <b>OK</b> button to close the dialog.</p>	
<b>OK</b>	Click on this button to scan the specified Packages, close the dialog and	




Field/Button	Action	See also
	generate the Chart.	
<b>Cancel</b>	Click on this button to abort the changes and close the dialog.  The aborted changes include updates to the <b>Results</b> column if you have clicked on the <b>Manual Record</b> button.	

### Edit Series Dialog

Field/Button	Action	See also
<b>Examples</b>	(Optional) Click on the drop-down arrow and select an example SQL Query to use intact or to edit to suit your requirements.	
<b>Series Name</b>	Type a name for the SQL Query, to use as a reference to the Query in the <b>Series</b> list on the Source page.	
<b>Query</b>	Type the text of the SQL Query.	<a href="#">Create Search Definitions</a> 
<b>Test Query</b>	(Optional) Click on this button to test the Query you have written or updated.  A message displays to confirm that the Query was successful or to identify that problems exist.	
<b>OK</b>	Click on this button to save the Query, close the dialog and add or update the Query in the <b>Series</b> list on the Source page.	
<b>Cancel</b>	Click on this button to abort the changes, close the dialog and return to the Source page.	

### Learn more

- [Charts](#) 
- [Chart Elements](#) 
- [Chart Appearance](#) 
- [Time Series Chart](#) 

### 19.4.4 Standard Chart Data

When you have created a Standard Chart element (for a Pie Chart or Bar Chart), the next step is to define:

- What type of Chart to create
- What data the Chart will collate and present
- What the source of the data is

You define these aspects on the Chart element's Properties dialog.

**Access**    **Double-click on Chart element | Chart Details | Source**

#### Define Chart Data

Task	Action	See also
<b>Define Chart Type</b>	<p>In the <b>Type</b> field, click on the drop-down arrow and click on the basic type of Chart to create (<b>Pie</b>, <b>2D Bar</b> or <b>3D Bar</b>).</p> <p>After you have specified the data type and source (below), you can further refine the Chart type using the Appearance page of the dialog.</p>	<a href="#">Chart Appearance</a> <small>[2774]</small>
<b>Define Data Type</b>	<p>In the <b>Series</b> field, click on the <b>Browse</b> icon ( ... ) and select the primary object type and property to show in the Chart, from the menu and sub-menus. For example:</p> <p><b>Element</b> and <b>Status</b></p> <p>For a <b>Bar Chart</b>, whether <b>2D</b> or <b>3D</b>, you can optionally select a secondary object and property by which to impose a grouping of the data on the Chart. In the <b>Group By</b> field, click on the <b>Browse</b> icon ( ... ) and select the secondary object type and property by which to group the data. For example:</p> <p><b>Element</b> and <b>ObjectType</b></p> <p>This combination produces a Chart that shows the numbers of elements having each status, grouped by element type.</p> <p><u>Notes</u></p> <ul style="list-style-type: none"> <li>• You cannot apply grouping to a Pie Chart, so if you select <b>Pie</b> in the <b>Type</b> field, the <b>Group By</b> field is disabled</li> <li>• SQL Searches and CSV strings define the data selection, so if you select either of these as the data source, the <b>Series Data</b> and <b>Group By</b> fields are disabled</li> </ul>	
<b>Define Data Source</b>	<p>The data you present in your Chart is likely to come from within your model, although you can also copy a simple <b>Comma Separated Value (CSV) spreadsheet</b> from an external file into the dialog, for display.</p> <p>You have several options for locating the data within your model. You can search:</p>	<a href="#">External Data</a> <small>[2773]</small>  <a href="#">Source</a>

Task	Action	See also
	<ul style="list-style-type: none"> <li>• Within one or more specific Packages</li> <li>• Across the whole model</li> <li>• As defined by a tailored SQL Query</li> </ul> <p>You can also filter the data you are collecting, to highlight specific features of the model. Again, as the CSV and SQL definitions specify the data characteristics themselves, the filtering option is not enabled for them.</p> <p>An advantage of defining the search or CSV table within the element is that the search or data is available with the Chart element where all users can access it, rather than being restricted to the workstation of the person who created the data Chart.</p> <p>Fully specifying and committing one type of data source will clear the definition of any previously-used data source.</p>	<a href="#">Package</a> <sup>[2771]</sup>  <a href="#">Custom Query</a> <sup>[2773]</sup>  <a href="#">Element Filters in Charts</a> <sup>[2772]</sup>

#### Learn more

- [Charts](#) <sup>[2762]</sup>
- [Chart Elements](#) <sup>[2763]</sup>

#### 19.4.4.1 Source Package

Your model contains a great deal of information on both project design and project management, and a major benefit of the Chart feature is providing the facility to capture, summarize and present that information from:

- A specific Package in the model
- A number of specific Packages in the Model, or
- Across the whole model

You can also select to include information from just a Package itself, or from the Package and all its child Packages.

If you want to further refine the data to omit certain quantities or highlight others, you can apply element filters to the data from the selected Packages.

**Access**    **Double-click on Chart element | Chart Details | Source > Package**

#### Specify Source Packages

Field/Button	Action	See also
<b>Add</b>	<p>Click on this button and select either:</p> <ul style="list-style-type: none"> <li>• <b>Add Package</b> to identify a specific Package from which to extract the information - the Select a Toolbox Profile Package dialog displays, from which you select the Package; or</li> <li>• <b>Search Model</b> to extract the information from the whole model; the text</li> </ul>	<a href="#">Select &lt;item&gt; dialog</a> <sup>[994]</sup>

Field/Button	Action	See also
	<p><i>Model</i> in the <b>Source</b> list confirms your selection</p> <p>You can click on the <b>Add</b> button a number of times, to add several specific Packages.</p>	
<b>Remove</b>	Click on a Package name in the Source list and click on this button to remove the Package as a source of data for the Chart.	
<b>Include Child Packages</b>	<p>Select this checkbox against a Package to extract information from its child Packages.</p> <p>Clear the checkbox to ignore any child Packages when extracting information from the parent Package.</p>	
<b>OK</b>	Click on this button to scan the specified Packages, close the dialog and generate the Chart.	
<b>Cancel</b>	Click on this button to abort the changes and close the dialog.	

#### Learn more

- [Charts](#) <sup>[2762]</sup>
- [Chart Elements](#) <sup>[2763]</sup>
- [Standard Chart Data](#) <sup>[2770]</sup>
- [Element Filters](#) <sup>[2772]</sup>

#### **19.4.4.1.1 Element Filters in Standard Charts**

If you are creating a Pie or Bar Chart of object properties from the **Packages** within your model, you can refine the data that is presented in the Chart to include only elements that have specific properties rather than every element encountered, using **element filters**.

Whilst you apply the element filters using a special page of the Chart element Properties dialog, this page is identical in use to the Element Filters tab in the Document Template Designer.

**Access**    **Double-click on Chart Object element | Chart Details | Element Filters**

#### Notes

- Element filters are not available for Charts constructed using an SQL Query or CSV file contents

#### Learn more

- [Element Filters](#) <sup>[2660]</sup>

### 19.4.4.2 Custom Query

Your model contains a great deal of information on both project design and project management, and a major benefit of the Chart feature is providing the facility to search through your model and extract specific aspects of that information using your own SQL Query, within the Chart element.

**Access** Double-click on Chart element | Chart Details | Source > Custom SQL

#### Extract Information Using SQL Query

Field/Button	Action	See also
panel	<p>Type in your SQL Query.</p> <p>The <b>Select</b> statement must include the alias <b>Series</b>. For Bar Charts you can also group the results using the alias <b>GroupName</b>. For example:</p> <pre>Select t_object.Status AS Series, t_object.Author AS Groupname from t_object</pre> <p>The field provides the facilities of the Common Code Editor, such as intellisense.</p>	<a href="#">Code Editor Functions</a> <sup>[2157]</sup>
OK	Click on this button to execute the query, close the dialog and generate the Chart.	
Cancel	Click on this button to abort the changes and close the dialog.	

#### Learn more

- [Charts](#) <sup>[2762]</sup>
- [Chart Elements](#) <sup>[2763]</sup>
- [Standard Chart Data](#) <sup>[2770]</sup>

### 19.4.4.3 External Data

Although your model contains a great deal of information on both project design and project management, you might want to check or present the status of some external spreadsheet data for comparison or for review before importing it into your model. A major benefit of the Chart feature is providing the facility to quickly paste in and display, in the appropriate Chart format, the contents of an external CSV file.

Before you start, copy the content of the CSV file to the clipboard.

**Access** Double-click on Chart element | Chart Details | Source > CSV

#### Review External CSV Data

Field/Button	Action	See also
panel	Right click on the panel and select the <b>Paste</b> option to paste the contents of the CSV file.  If necessary, edit the data so that the first column represents the series object (the objects listed along the X axis of the Chart). You need perform only very basic editing on the text.	
<b>OK</b>	Click on this button to process the data, close the dialog and generate the Chart.	
<b>Cancel</b>	Click on this button to abort the changes and close the dialog.	

#### Learn more

- [Charts](#)<sup>[2762]</sup>
- [Chart Elements](#)<sup>[2763]</sup>
- [Standard Chart Data](#)<sup>[2770]</sup>

### 19.4.5 Chart Appearance

Each of the Chart formats provided on the system - Model View, Time Series, 2D Bar Chart, 3D Bar-Chart and Pie Chart - has a default appearance, but you can modify this appearance to better suit the information you are presenting. You can, for example, change the:

- Configuration
- Direction of shading
- Orientation
- Use of a legend or key
- Color intensity and/or
- Position of labels, if you choose to display them

The options vary between the Chart types, and are described separately.

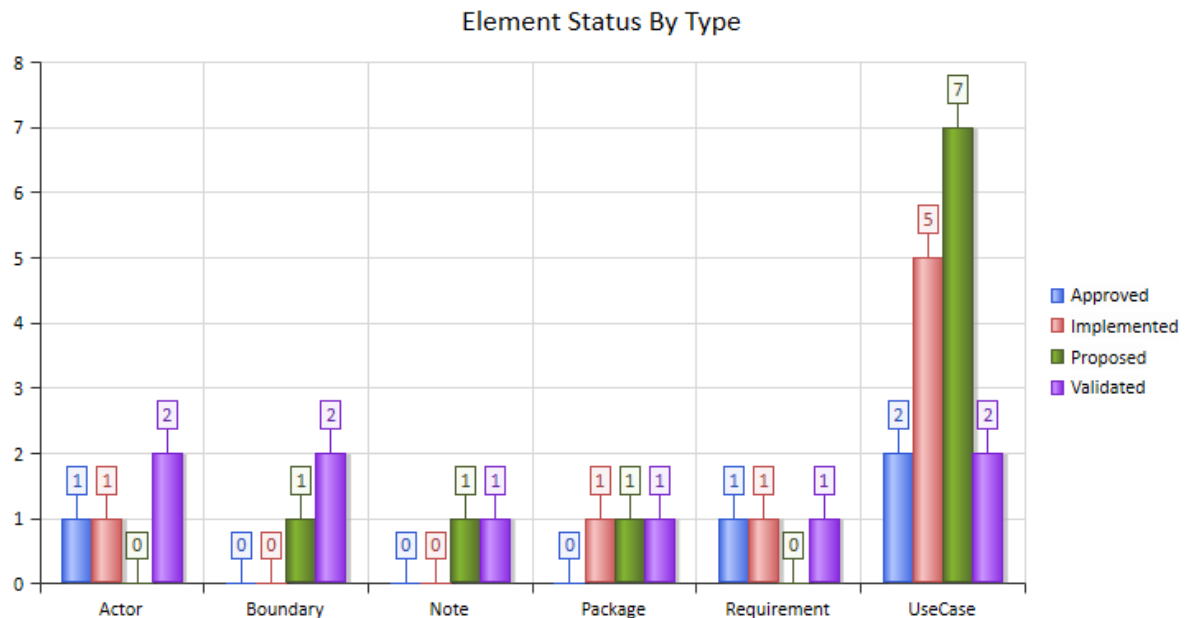
**Access**    **Double-click on Chart Object element | Chart Details | Appearance**

#### Learn more

- [Define a Model View Chart](#)<sup>[2765]</sup>
- [Time Series Chart](#)<sup>[2785]</sup>
- [2D Bar Chart](#)<sup>[2775]</sup>
- [3D Bar Chart](#)<sup>[2778]</sup>
- [Pie Chart](#)<sup>[2782]</sup>
- [Including Charts in Reports](#)<sup>[2788]</sup>

### 19.4.5.1 2D Bar Chart

A 2-dimensional Bar Chart can have this appearance:

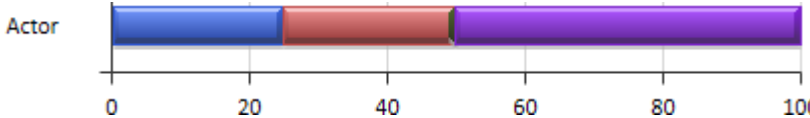

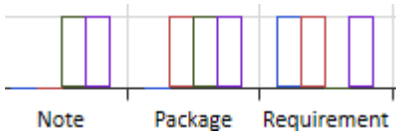


You define this appearance - or its alternatives - on the Appearance page of the element Properties dialog, after having set the **Type** field on the Source page to **2D Bar**. Each change in setting is immediately illustrated by an example Chart on the Appearance page.

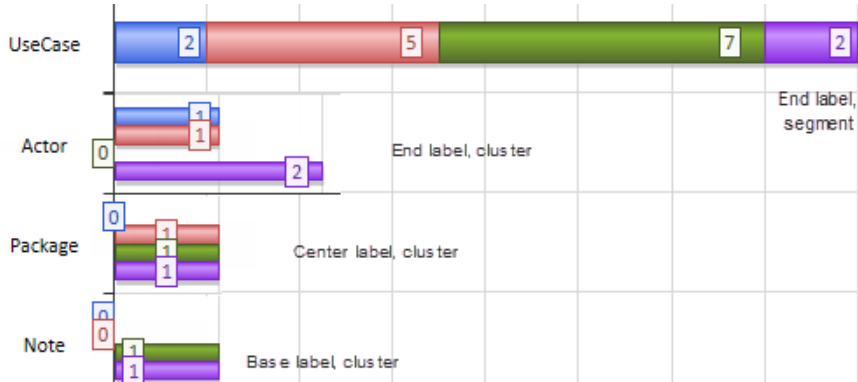
**Access** [Double-click on Chart element](#) | [Chart Details](#) | [Appearance](#)

#### Define 2D Bar Chart Appearance

Field	Action	See also
<b>Category</b>	<p>Click on the drop down arrow and select the Bar Chart category:</p> <ul style="list-style-type: none"> <li><b>Column</b> (the default, as illustrated above) or</li> <li>Horizontal <b>Bar</b></li> </ul> <div style="text-align: center;"> </div>	
<b>Type</b>	<p>Click on the drop down arrow and select how the object Type is represented:</p> <ul style="list-style-type: none"> <li>As a <b>Cluster</b> of columns (the default, illustrated above) with each property</li> </ul>	

Field	Action	See also
	<p>represented by a column or bar in the cluster</p> <ul style="list-style-type: none"> <li>As a <b>Stack</b>, where each object is represented by a single column or bar and each property is represented by a segment of that bar; the bars are <b>different</b> lengths, depending on the sum of the values of the segments</li> <li>As a <b>100% Stack</b>, where the single bars or columns are of <b>equal</b> lengths (100%) and the segments represent the percentage contribution of each property</li> </ul>  <p>This option is grayed out if the <b>Group By</b> properties field on the <b>Chart Details   Details</b> page has no value. The Chart then displays as a simple column or horizontal bar for each object.</p>	
<b>Gradient</b>	<p>Click on the drop down arrow and select the direction of the color gradient in each bar:</p> <ul style="list-style-type: none"> <li><b>Solid Color</b> (no color gradient)</li> <li><b>Horizontal</b> (color fades from bottom to top)</li> <li><b>Vertical</b> (color fades from left to right)</li> <li><b>Pipe</b> (the default, as shown above; color fades from edges to center)</li> <li><b>Bevel</b> (each bar appears raised, with color shading)</li> </ul> 	
<b>Transparency</b>	<p>Click on the slider and drag it across until the illustration shows the degree of transparency you require.</p> <p>The left extreme (no transparency) sets the bars to full fill color (as shown above); the right extreme (full transparency) sets the bars to colored edges and no fill.</p> 	
<b>Show Data Labels</b>	<p>This checkbox defaults to selected, to display the column or segment values as a framed label (as shown above).</p> <p>Deselect the checkbox to hide the labels (as shown for the <b>Gradient</b>, <b>Type</b> and <b>Transparency</b> fields).</p>	
<b>Display Shadow</b>	<p>This checkbox defaults to selected, to display the column or bar with a gray shadow (as shown above).</p>	



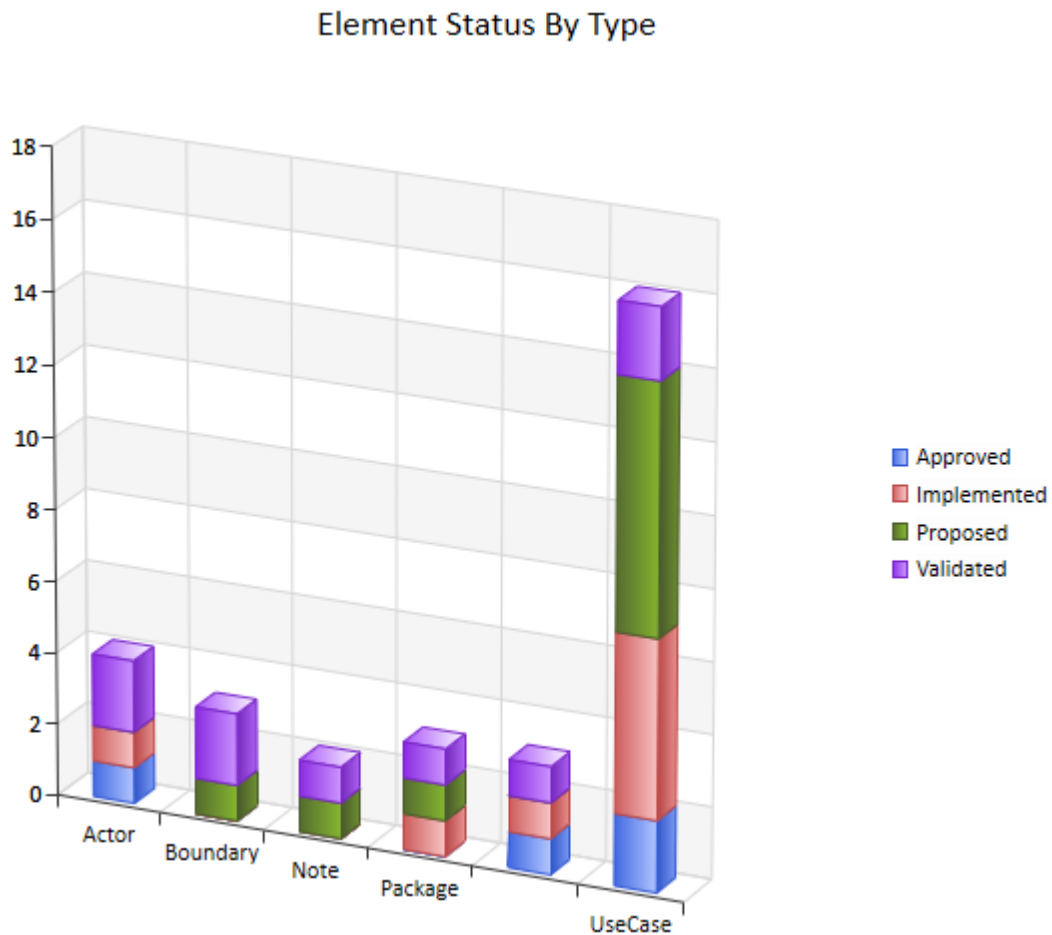
Field	Action	See also
	Deselect the checkbox to omit the shadow. (At a transparency of approximately 50% through to 100% the bars do not have a shadow either.)	
<b>Display Legend</b>	Select this checkbox to show the meaning of the Chart bars, columns or segments as a legend to the right of the diagram.  Deselect the checkbox to hide the legend.	
<b>Label Position</b>	<p>If <b>Show Data Labels</b> is selected, click on the drop down arrow and select the position in which to display the labels relative to the column, bar or segment:</p> <ul style="list-style-type: none"> <li>• <b>Center</b> - the label is shown in the center of the column, bar or segment</li> <li>• <b>Inside End</b> - the label is shown inside the top of the column or segment, or at the right end of the bar</li> <li>• <b>Inside Base</b> - the label is shown inside the base of the column or segment, or at the left end of the bar</li> <li>• <b>Outside End</b> - the label is shown off the end of the column, bar or segment and linked to it by a line (the default, as shown above)</li> </ul> <p>The first three options are illustrated in this composite picture:</p> 	
<b>OK</b>	Click on this button to apply the changes and close the dialog.	
<b>Cancel</b>	Click on this button to abort the changes and close the dialog.	

#### Learn more

- [Charts](#) <sup>[2762]</sup>
- [Chart Elements](#) <sup>[2763]</sup>
- [Standard Chart Data](#) <sup>[2770]</sup>
- [Chart Appearance](#) <sup>[2774]</sup>

### 19.4.5.2 3D Bar Chart

A 3-dimensional Bar Chart can have this appearance:

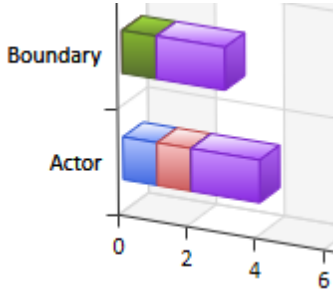
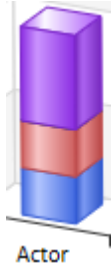
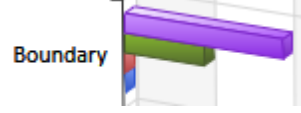
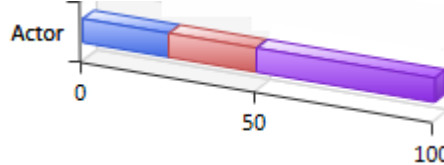


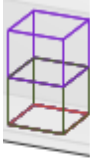

You define this appearance - or its alternatives - on the Appearance page of the element Properties dialog, after having set the **Type** field on the Source page to **3D Bar**. Each change in setting is immediately illustrated by an example Chart on the Appearance page.

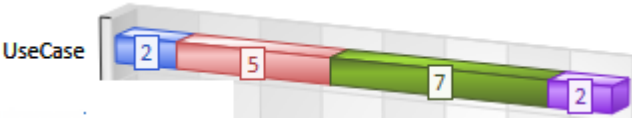
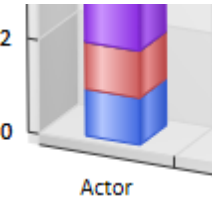

**Access** [Double-click on Chart Object element](#) | [Chart Details](#) | [Appearance](#)

#### Define 3D Bar Chart Appearance

Field	Action	See also
<b>Category</b>	<p>Click on the drop down arrow and select the Bar Chart category:</p> <ul style="list-style-type: none"> <li>• <b>Column</b> (the default, as illustrated above) or</li> <li>• Horizontal <b>Bar</b></li> </ul>	

Field	Action	See also
		
<b>Gradient</b>	<p>Click on the drop down arrow and select the direction of the color gradient in each bar:</p> <ul style="list-style-type: none"> <li>• <b>Solid Color</b> (no color gradient)</li> <li>• <b>Horizontal</b> (color fades from bottom to top)</li> <li>• <b>Vertical</b> (color fades from left to right)</li> <li>• <b>Bevel</b> (color shading from the edges to the center of each face)</li> <li>• <b>Pipe</b> (the default, as shown above; color fades from edges to center)</li> </ul> 	
<b>Type</b>	<p>Click on the drop down arrow and select how the object Type is represented:</p> <ul style="list-style-type: none"> <li>• As a <b>Cluster</b> of columns (the default) with each property represented by a column or bar in the cluster</li> </ul>  <ul style="list-style-type: none"> <li>• As a <b>Stack</b> (illustrated above) where each object is represented by a single column or bar and each property is represented by a segment of that bar; the bars are <b>different</b> lengths, depending on the sum of the values of the segments</li> <li>• As a <b>100% Stack</b>, where the single bars or columns are of <b>equal</b> lengths (100%) and the segments represent the percentage contribution of each property</li> </ul> 	

Field	Action	See also
	<p>This option is grayed out if the <b>Group By</b> properties field on the <b>Chart Details   Details</b> page has no value. The Chart then displays as a simple column or horizontal bar for each object.</p>	
<b>Transparency</b>	<p>Click on the slider and drag it across until the illustration shows the degree of transparency you require.</p> <p>The left extreme (no transparency) sets the bars to full fill color (as shown above); the right extreme (full transparency) sets the bars to colored edges and no fill.</p>  <p>Note</p>	
<b>Label Position</b>	<p>Click on the drop down arrow and select the position in which to display the default data labels relative to the column, bar or segment:</p> <ul style="list-style-type: none"> <li>• <b>Center</b> - the label is shown in the center of the column, bar or segment</li> <li>• <b>Inside End</b> - the label is shown inside the top of the bar or segment</li> <li>• <b>Inside Base</b> - the label is shown inside the base of the bar or segment</li> <li>• <b>Outside End</b> - the label is shown off the end of the bar or segment (the default)</li> </ul> <p>The first option is illustrated in the <b>Show Data Labels</b> description, below; the other three options are illustrated in this composite picture:</p> 	
<b>Show Data Labels</b>	<p>This checkbox defaults to selected, to display the column or segment values as a framed label:</p>	

Field	Action	See also
	 <p>Deselect the checkbox to hide the labels.</p>	
<b>Display Legend</b>	<p>Select this checkbox to show the meaning of the Chart bars, columns or segments as a legend to the right of the diagram.</p> <p>Deselect the checkbox to hide the legend.</p>	
<b>Fill Walls and Floor</b>	<p>Select this checkbox to add darker shading to the end and back walls of the Chart, to provide contrast.</p> <p>Clear the checkbox to leave the walls pale.</p>	
<b>Thick Walls</b>	<p>If the <b>Fill Walls and Floor</b> checkbox is selected, this checkbox is enabled.</p> <p>Leave the checkbox unselected to leave the walls and floor as open, 2-dimensional lines (as illustrated above).</p> <p>Select this check box to show the walls and floor of the Chart as closed, 3-dimensional blocks.</p> 	
	<p>Click on the arrows on this dial to alter the perspective of the reader in viewing the Chart.</p> <ul style="list-style-type: none"> <li>The rotating arrow at the <b>center</b> returns the Chart to its default position (viewed from the upper right front of the Chart); the remaining arrow descriptions identify changes from this position</li> <li>The small <b>top</b> arrow flattens the Chart, as if you were looking at it from higher up and further out</li> <li>The <b>left</b>-pointing arrow rotates the Chart by increments anti-clockwise around the vertical axis, so that you can swing the y-axis of the Chart around to the right of the Chart and back; the end of the Chart pointing 'into' the screen has the end wall visible</li> <li>The <b>right</b>-pointing arrow rotates the Chart in a clockwise direction around the vertical axis</li> <li>The <b>upward</b> broad arrow swings the left rear corner of the Chart <b>out of</b> the screen (from the default position, only one move)</li> <li>The <b>downward</b> broad arrow swings the left rear corner of the Chart <b>into</b> the screen in several increments</li> </ul>	

Field	Action	See also
	<ul style="list-style-type: none"> <li>The small <b>bottom</b> arrow swings the base of the Chart around 'out of the screen', as if you were looking at it from further out and lower down; ultimately, you see the Chart from the base</li> </ul>	
<b>OK</b>	Click on this button to apply the changes and close the dialog.	
<b>Cancel</b>	Click on this button to abort the changes and close the dialog.	

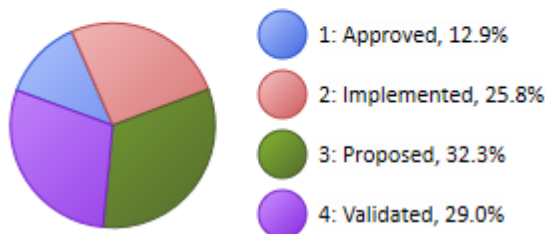
#### Learn more

- [Charts](#) <sup>[2762]</sup>
- [Chart Elements](#) <sup>[2763]</sup>
- [Standard Chart Data](#) <sup>[2770]</sup>
- [Chart Appearance](#) <sup>[2774]</sup>

### 19.4.5.3 Pie Chart

A **Pie Chart** can have this appearance:

Element Status By Type



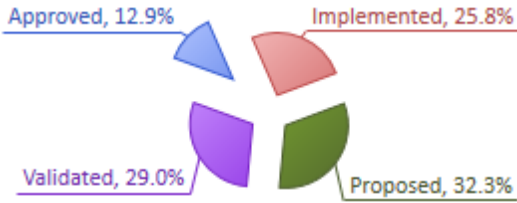

You define this appearance - or its alternatives - on the Appearance page of the element Properties dialog, after having set the **Type** field on the Source page to **Pie**. Each change in setting is immediately illustrated by an example Chart on the Appearance page.

**Access** Double-click on Chart element | Chart Details | Appearance

#### Define Pie Chart Appearance

Field	Action	See also
<b>Category</b>	Click on the drop down arrow and select the Pie Chart category:	

Field	Action	See also
	<ul style="list-style-type: none"> <li>• <b>3D Chart</b> - the Chart displays as a 3-D disc tilted away from you</li> <li>• <b>Doughnut</b> - the Chart displays as a 2-D ring</li> <li>• <b>Doughnut 3D</b> - the Chart displays as a 3-D ring with a rectangular profile</li> <li>• <b>Torus 3D</b> - the Chart displays as a 3-D ring with an elliptical profile</li> <li>• <b>Pie</b> - the Chart displays as a 2-D disc (the default, as illustrated above)</li> </ul>	
<b>Label Position</b>	<p>You can show the meaning and value of each segment of the Chart as a label (see <b>Show Data Labels</b> below); if you want to do this, you can specify where on the Chart the label will display.</p> <p>Click on the drop down arrow and select the position in which to display the labels relative to the segments:</p> <ul style="list-style-type: none"> <li>• <b>Center</b> - the label is shown in the middle of the segment</li> <li>• <b>Inside End</b> - the label is shown inside the rim of the segment,</li> <li>• <b>Inside Base</b> - the label is shown closer to the point of the segment</li> <li>• <b>Outside End</b> - the label is shown outside the rim of the segment and linked to it by a line (the default, as shown for the <b>Exploded</b> field)</li> </ul>	
<b>Show Data Labels</b>	<p>Select the checkbox to display a label against each segment of the Pie Chart.</p> <ul style="list-style-type: none"> <li>• If the <b>Show Index in Labels</b> checkbox <b>is</b> selected, only the index numbers ( <b>1, 2, 3...</b>) display in the labels, keyed to the legend (see <b>Display Legend</b>, below)</li> <li>• If the <b>Show Index in Labels</b> checkbox is <b>not</b> selected, the property name and percentage (such as <i>Approved, 12.9%</i>) display in the labels</li> </ul>	
<b>Gradient</b>	<p>Click on the drop down arrow and select the direction of the color gradient across the Chart:</p> <ul style="list-style-type: none"> <li>• <b>Solid Color</b> (no color gradient)</li> <li>• <b>Diagonal Left</b> (the default, as shown above; color fades from right to left)</li> <li>• <b>Diagonal Right</b> (color fades from left to right)</li> <li>• <b>Radial</b> (color fades from edges to center)</li> <li>• <b>Bevel</b> (dark rim and color fading from the bottom to the top)</li> </ul> <p>The <b>Torus 3D</b> category of Chart relies on a specific color gradient to create the effect of the Torus, therefore changes in the <b>Gradient</b> settings have no effect on Charts of this type.</p>	
<b>Hole Size</b>	<p>This field is available for the <b>Doughnut</b> and <b>Doughnut 3D</b> categories of the Pie Chart.</p> <p>Click on the slider and drag it to expand or contract the size of the hole in the middle of the Chart. The default (and the fixed size of the <b>Torus 3D</b> category) is <b>50%</b></p> <p>If you drag the hole size to <b>0%</b>, the Chart is effectively a <b>Pie Chart</b> or <b>3D Chart</b>.</p>	

Field	Action	See also
<b>Show Index in Labels</b>	<p>If you are displaying the meaning and value of each segment, you can display a legend providing this information (see <b>Display Legend</b>, below) and have a simple index to the legend as a label against each segment.</p> <p>Select the checkbox to display the index (<b>1, 2, 3...</b>) as the label text</p> <p>Clear the checkbox to display more complete information in the label.</p>	
<b>Exploded</b>	<p>Select the checkbox to pull each segment of the Chart out away from the other segments, with the 'first' segment separated out even further.</p> 	
<b>Fit Diagram Area</b>	<p>As you re-size your Chart element, the Chart width and height normally change with the element but stay in proportion.</p> <p>Leave this checkbox unselected to maintain this behavior.</p> <p>Select this checkbox to expand the Chart to fit the element space, even if the Chart has to elongate disproportionately up and down or sideways.</p>	
<b>Display Legend</b>	<p>You can select to show the meaning and value of the Chart segments on the Chart itself, as labels, or you can display some or all of this information as a legend to the right of the diagram.</p> <p>Select this checkbox to display a legend for the Chart.</p> <p>Deselect the checkbox to hide the legend.</p>	
	<p>For the <b>3D Chart</b> and <b>Doughnut 3D</b> categories:</p> <ul style="list-style-type: none"> <li>The small <b>top</b> (up) arrow swings the Chart toward horizontal, so you are increasing the side view; the extreme is to show the front edge of the Chart</li> <li>The small <b>bottom</b> (down) arrow swings the Chart towards vertical, so you are increasing the top view; the extreme is to show a <b>Pie Chart</b> or <b>Doughnut Chart</b></li> </ul> <p>For all categories of Pie Chart:</p> <ul style="list-style-type: none"> <li>The rotating arrow at the <b>center</b> returns the Chart to its default perspective</li> <li>The <b>left</b>-pointing arrow rotates the Chart by 24-degree increments anti-clockwise</li> <li>The <b>right</b>-pointing arrow rotates the Chart in 24-degree increments clockwise</li> </ul>	



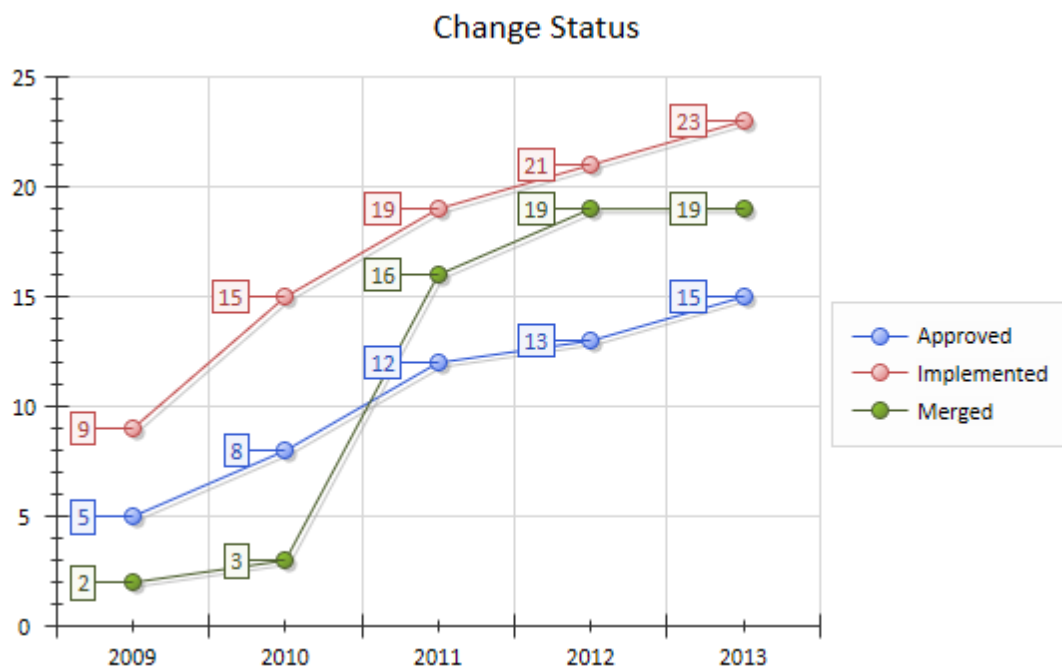
Field	Action	See also
<b>OK</b>	Click on this button to apply the changes and close the dialog.	
<b>Cancel</b>	Click on this button to abort the changes and close the dialog.	

#### Learn more

- [Charts](#) <sup>[2762]</sup>
- [Chart Elements](#) <sup>[2763]</sup>
- [Standard Chart Data](#) <sup>[2770]</sup>
- [Chart Appearance](#) <sup>[2774]</sup>

#### 19.4.5.4 Time Series Chart

A **Time Series Chart** can have this appearance:



You define this appearance - or its alternatives - on the Appearance page of the element Properties dialog, after having set the data source, content and refresh interval on the Source page. Each change in appearance setting is immediately illustrated by an example Chart on the Appearance page.

**Access**    **Double-click on Chart element | Chart Details | Appearance**

#### Define Time Series Chart Appearance

Field/Option/ Button	Action	See also
<b>Type</b>	<p>Click on the drop down arrow and select how the object Type is represented:</p> <ul style="list-style-type: none"> <li>As a <b>Stacked Line</b>, where the summed values of the data points for a time interval are represented by the positions of the markers, and the label on each marker identifies the contribution that data point made to the total; for example, at a given time there are 3 Approved changes, 5 Implemented changes and 9 Merged changes (as recorded in the marker labels), so the total number of changes is 17 and the Chart shows the markers at 3, 8 (3+5) and 17 (3+5+9)</li> <li>As a <b>100% Stacked Line</b>, where the summed values of data points total 100%, the markers for each data point indicate the percentage contribution each data point made to that total, and the label on each marker indicates the value of the data point</li> <li>As a <b>Stacked Spline</b>, similar to a Stacked Line but instead of the connecting lines being straight the trend of the line is extrapolated into a curve</li> <li>As a <b>Step line</b>, where the markers represent the value of the data points and the connecting lines are horizontal until the data value changes at the next sample point; that is, there is no assumption of change <b>between</b> the sample points</li> <li>As a <b>Line</b> (the default, illustrated above), where a marker indicates the value of each data point and the data points for an object are connected by a straight line</li> </ul>	
<b>Show Data Markers</b>	<p>Defaults to selected, to show a marker symbol to highlight the data point.</p> <p>Deselect the checkbox to hide the data markers.</p>	
<b>Marker Shape</b>	<p>Disabled if the <b>Show Data Markers</b> checkbox is not selected.</p> <p>Click on the drop-down arrow and select a shape for the data markers. The options are:</p> <ul style="list-style-type: none"> <li><b>Circle</b></li> <li><b>Triangle</b></li> <li><b>Rectangle</b></li> <li><b>Diamond</b></li> </ul>	
<b>Marker Size</b>	<p>Disabled if the <b>Show Data Markers</b> checkbox is not selected.</p> <p>Click on the drop-down arrow and select the size of the data markers, in pixels. Options are:</p> <ul style="list-style-type: none"> <li><b>7</b></li> <li><b>10</b></li> <li><b>15</b></li> <li><b>20</b></li> </ul>	
<b>Show Data Labels</b>	<p>This checkbox defaults to selected, to display each data value as a framed label (as shown above).</p>	

Field/Option/ Button	Action	See also
	Deselect the checkbox to hide the labels, if you want to observe the trend rather than the actual values.	
<b>Label Angle</b>	<p>Disabled if the <b>Show Data Labels</b> checkbox is not selected.</p> <p>Click on the drop down arrow and select the angle at which to display the data labels on the data points. Options are:</p> <ul style="list-style-type: none"> <li>• <b>-90</b> - parallel to and left of the data point</li> <li>• <b>-45</b> - at a 45 degree angle above and to the left of the data point</li> <li>• <b>0</b> - directly above the data point</li> <li>• <b>45</b> - at a 45 degree angle above and to the right of the data point</li> <li>• <b>90</b> - parallel to and right of the data point</li> </ul> <p>If there is any overlap of labels the position of those labels adjusts to ensure a clear display.</p>	
<b>Display Shadow</b>	<p>This checkbox defaults to selected, to display the line and markers with a gray shadow (as shown above).</p> <p>Deselect the checkbox to omit the shadow.</p>	
<b>Line width</b>	<p>Click on the down-arrow to change the width of the line on the Chart. The options are:</p> <ul style="list-style-type: none"> <li>• <b>1 Pixel</b></li> <li>• <b>2 Pixels</b></li> <li>• <b>3 Pixels</b></li> <li>• <b>4 Pixels</b></li> <li>• <b>5 Pixels</b></li> </ul>	
<b>Line Style</b>	<p>Click on the down arrow and select the format of the line. Options are</p> <ul style="list-style-type: none"> <li>• <b>Solid</b> (———)</li> <li>• <b>Dash</b> (- - -)</li> <li>• <b>Dot</b> (· · ·)</li> <li>• <b>Dash Dot</b> (- · - ·)</li> <li>• <b>Dash Dot Dot</b> (- · · - · ·)</li> </ul>	
<b>OK</b>	Click on this button to apply the changes and close the dialog.	
<b>Cancel</b>	Click on this button to abort the changes and close the dialog.	

**Learn more**

- [Charts](#)<sup>[2762]</sup>
- [Chart Elements](#)<sup>[2763]</sup>
- [Define a Time Series Chart](#)<sup>[2767]</sup>
- [Chart Appearance](#)<sup>[2774]</sup>

**19.4.6 Including Charts in Reports**

If you want to distribute your Charts in the form of a printed or printable report, you can do so in two ways:

- **Print the diagram** containing the Chart (for small and/or informal distributions of a Chart or group of Charts)
- Generate a **Diagrams-Only report** to show all Charts in the selected Package, without any other model or element details

If you want to show the Chart **definitions** and other properties of each Chart element, you can generate other types of document report on the Packages containing the diagrams and elements.

It is also possible to include the Charts (in the form of a normal diagram) as a feature of any web documentation that you might generate on the model or Packages.

**Learn more**

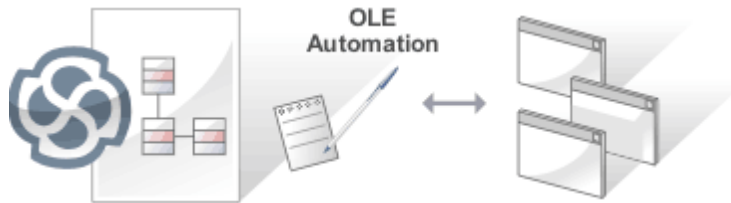
- [File Menu](#)<sup>[79]</sup>
- [Diagrams Only Report](#)<sup>[2741]</sup>
- [Generate Documentation](#)<sup>[2644]</sup>
- [Web Reports](#)<sup>[2744]</sup>

**Part**

---

**XX**

## 20 Automation and Scripting



In Enterprise Architect you can extend and automate the facilities and functionality through the use of Scripts, the Enterprise Architect Object Model and the Enterprise Architect Add-in Model:

Topic	Link
Scripts	<a href="#">Scripts</a> <sup>[2791]</sup>
The Enterprise Architect Object Model	<a href="#">The Enterprise Architect Object Model</a> <sup>[2804]</sup>
The Enterprise Architect Add-In Model	<a href="#">The Enterprise Architect Add-In Model</a> <sup>[3010]</sup>

### Learning Center topics

- **Alt+F1 | Enterprise Architect | Automation | Automation | Introduction to Automation**
- **Alt+F1 | Enterprise Architect | Automation | Addins**
- **Alt+F1 | Enterprise Architect | Automation | Scripting**

## 20.1 Scripting



Enterprise Architect implements a flexible and easy to use scripting capability based on both industry standard Javascript and Microsoft JScript and VBScript. Using a built in '**Repository**' object, you can programmatically inspect and/or modify elements within your currently open model. There are built in tools to edit, run, debug and manage your scripts.



Scripts can be used for a wide range of purposes, including:

- Inspecting and reporting on model and element composition
- Modifying and updating element properties
- Running queries to obtain extended model information
- Modifying diagram layouts
- Being called from report document templates to populate reports
- Creating and implementing process workflows
- Being included in MDG Technologies to augment domain specific languages

Scripting can help you get the custom information you need from your model, verify the structure of your model and help you automate time consuming manual tasks.

### Topics

Topic	Detail	See also
<b>Usage</b>	<p>Scripts executed in Enterprise Architect have access to the currently open model and are a powerful tool for querying and updating the model in situations that would otherwise require you to perform time consuming and repetitive GUI tasks.</p> <p>Enterprise Architect supports management of scripts using the following script engines:</p> <ul style="list-style-type: none"> <li>• JavaScript</li> <li>• Microsoft JScript</li> <li>• Microsoft VBScript</li> </ul> <p>The management interface for Scripting is the Scripting window, which contains the:</p> <ul style="list-style-type: none"> <li>• Script Tree View (Scripts tab), which you use to review, create and edit scripts</li> <li>• Script Console (Console tab), which you use to operate on an executing script</li> </ul> <p>Scripts are managed in groups: the first group in the list is always Local</p>	<p><a href="#">Scripts Tab</a> <sup>[2792]</sup></p> <p><a href="#">Console Tab</a> <sup>[2793]</sup></p> <p><a href="#">Script Group Properties</a> <sup>[2794]</sup></p>

Topic	Detail	See also
	<p>Scripts, which are files in the Scripts subdirectory of the Enterprise Architect installation - any instance of Enterprise Architect that has a currently open model can see these scripts; you cannot create, edit, drag-and-drop or delete Local scripts.</p> <p>All other groups are User Scripts, which you create yourself; a user group can be <b>one of several types</b>, each of which applies a template and certain conditions to the scripts you create within that group.</p> <p>User scripts are only visible inside the model in which they were created; the contents of the scripts are stored with the model, although they can be saved to the file system easily using the Script Editor.</p> <p>During the development of a script, you can debug it as you test it and analyse the execution; for example, by recording a Sequence diagram of the script execution, and halting execution to view local variables.</p>	<p><a href="#">Script Editor</a>  <sup>[2798]</sup></p> <p><a href="#">Script Debugging</a>  <sup>[2801]</sup></p>

### Notes

- This facility is available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions
- If you intend to use the Scripting facility under Crossover/WINE, you must also install Internet Explorer version 6.0 or above



### Learning Center topics

- **Alt+F1 | Enterprise Architect | Automation | Scripting**

## 20.1.1 Scripts Tab

**Access**   **Tools | Scripting > Scripts**

### Topics









Topic	Detail	See also
<b>Usage</b>	<p>The Scripts tab is composed of a toolbar and a view of all scripts according to group.</p> <p>To execute a script, press ( <b>Ctrl</b> ) while you double-click on the script name.</p> <p>To write or edit a user script, <b>double-click on the script name</b> to display the Script Editor; this usually displays a script template, determined by the user group type as assigned on creation or on the Script Group Properties dialog.</p> <p>You can move or copy a script from one user scripts folder to another; to:</p> <ul style="list-style-type: none"> <li>• Move a script, highlight it in the Scripts tab and drag it into the user scripts folder it now belongs to</li> </ul>	<p><a href="#">Script Group Properties</a>  <sup>[2794]</sup></p> <p><a href="#">Script Editor</a>  <sup>[2798]</sup></p>











Topic	Detail	See also
	<ul style="list-style-type: none"> <li>Copy a script, highlight it in the Scripts tab and press ( <b>Ctrl</b> ) while you drag it into the user scripts folder in which to duplicate it</li> </ul>	
<b>Context Menus</b>	<p>The script groups and their scripts also have context menus that provide some or all of the following options:</p> <ul style="list-style-type: none"> <li><b>Group Properties</b> - to display or edit script group properties in the Script Group Properties dialog</li> <li><b>Run Script</b> - to execute the selected script</li> <li><b>Edit Script</b> - to update the selected script</li> <li><b>Rename Script</b> - to change the name of the selected group or script</li> <li><b>New VBScript/JScript/JavaScript</b> - add a new script to the selected user group</li> <li><b>Import Workflow Script</b> - to display the Browser dialog through which you locate and select a workflow script source (.vbs) file to import into the Workflow script folder</li> <li><b>Delete Group/Script</b> - to delete the selected user group or script</li> </ul>	<a href="#">Script Group Properties</a> <small>[2794]</small>

The Scripts tab toolbar provides the following options:

### Script Toolbar

Icon	Action	See also
	<p>Create a new script group; this option displays a short menu of the types of script group you can create, namely:</p> <ul style="list-style-type: none"> <li>Normal Group ()</li> <li>Project Browser Group ()</li> <li>Diagram Group ()</li> <li>Workflow Group ()</li> <li>Search Group ()</li> <li>Model Search Group</li> </ul> <p>The new group is added to the end of the list in the Scripting window, with the ' <i>New group</i> ' text highlighted so that you can type in the group name.</p>	
	<p>Create a new script file in the selected script group; displays a short menu of the types of script you can create, namely:</p> <ul style="list-style-type: none"> <li>VBScript ()</li> </ul>	

Icon	Action	See also
	<ul style="list-style-type: none"> <li>JScript (  )</li> <li>JavaScript (  )</li> </ul> <p>The new script is added to the end of the list in the selected group, with the 'New script' text highlighted so that you can type in the script name.</p>	
	Refresh the script tree in the Scripting window; this icon also reloads any changes made to a workflow script.	
	<p>Compile and execute the selected script.</p> <p>The output from the script is written to the Script tab of the Output window, which you display using the <b>View Script Output</b> button (below).</p>	
	Stop an executing script; the icon is disabled if no script is executing.	
	<p>Delete a <i>script</i> from the model; you cannot use this icon to delete a script <i>group</i> (see the <i>Context Menu</i> item above), scripts in the Local Scripts group, or a script that is executing.</p> <p>The system prompts you to confirm the deletion only if the <b>Confirm Deletes</b> checkbox is selected in the Project Browser panel of the General page of the Options dialog; if this option is not selected, no prompt is displayed.</p> <p>Script deletion is permanent - scripts cannot be recovered.</p>	<a href="#">General Options</a> 
	Display the Output window with the results of the most recently executed script displayed in the Script tab.	

### Notes

- This facility is available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions
- If you add, delete or change a script, you might have to reload the model in order for the changes to take effect
- If you select to delete a script group that contains scripts, the system always prompts you to confirm the action regardless of any system settings for delete operations; be certain that you intend to delete the group and its scripts before confirming the deletion - deletion of script groups and scripts is permanent

#### 20.1.1.1 Script Group Properties

When you create a script you develop it within a script group, the properties of which determine how that script is to be made available to the user - through the Project Browser context menu to operate on objects of a specific type, or through a diagram context menu. You create a Script Group using the first icon on the Scripts tab toolbar.

**Access**   **Tools | Scripting > Scripts | Script Group name | Group Properties**

### Define the Script Group Properties

Field/Button	Action	See also
<b>Name</b>	Type in the name of the script group.	
<b>Group UID</b>	(Read only) The automatically assigned GUID for the group.	
<b>Source</b>	(Read only) The location of the template used to create the script.	
<b>Group Type</b>	<p>Click on the drop-down arrow and select the type of script contained in the group; this can be one of:</p> <ul style="list-style-type: none"> <li>• <b>Normal</b> - (📄) General model scripts</li> <li>• <b>Project Browser</b> - (📁) Scripts that are listed in and can be executed from the Project Browser <b>Scripts</b> context menu option</li> <li>• <b>Workflow</b> - (🔄) Scripts executed by Enterprise Architect's workflow engine; you can create only VB scripts of this type</li> <li>• <b>Search</b> - (🔍) Scripts that can be executed as <b>model searches</b>; these scripts are listed in the <b>Search</b> field of the Model Search window, in the last category in the list</li> <li>• <b>Diagram</b> - (📐) Scripts that can be executed from the <b>Scripts</b> submenu of the diagram context menu</li> <li>• <b>Find in Project</b> - (🔍) Scripts that can be executed from the <b>Scripts</b> submenu of a context menu within the Model Search view, on the results of a successfully-executed SQL search that includes CLASSGUID and CLASSTYPE, or a Query-built search</li> <li>• <b>Element</b> - Scripts that can be executed from the <b>Scripts</b> submenu of element context menus; accessible from the Project Browser, Diagram, Model Search, Element List, Package Browser and Gantt views</li> <li>• <b>Package</b> - Scripts that can be executed from the <b>Scripts</b> submenu of package context menus; accessible from the Project Browser</li> <li>• <b>Diagram</b> - Scripts that can be executed from the <b>Scripts</b> context menu option for diagrams; accessible from the Project Browser and diagrams</li> <li>• <b>Link</b> - Scripts that can be executed from the <b>Scripts</b> context menu option for connectors; accessible from diagrams</li> </ul>	<p><a href="#">Package Context Menu</a><sup>65↑</sup></p> <p><a href="#">Workflow Scripts</a><sup>36↑</sup></p> <p><a href="#">Model Search</a><sup>70↑</sup></p> <p><a href="#">Diagram Context Menu</a><sup>78↑</sup></p>
<b>Notes</b>	Type in any comments you need regarding this script group.	

Field/Button	Action	See also

#### Learn more

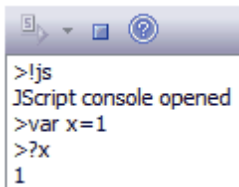


- [Scripts Tab](#)<sup>[2792]</sup>

## 20.1.2 Console Tab

**Access** **Tools | Scripting > Console**

#### Topics

Topic	Detail	See also
<b>Abstract</b>	<p>The script console is a tab of the Scripting window; it is a command line interpreter through which you can quickly enable a script engine and enter commands to act on the script.</p> <p>You type the commands in the field at the bottom of the tab; when you press the ( <b>Enter</b> ) key, the script console executes the commands and displays any output immediately</p> <p>You can input two types of command:</p> <ul style="list-style-type: none"> <li>• Console commands</li> <li>• Script commands</li> </ul>	
<b>Console Commands</b>	<p>Console commands are preceded by the ! character and instruct the console to perform an action.</p> <p>The available console commands are listed below; to list these commands on the <b>Console</b> tab itself, type ? in the console field (without the preceding ! character) and press the ( <b>Enter</b> ) key.</p> <ul style="list-style-type: none"> <li>• <b>c(lear)</b> - clears the console display</li> <li>• <b>sa(ve)</b> - saves the console display to a file</li> <li>• <b>h(elp)</b> - prints a list of commands, as for ?</li> <li>• <b>VB</b> - opens a VBScript console</li> <li>• <b>JA</b> - opens a JavaScript console</li> <li>• <b>JS</b> - opens a JScript console</li> <li>• <b>st(op)</b> - closes any script running console</li> <li>• <b>i(nclude)</b> name - executes the named script item; name is of the format <i>GroupName.ScriptName</i> (spaces are allowed in names)</li> <li>• <b>?</b> - (without the !) lists commands</li> <li>• <b>?name</b> - Outputs the value of a variable name (only if a script console is opened).</li> </ul>	

Topic	Detail	See also
<b>Script Commands</b>	<p>A script command is script code that depends on the script engine.</p> <p>Script commands can be executed only once a script console has been created.</p> <p><i>Examples:</i></p> <p>The following lines, entered into the console, create a <i>VBScript</i> console and then execute the script <i>MyScript</i> in the user group <i>MyGroup</i>:</p> <pre>&gt;! VB &gt;! i MyGr oup. MyScr i pt</pre> <p>The following lines, entered into the console, create a <i>JScript</i> console and then create a variable called <i>x</i> with the value <b>1</b>:</p> <pre>&gt;! JS &gt;var x = 1</pre> <p>The following image shows the result of entering the above <i>JScript</i> example; remember that you can use <b>?&lt;variable name&gt;</b> to get the current value of any item you have created during the console session.</p> 	
<b>Console Tab Toolbar</b>	<p>The <b>Console</b> tab has two operations available through the toolbar:</p> <ul style="list-style-type: none"> <li>• <b>Open Console</b> (  ) - click on the down-arrow and select to open a VBScript console, JScript console or JavaScript console</li> <li>• <b>Stop Script</b> (  ) - click to stop an executing script and close the current console</li> </ul>	

### Notes

- This facility is available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions
- You can save the output of the console to an external .txt file; right-click on the console window, select the **Save As** context menu option, browse for an appropriate file location and specify the file name

### 20.1.3 Script Editor

Using the **Script Editor** you can perform a number of operations on an open script file, such as:

- Save changes to the current script
- Save the current script under a different name
- Run the script
- Debug the script
- Stop the executing script
- View the script output in the Scripts tab of the System Output window

The editor is based on, and provides the facilities of, the common Code Editor in the application work area.

**Access** **Tools | Scripting > Scripts: double-click on script name**

#### Facilities

Facility	Detail	See also
<b>Scripting Objects</b>	<p>Enterprise Architect adds to the available functionality and features of the editor script language by providing inbuilt objects; these are either <i>Type Libraries</i>, providing Intelli-sense for editing purposes, or <i>Runtime objects</i>, providing access to objects of the types described in the Type Libraries.</p> <p>The available Intelli-sense scripting objects are:</p> <ul style="list-style-type: none"> <li>• EA</li> <li>• MathLib</li> <li>• System</li> </ul> <p>The runtime scripting objects are:</p> <ul style="list-style-type: none"> <li>• Repository ( Type: IDualRepository, an instance of EA. Repository, the Enterprise Architect <b>Automation Interface</b> )</li> <li>• Maths ( Type: IMath, an instance of MathLib; this exposes functions from the Cephes mathematical library for use in scripts - see below )</li> <li>• Session ( Type: ISession, an instance of System )</li> </ul>	<p><a href="#">Automation Interface</a> [2804]</p> <p><a href="#">Session Object</a> [2806]</p>
<b>Script Editing Intelli-sense (Required Syntax)</b>	<p>Intelli-sense is available not only in the Script Editor, but also in the Script Console; Intelli-sense at its most basic is presented for the inbuilt functionality of the script engine.</p> <p>For Intelli-sense on the additional Enterprise Architect scripting objects listed above, you must declare variables according to syntax that specifies a type; it is not necessary to use this syntax to execute a script properly, it is only present so that the correct Intelli-sense can be displayed for an item.</p> <p>The syntax can be seen in, for example:</p> <pre>Dim e as EA.Element</pre>	

Facility	Detail	See also
	<p>Then when you type, in this case, <b>e.</b>, the editor displays a list of member functions and properties of <b>e</b>'s type.</p> <p>You select one of these to complete the line of script; you might, therefore, type:</p> <pre>VBTrace( e.</pre> <p>As you type the period, the editor presents the appropriate list and you might double-click on, for example, <b>Abstract</b>; this is inserted in the line, and you continue to type or select the rest of the statement, in this case adding the end space and parenthesis:</p> <pre>VBTrace( e. Abstract )</pre>	
<b>Keystrokes</b>	<p>In the Script Editor or Console, Intelli-sense is presented on the following keystrokes.</p> <ul style="list-style-type: none"> <li>• Press ( <b>.</b> ) (period) after an item to list any members for that item's type</li> <li>• Press ( <b>Ctrl</b> ) + ( <b>Spacebar</b> ) on a word to list any Intelli-sense items with a name starting with the string at the point the keys were pressed</li> <li>• Press ( <b>Ctrl</b> ) + ( <b>Spacebar</b> ) when not on a word to display any available top level Intelli-sense items - these are the Intelli-sense objects described above plus any built-in methods and properties of the current scripting language.</li> </ul>	
<b>Include script libraries</b>	<p>An <b>Include statement</b> ( <b>!INC</b> ) allows a script to reference constants, functions and variables defined by another script accessible within the Scripting Window. Include statements are typically used at the beginning of a script.</p> <p>To include a script library, use the following syntax:</p> <pre>!INC [ Script Group Name] . [ Script Name]</pre> <p>For example:</p> <pre>!INC Local Scripts.EAConstants-VBScript</pre>	
<b>Using Inbuilt Math Functions</b>	<p>Various mathematical functions are available within the Script Editor, through the use of the inbuilt <b>Maths</b> object.</p> <p>You can access the Maths object within the Script Editor by typing <b>Maths</b> followed by a period. The Intelli-sense feature displays a list of the available mathematical functions provided by the Cephes Mathematical Library. For example:</p> <pre>Session.Output "The square root of 9 is " &amp; Maths.sqrt(9)</pre> <pre>Session.Output "2^10 = " &amp; Maths.pow(2, 10)</pre>	

Facility	Detail	See also
<b>System Script Library</b>	<p>When Enterprise Architect is installed on your system, it includes a default script library that provides a number of helpful scripting functions, varying from simple string functions to functions for defining your own CSV or XML import and export.</p> <p>To use the script library you must enable it in the MDG Technologies dialog (<b>Settings   MDG Technologies</b>).</p> <p>Scroll through the list of technologies, and select the <b>Enabled</b> checkbox against <b>EAScriptLib</b>.</p>	

### Notes

- The Script Editor is available in the Corporate, Business and Software Engineering, Systems Engineering and Ultimate editions
- Enterprise Architect scripting supports declaring variables to match the Enterprise Architect types; this enables the editor to present **Intelli-sense**, but is not necessary for executing the script

### Learn more

- [Editing Source Code](#) <sup>[2146]</sup>
- [Scripts Tab](#) <sup>[2792]</sup>
- [Script Debugging](#) <sup>[2801]</sup>
- [Cephes Mathematical Library](#) (3rd-Party, Online Resource)

## 20.1.4 Session Object

The **Session** runtime object provides a common input/feedback mechanism across all script languages, giving access to objects of the types described in the **System Type** library. It is available through both the Scripts tab and the script console to any script run within Enterprise Architect.

### Properties

Properties	Detail	See also
<b>Attributes</b>	<ul style="list-style-type: none"> <li>• <b>UserName</b> - Returns the current windows username (read only)</li> <li>• <b>Version</b> - Returns the version of this object (read only)</li> </ul>	
<b>Methods</b>	<ul style="list-style-type: none"> <li>• <b>Input(string Prompt)</b> - displays a dialog box prompting the user to input a value; returns the string value that was entered by the user</li> <li>• <b>Output(string Output)</b> - writes text to the current default output location; during: <ul style="list-style-type: none"> <li>• Normal script execution, output is written to the Script tab of the Output window</li> </ul> </li> </ul>	



Properties	Detail	See also
	<ul style="list-style-type: none"> <li>Script Debugging, output is written to the Debug window</li> <li>Use of the Script Console, output is written to the Console</li> <li><b>Prompt(string Prompt, long PromptType)</b> - displays a modal dialog to the user containing the specified prompt text and button types; returns the <i>PromptResult</i> value corresponding to the button that the user clicked</li> </ul>	
<b>PromptType values</b>	<ul style="list-style-type: none"> <li>promptOK = 1</li> <li>promptYESNO = 2</li> <li>promptYESNOCANCEL = 3</li> <li>promptOKCANCEL = 4</li> </ul>	
<b>PromptResult values</b>	<ul style="list-style-type: none"> <li>resultOK = 1</li> <li>resultCancel = 2</li> <li>resultYes = 3</li> <li>resultNo = 4</li> </ul>	
<b>Session.Prompt Example</b>	<p>(VBScript)</p> <pre>If ( Session.Prompt ( " Continue?", prompt YESNO) = resultYes ) Then</pre>	

Learn more

- [Scripting](#)<sup>[2791]</sup>
- [Script Editor](#)<sup>[2798]</sup>


### 20.1.5 Script Debugging

Script debugging aids in the development and maintenance of model scripts, and monitoring their activity at the time of execution. While debugging a script, you can:

- Control execution flow using the **Debug**, **Step Over**, **Step Into**, **Step Out** and **Stop Script** buttons on the Script Editor toolbar
- Set Breakpoints, Recording Markers and Tracepoint Markers
- Use the Debug window to view output generated by the script
- Use the Locals window to inspect values of variables, including objects from the Automation Interface
- Use the Record & Analyze window to record a Sequence diagram of the script execution

**Access**    **Script Editor Toolbar: Debug button**

**Begin debugging a model script**

Step	Action	See Also
1	Open a model script in the Script Editor.	<a href="#">Scripts Tab</a> <sup>[2792]</sup> <a href="#">Script Editor</a> <sup>[2798]</sup>
2	Set any Breakpoints on the appropriate line(s) of code.	<a href="#">Setting Code Breakpoints</a> <sup>[2226]</sup> <a href="#">Marker Types</a> <sup>[2538]</sup>
3	Click on the  toolbar icon ( <b>Debug</b> ).	

**Notes**

- Script debugging is supported for VBScript, JScript and Javascript
- VBScript and JScript require the Microsoft Process Debug Manager to be installed on the local machine; this is available through various Microsoft products including the free 'Microsoft Script Debugger'
- Breakpoints are not saved for scripts and will not persist when the script is next opened
- While debugging, script output is redirected to the Debug window

**Learn more**

- [Debugging](#) <sup>[2222]</sup>
- [Run the Debugger](#) <sup>[2231]</sup>
- [Breakpoint and Marker Management](#) <sup>[2224]</sup>
- [The Recording History](#) <sup>[2533]</sup>
- [View the Local Variables](#) <sup>[2234]</sup>
- [View Debug Variables in Code Editors](#) <sup>[2236]</sup>
- [Rapid Technology Development with Profile Helpers & Model Scripts](#) (Online Resource)

**20.1.6 Simulation Scripts**

You can perform **dynamic simulation** of your models using **scripts** written in **JavaScript**. All simulation objects and variables are created in the Java run time environment.

The current simulation objects have a root element **sim** and the object or Class type **SimObject**. The **sim** element is the root of the tree structure of generated simulation objects and their properties shown in the Locals window, although - unlike user interfaces or Java variables - **sim** itself as the root is not shown. Each **SimObject** has properties, some of which are themselves **SimObjects**. For example: **sim.a.b** can be a simulation object, having a single value - an integer or a string.

Java variables created by Java data types, such as **var**, **Object** and **Array** are not shown in the Locals window, though they exist in the Java runtime environment. You can check for them using code; for example:

```
var a = 5; sim.a = a;
```

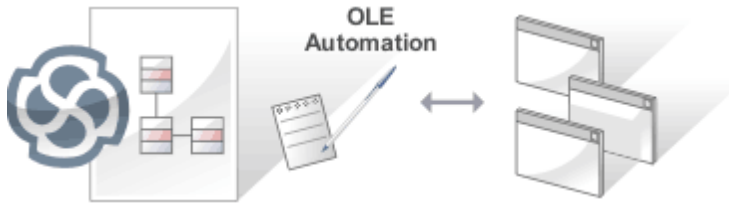
In this way, `sim.a` will be shown in the Locals window.

You can also create an Array or an Object using this code:

```
sim.a = new Array();  
sim.a = [ 1000, 1100, 1200, 1300];  
sim.simArray = nativeArray; //nativeArray is a Java Array  
sim.o = new Object();  
sim.o = { property1: 5, property2: "myString" };
```

The resulting objects are **simulation** objects but not **Java** objects; that is, this is a **transformation** process, creating Java objects and transforming them into simulation objects.

## 20.2 Enterprise Architect Object Model






### Topics

Topic	Detail	See also
<b>Introduction</b>	<p>Automation provides a way for other applications to access the information in an Enterprise Architect model using Windows OLE Automation (ActiveX). Typically this involves scripting clients such as MS Word or Visual Basic, or using scripts created within Enterprise Architect using the Scripting window.</p> <p>The Automation Interface provides a way of accessing the internals of Enterprise Architect models. Examples of things you can do using the Automation Interface include:</p> <ul style="list-style-type: none"> <li>• Perform repetitive tasks, such as update the version number for all elements in a model</li> <li>• Generate code from a State Machine diagram</li> <li>• Produce custom reports</li> <li>• Perform ad hoc queries</li> </ul>	<a href="#">Scripting</a> <sup>[2791]</sup>
<b>Connecting to the Automation Interface</b>	All development environments capable of generating <i>ActiveX</i> Com clients should be able to connect to the Enterprise Architect Automation Interface. This guide provides detailed instructions on connecting to the interface using Microsoft Visual Basic 6.0, Borland Delphi 7.0, Microsoft C# and Java. There are also more detailed steps on how to set-up Visual Basic; the principles are applicable to other languages.	<a href="#">Connecting to the Interface</a> <sup>[2805]</sup> <a href="#">Set References in Visual Basic</a> <sup>[2808]</sup>
<b>Examples and Tips</b>	Instruction on how to use the Automation Interface is provided by means of sample code. See pointers to the samples and other available resources. Also, consult the extensive Reference Section.	<a href="#">Pointers to the Samples</a> <sup>[2809]</sup> <a href="#">Available Resources</a> <sup>[2811]</sup> <a href="#">Reference</a> <sup>[2812]</sup>
<b>Calling Executables from Enterprise Architect</b>	Enterprise Architect can be set up to call an external application. You can pass parameters on the current position selected in the Project Browser to the application being called. For instructions, go to the Call from Enterprise Architect topic. A more sophisticated method is to create Add-Ins, which are discussed in a separate topic.	<a href="#">Call from Enterprise Architect</a> <sup>[2810]</sup> <a href="#">Add-Ins</a> <sup>[3010]</sup>

## 20.2.1 Using the Automation Interface

This section provides instructions on how to connect to and use the Automation Interface, including:

Topic	See also
Connecting to the Interface	<a href="#">Connecting to the Interface</a> 
Setting References In Visual Basic	<a href="#">Setting References In Visual Basic</a> 
Examples and Tips	<a href="#">Examples and Tips</a> 

### 20.2.1.1 Connect to the Interface

All development environments capable of generating ActiveX Com clients can connect to the Enterprise Architect Automation Interface.

By way of example, the following sections describe how to connect using several such tools. The procedure might vary slightly with different versions of these products.

#### Microsoft Visual Basic 6.0

Step	Action	See also
1	Create a new project.	
2	Select the <b>Project   References</b> menu option.	
3	Select <i>Enterprise Architect Object Model 2.0</i> from the list. If this does not appear, go to the command line and re-register Enterprise Architect using:  <i>EA.exe /unregister</i>  then  <i>EA.exe /register</i>	
4	See the general library documentation on the use of Classes. The following example creates and opens a repository object:  <pre>Public Sub ShowRepository()     Dim MyRep As New EA.Repository     MyRep.OpenFile "c:\eat est.eap" End Sub</pre>	

**Borland Delphi 7.0**

Step	Action	See also
1	Create a new project.	
2	Select the <b>Project   Import Type Library</b> menu option.	
3	<p>Select <b>Enterprise Architect Object Model 2.0</b> from the list.</p> <p>If this does not appear, go to the command line and re-register Enterprise Architect using:</p> <pre>EA.exe /unregister</pre> <p>then</p> <pre>EA.exe /register</pre>	
4	Click on the <b>Create Unit</b> button.	
5	Include <i>EA_TLB</i> in Project1's <i>Uses</i> clause.	
6	<p>See the general library documentation on the use of Classes. The following example creates and opens a repository object:</p> <pre> procedure TForm1.Button1Click(Sender: TObject); var   r: TRepository;   b: boolean; begin   r := TRepository.Create(nil);   b := r.OpenFile('c:\eat\eat.eap'); end; </pre>	

**Microsoft C#**

Step	Action	See also
1	Select the Visual Studio <b>Project   Add Reference</b> menu option.	
2	Click on the Browse tab.	
3	<p>Navigate to the folder in which you installed Enterprise Architect; usually:</p> <pre>Program Files\ Sparx Systems\ EA</pre>	

Step	Action	See also
	<p>Select</p> <p><code>Interop.EA.dll</code></p>	
4	<p>See the general library documentation on the use of Classes. The following example creates and opens a repository object:</p> <pre>private void button1_Click(object sender, System. EventArgs e) {     EA.Repository r = new EA.Repository();     r.OpenFile("c:\\eatest.eap"); }</pre>	

### Java

Step	Action	See also
1	<p>Copy the file:</p> <p><code>SSJavaCOM.dll</code></p> <p>from the Java API subdirectory of your installed directory, usually:</p> <p><code>Program Files/Sparx Systems/EA</code></p> <p>into any location within the Windows PATH</p> <p><code>windows\system32\directory.</code></p>	
2	<p>Copy the file</p> <p><code>eaapi.jar</code></p> <p>from the Java API subdirectory of your installed directory, usually:</p> <p><code>Program Files/Sparx Systems/EA</code></p> <p>to a location in the Java CLASSPATH or where the Java class loader can find it at run time.</p>	
3	<p>All of the Classes described in the documentation are in the package <i>org.sparx</i>. See the <b>general library documentation</b> for their use. The following example creates and opens a repository object:</p> <pre>public void OpenRepository() {     org.sparx.Repository r = new org.sparx.Repository();     r.OpenFile("c:\\eatest.eap"); }</pre>	<a href="#">Reference</a> <sup>[2812]</sup>

Step	Action	See also

### 20.2.1.1.1 Set References In Visual Basic

This topic describes how to use the Enterprise Architect ActiveX interface with Visual Basic (VB). Use is ensured for Visual Basic version 6, but might vary slightly with versions other than version 6.

It is assumed that you have accessed VB through a Microsoft Application such as VB 6.0, MS Word or MS Access. If the code is not called from within Word, the *Word VB* reference must also be set.

#### How to

On creating a new VB project, to set a reference to an Enterprise Architect Type Library and a Word Type Library follow the steps below:

Step	Action	See also
1	Select the <b>Tools   References</b> menu option.	
2	Select the <b>Enterprise Architect Object Model 2.10</b> checkbox from the list.	
3	Do the same for VB or VB Word: select the checkbox for the <b>Microsoft Word 10.0 Object Library</b> .	
4	Click on the <b>OK</b> button.	

#### Notes

- If Enterprise Architect Object Model 2.10 does not appear in the list, go to the command line and manually re-enter Enterprise Architect using the following:
  - To unregister Enterprise Architect: `ea.exe /unregister`
  - To register Enterprise Architect: `ea.exe /register`
- **Visual Basic 5/6** users should also note that the version number of the Enterprise Architect interface is stored in the VBP project file in a form similar to the following:

```
Reference=* \ G{ 64FB2BF4- 9EFA- 11D2- 8307- C45586000000} #2. 2#0#. . \ . . \ . .
\ Program Files\ Sparx Systems\ EA\ EA. TLB#Enterprise Architect Object
Model 2. 02
```

If you experience problems moving from one version of Enterprise Architect to another, open the VBP file in a text editor and remove this line, then open the project in Visual Basic and use Project-References to create a new reference to the Enterprise Architect Object model

Reference to objects in Enterprise Architect and Word should now be available in the **Object Browser**,




which can be accessed from the main menu by selecting **View | Object Browser**, or by pressing ( **F2** )

The drop-down list on the top-left of the window should now include Enterprise Architect and Word; if MS-Project is installed, also set this up

### 20.2.1.2 Examples and Tips

#### Topics

Topic	Detail	See also
<b>Usage</b>	<p>Instructions for using the interface are provided through sample code. There are several sets of examples:</p> <ul style="list-style-type: none"> <li>• VB 6 and C# examples are available in the Code Samples folder under your Enterprise Architect installation (default: C:\Program Files\Sparx Systems\EA\Code Samples)</li> <li>• Enterprise Architect can be set up to <b>call an external application</b></li> <li>• Several VB.NET code snippets are provided in the <b>reference section</b></li> <li>• A comprehensive example of using Visual Basic to create MS Word documentation is available from the internet at <a href="http://www.sparxsystems.com/resources/developers/autint_vb.html">www.sparxsystems.com/resources/developers/autint_vb.html</a></li> <li>• Additional samples are available from the Sparx Systems website; see the <i>Available Resources</i> topic</li> </ul>	<p><a href="#">Call an External Application</a> <sup>[2810]</sup></p> <p><a href="#">Code Samples</a> <sup>[2994]</sup></p> <p><a href="http://www.sparxsystems.com/resources/developers/autint_vb.html">www.sparxsystems.com/resources/developers/autint_vb.html</a></p> <p><a href="#">Available Resources</a> <sup>[2811]</sup></p>
<b>Tips and Tricks</b>	<p>Additionally, you should note the following tips and tricks:</p> <ul style="list-style-type: none"> <li>• An instance of the Enterprise Architect (<i>EA.exe</i>) process is executed when you initialize a new repository object - this process must remain running in order to perform automation tasks; if the main window is visible, you can safely minimize it, but it must remain running</li> <li>• The Enterprise Architect ActiveX Interface is a functional interface rather than a data interface; when you load data through the interface there is a noticeable delay as Enterprise Architect user interface elements (such as Windows and menus) are loaded and the specified database connection is established</li> <li>• Collections use a zero-based index; for example, <code>Repository.Models(0)</code> represents the first model in the repository</li> <li>• During the development of your client software your program might terminate unexpectedly and leave <i>EA.exe</i> running in such a state that it is unable to support further interface calls; if your program terminates abnormally, ensure that Enterprise Architect is not left running in the background (see</li> </ul>	<p><a href="#">Repository Class</a> <sup>[2850]</sup></p> <p><a href="#">App Class</a> <sup>[2814]</sup></p>

Topic	Detail	See also
	<p>the Windows Task Manager / Process tab)</p> <ul style="list-style-type: none"> <li>A handle to a currently running instance of Enterprise Architect can be obtained through the use of a <i>GetObject()</i> call (see the reference page for the <i>App</i> object); accessing your Enterprise Architect model via the <i>App</i> object enables querying the current User Interface status, such as using <i>GetCurrentItem()</i> on the <b>Repository</b> object to detect the current selection by the user, allowing for rapid prototyping and testing</li> </ul>	
<b>Enterprise Architect Not Closing</b>	<p>If your automation controller was written using the .NET framework, Enterprise Architect does not close even after you release all your references to it.</p> <p>To force the release of the COM pointers, call the memory management functions as shown below:</p> <pre>GC. Collect ( ) ; GC. Wait For PendingFinalizers ( ) ;</pre> <p>There are additional concerns when controlling a running instance of Enterprise Architect that loads Add-Ins - see the <i>Tricks and Traps</i> topic for details.</p>	<a href="#">Tricks and Traps</a> 

#### 20.2.1.2.1 Call from Enterprise Architect

Enterprise Architect can be set up to call an external application. You can pass parameters on the current position selected in the Project Browser to the application being called.

**Access**   **Tools | Customize > Tools**

#### Topics

Topic	Detail	See also
<b>Uses</b>	<p>With this you can:</p> <ul style="list-style-type: none"> <li>Add a command line for an application</li> <li>Define parameters to pass to this application</li> </ul> <p>The parameters required for running the AutInt executable are:</p> <ul style="list-style-type: none"> <li>The Enterprise Architect file parameter \$f and</li> <li>The current PackageID \$p</li> </ul> <p>Hence the arguments should simply contain: \$f , \$p</p>	

The available parameters for passing information to external applications are:

Parameter	Description	Notes
\$d	Diagram ID	ID for accessing associated diagram.
\$D	Diagram GUID	GUID for accessing the associated diagram.
\$e	Comma separated list of element IDs	All elements selected in the current diagram.
\$E	Comma separated list of element GUIDs	All elements selected in the current diagram.
\$f	Project Name	For example: C: \ p r o j e c t s \ EAexamp l e . eap .
\$F	Calling Application (Enterprise Architect)	'Enterprise Architect'.
\$p	Current Package ID	For example: 144.
\$P	Package GUID	GUID for accessing this package.

Once this has been set up, the application can be called from the main menu in Enterprise Architect using the **Extensions | <YourApplication>** menu option.

#### 20.2.1.2.2 Available Resources

Other available resources include:

Resource	Download Link
VB 6 Add-In for generating MS Word documentation.	<a href="http://www.sparxsystems.com/resources/developers/autint_vb.html">www.sparxsystems.com/resources/developers/autint_vb.html</a>
VB 6 Add-In to display a custom ActiveX graph control within the Enterprise Architect window as a new view.	<a href="http://www.sparxsystems.com/resources/developers/autint_vb_custom_view.html">www.sparxsystems.com/resources/developers/autint_vb_custom_view.html</a>
A basic Add-In framework written in C#. Useful as a starting point for authoring your own custom Enterprise Architect Add-In.	<a href="http://www.sparxsystems.com/bin/CS_AddinFramework.zip">www.sparxsystems.com/bin/CS_AddinFramework.zip</a>
An extension on the <i>CS_AddinFramework</i> example showing how to export Tagged	<a href="http://www.sparxsystems.com/bin/CS_AddinTaggedCSV.zip">www.sparxsystems.com/bin/CS_AddinTaggedCSV.zip</a>

Resource	Download Link
Values to a .csv file.	
A basic Add-In skeleton written in Delphi.	<a href="http://www.sparxsystems.com/bin/DelphiDemo.zip">www.sparxsystems.com/bin/DelphiDemo.zip</a>
A simple example Add-In written in C#.	<a href="http://www.sparxsystems.com/bin/CS_Sample.zip">www.sparxsystems.com/bin/CS_Sample.zip</a>



#### Learn more

- For further information, see [www.sparxsystems.com/resources/developers/autint.html](http://www.sparxsystems.com/resources/developers/autint.html)

## 20.2.2 Reference






This section provides detailed information on all the objects available in the object model provided by the Automation Interface, covering:

Topic	Link
Interface Overview Package	<a href="#">Interface Overview Package</a> <sup>[2813]</sup>
App Object	<a href="#">App Object</a> <sup>[2814]</sup>
Enumerations	<a href="#">Enumerations</a> <sup>[2815]</sup>
Repository Package	<a href="#">Repository Package</a> <sup>[2826]</sup>
Element Package	<a href="#">Element Package</a> <sup>[2877]</sup>
Element Features Package	<a href="#">Element Features Package</a> <sup>[2910]</sup>
Connector Package	<a href="#">Connector Package</a> <sup>[2933]</sup>
Diagram Package	<a href="#">Diagram Package</a> <sup>[2948]</sup>
Project Interface Package	<a href="#">Project Interface Package</a> <sup>[2962]</sup>
Document Generator Interface Package	<a href="#">Document Generator Interface Package</a> <sup>[2985]</sup>

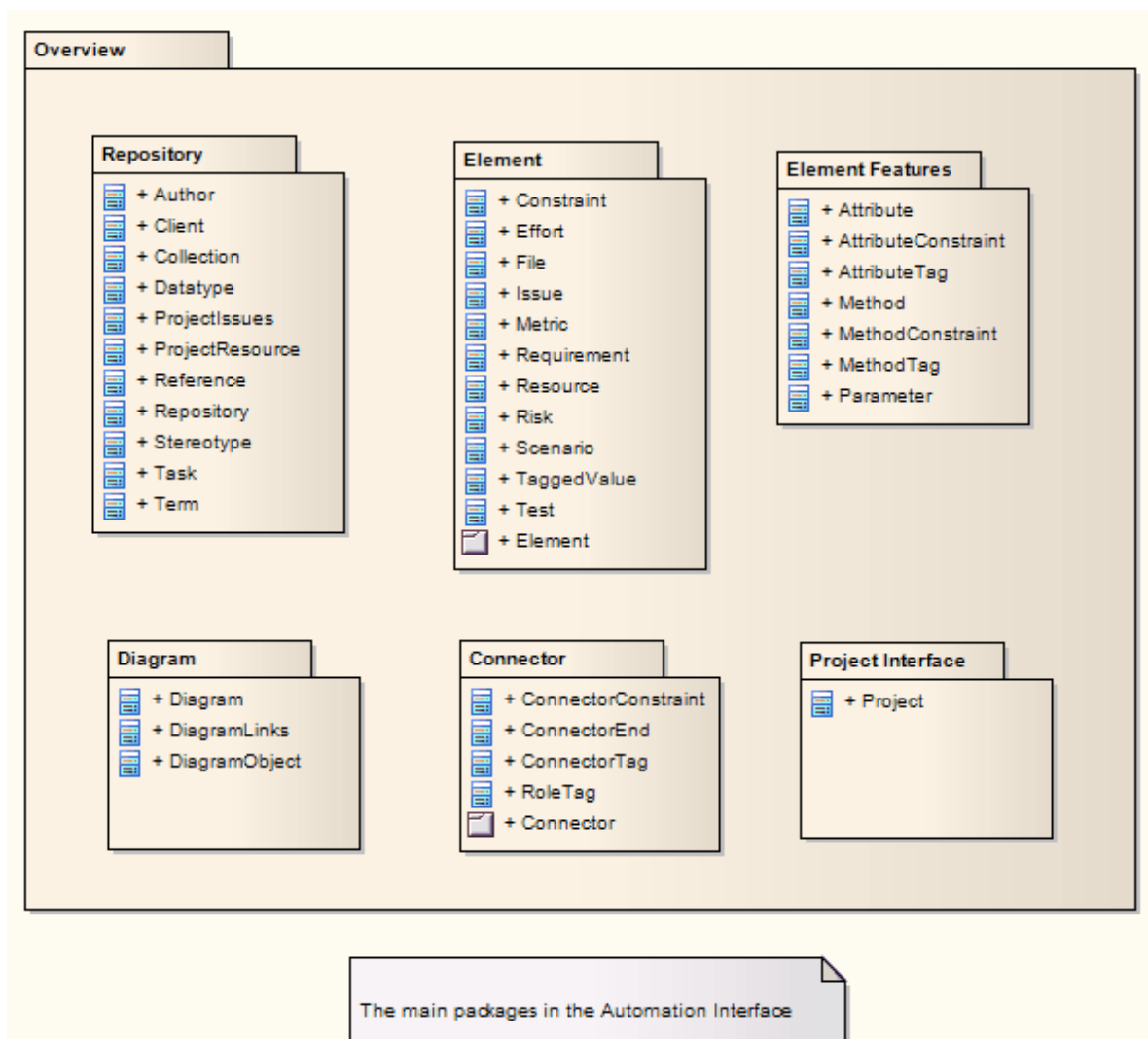
Topic	Link
Mail Interface Package	<a href="#">Mail Interface Package</a> 
Code Samples	<a href="#">Code Samples</a> 

### 20.2.2.1 Interface Overview Package

The Interface Overview Package provides an overview of the main components of the Automation Interface. These include the:

Topic	Detail	Link
<b>Repository Package</b>	Represents the model as a whole and provides entry to model packages and collections.	<a href="#">Repository Package</a> 
<b>Element Package</b>	Identifies the basic structural units (such as Class, Use Case and Object).	<a href="#">Element Package</a> 
<b>Element Features Package</b>	Identifies the attributes and operations defined on an element.	<a href="#">Element Features Package</a> 
<b>Diagram Package</b>	Describes the visible drawings contained in the model.	<a href="#">Diagram Package</a> 
<b>Connector Package</b>	Defines the relationships between elements.	<a href="#">Connector Package</a> 

The following diagram illustrates the main interface packages and their associated contents. Each element in this document is creatable by Automation and can be accessed through the various collections that exist or, in some cases, directly.



The *Repository* Class is the starting point for all use of the automation interface. It contains the high level system objects and entry point into the model itself using the *Models* collection and the other system level collections.

### 20.2.2.2 App Object

The *App* object represents a running instance of Enterprise Architect. Its object provides access to the Automation Interface.



Attribute	Type	Notes
Project	Project	Read only Provides a handle to the Project Interface.
Repository	Repository	Read only Provides a handle to the Repository object.
Visible	Boolean	Read/Write

Attribute	Type	Notes
		Whether or not the application is visible.

Topic	Detail	See also
<b>GetObject() Support</b>	<p>The <i>App</i> object is creatable and a handle can be obtained by creating one. In addition, clients can use the equivalent of Visual Basic's <i>GetObject()</i> to obtain a reference to a currently running instance of Enterprise Architect.</p> <p>Use this method to more quickly test changes to Add-Ins and external clients, as the Enterprise Architect application and data files do not have to be constantly re-loaded.</p> <p>For example:</p> <pre>Dim App as EA.App Set App = GetObject(, "EA.App") MsgBox App.Repository.Models.Count</pre> <p>Another example, which uses the App object without saving it to a variable:</p> <pre>Dim Rep as EA.Repository Set Rep = GetObject(, "EA.App").Repository MsgBox Rep.ConnectionString</pre>	

### 20.2.2.3 Enumerations

The following enumerations are defined by the Automation Interface:

Enumeration	Link
Constant Layout Styles	<a href="#">ConstLayoutStyles</a>  <sup>[2816]</sup>
Create Baseline Flag	<a href="#">CreateBaselineFlag</a>  <sup>[2817]</sup>
Create Model Type	<a href="#">CreateModelType</a>  <sup>[2818]</sup>
Document Break	<a href="#">DocumentBreak</a>  <sup>[2818]</sup>
Document Page Orientation	<a href="#">DocumentPageOrientation</a>  <sup>[2818]</sup>
Document Type	<a href="#">DocumentType</a>  <sup>[2819]</sup>
EA Edition Types	<a href="#">EAEditionTypes</a>  <sup>[2819]</sup>

Enumeration	Link
Enumeration Relation Set Type	<a href="#">EnumRelationSetType</a>  [2820]
Export Package XML Flag	<a href="#">ExportPackageXMLFlag</a>  [2820]
Mail Interface Message Flag	<a href="#">MessageFlag</a>  [2821]
MDG Menus	<a href="#">MDGMenus</a>  [2821]
Object Type	<a href="#">ObjectType</a>  [2822]
PropType	<a href="#">PropType</a>  [2823]
Reload Type	<a href="#">ReloadType</a>  [2823]
Scenario Diagram Type	<a href="#">ScenarioDiagramType</a>  [2824]
Scenario Step Type	<a href="#">ScenarioStepType</a>  [2825]
Scenario Test Type	<a href="#">ScenarioTestType</a>  [2825]
XML Type	<a href="#">XMLType</a>  [2825]

#### 20.2.2.3.1 ConstLayoutStyles

The *enum* values defined here are used exclusively for the *Lay Out a Diagram* method. They enable you to define the layout options as depicted in the **Diagram | Layout Diagram** menu option.

Value	Meaning
<i>IsCrossReduceAggressive</i>	Perform aggressive Cross-reduction in the layout process (time consuming).
<i>IsCycleRemoveDFS</i>	Use the <i>Depth First Cycle Removal</i> algorithm.
<i>IsCycleRemoveGreedy</i>	Use the <i>Greedy Cycle Removal</i> algorithm.



Value	Meaning
<i>IsDiagramDefault</i>	Use existing layout options specified for this diagram.
<i>IsInitializeDFSIn</i>	Initialize the layout using the <i>Depth First Search Inward</i> algorithm.
<i>IsInitializeNaive</i>	Initialize the layout using the <i>Naïve Initialize Indices</i> algorithm.
<i>IsInitializeDFSOut</i>	Initialize the layout using the <i>Depth First Search Outward</i> algorithm.
<i>IsLayeringLongestPathSink</i>	Layer the diagram using the <i>Longest Path Sink</i> algorithm.
<i>IsLayeringLongestPathSource</i>	Layer the diagram using the <i>Longest Path Source</i> algorithm.
<i>IsLayeringOptimalLinkLength</i>	Layer the diagram using the <i>Optimal Link Length</i> algorithm.
<i>IsLayoutDirectionDown</i>	Direct connectors to point down.
<i>IsLayoutDirectionLeft</i>	Direct connectors to point left.
<i>IsLayoutDirectionRight</i>	Direct connectors to point right.
<i>IsLayoutDirectionUp</i>	Direct connectors to point up.
<i>IsProgramDefault</i>	Use factory default layout options as specified by Enterprise Architect.

#### Learn more

- [Lay Out a Diagram](#)<sup>[89]</sup>

#### 20.2.2.3.2 CreateBaselineFlag

The *CreateBaselineFlag* enumeration is used in Baseline Management, when creating a Baseline.

Value	Meaning
<i>cbSaveToStub</i>	Baseline this package with only immediate children (child packages are included as stubs only).

### 20.2.2.3.3 CreateModelType

The *CreateModelType* enumeration is used in the **CreateModel** method on the Repository Class.

Value	Meaning
<i>cmEAPFromBase</i>	Create a copy of the EABase model file to the specified file path.
<i>cmEAPFromSQLRepository</i>	Create a .eap file shortcut to an SQL-based repository; requires user interaction to provide sql connection details.

#### Learn more

- [CreateModel Method](#) 

### 20.2.2.3.4 DocumentBreak

The *DocumentBreak* enumeration is used in the **InsertBreak** method on the *DocumentGenerator* Class.

Value	Meaning
<i>breakPage</i>	Insert a page break in the document.
<i>breakSection</i>	Insert a section break in the document.

#### Learn more

- [InsertBreak Method](#) 

### 20.2.2.3.5 DocumentPageOrientation

The *DocumentPageOrientation* enumeration is used in the **SetPageOrientation** method on the *DocumentGenerator* Class.

Value	Meaning
<i>pagePortrait</i>	Sets the current page orientation to <b>Portrait</b> .
<i>pageLandscape</i>	Sets the current page orientation to <b>Landscape</b> .

#### Learn more

- [SetPageOrientation Method](#) 

### 20.2.2.3.6 DocumentType

The *DocumentType* enumeration is used in the **SaveDocument** method on the *DocumentGenerator* Class.

Value	Meaning
<i>dtRTF</i>	Save the document file to disk as an RTF document.
<i>dtHTML</i>	Save the document file to disk as a HTML document.
<i>dtPDF</i>	Save the document file to disk as a PDF document.
<i>dtDOCX</i>	Save the document file to disk as a DOCX document.

#### Learn more

- [SaveDocument Method](#) 

### 20.2.2.3.7 EAEditionTypes

#### Topics

Topic	Detail	See also
<b>General Usage</b>	<p>The <i>EAEditionTypes</i> enumeration identifies the current level of licensed functionality available</p> <pre> EAEditionTypes theEdition = theRepository. GetEAEdition(); if ( theEdition == EAEditionTypes.piDesktop )     ... else if ( theEdition == EAEditionTypes. piProfessional )     ... </pre> <p>The enumeration defines the following formal values:</p> <ul style="list-style-type: none"> <li>• <i>piLite</i></li> <li>• <i>piDesktop</i></li> <li>• <i>piProfessional</i></li> <li>• <i>piCorporate</i></li> <li>• <i>piBusiness</i></li> <li>• <i>piSystemEng</i></li> <li>• <i>piUltimate</i></li> </ul> <p>There is no separate value for the trial edition; the <i>Repository.GetEAEdition()</i> function returns the appropriate <b>EAEditionTypes</b> value for whichever edition the user has selected to trial.</p>	

Topic	Detail	See also

#### 20.2.2.3.8 *EnumRelationSetType*

This enumeration represents values returned from the *GetRelationSet* method of the **Element** object.

Value	Meaning
<i>rsDependEnd</i>	List of elements that depend on the current element.
<i>rsDependStart</i>	List of elements that the current element depends on.
<i>rsGeneralizeEnd</i>	List of elements that are generalized by the current element.
<i>rsGeneralizeStart</i>	List of elements that the current element generalizes.
<i>rsParents</i>	List of all parent elements of the current element.
<i>rsRealizeEnd</i>	List of elements that are realized by the current element.
<i>rsRealizeStart</i>	List of elements that the current element realizes.

#### Learn more

- [Element Class](#) 

#### 20.2.2.3.9 *ExportPackageXMIFlag*

The *ExportPackageXMIFlag* enumeration is used in package control, when exporting to XML.

Value	Meaning
<i>epSaveToStub</i>	Export this package with only immediate children (child packages are included as stubs only).

#### 20.2.2.3.10 MDGMenus

Use this enumeration when providing the *HiddenMenus* property to *MDG\_GetProperty*.

These options are exclusive of one another and can be read or added to hide more than one menu.

Value	Meaning
<i>mgBuildProject</i>	Hide <b>Build Project</b> menu option.
<i>mgMerge</i>	Hide <b>Merge</b> menu option.
<i>mgRun</i>	Hide <b>Run</b> menu option.

#### Learn more

- [MDG\\_GetProperty](#)<sup>[3102]</sup>

#### 20.2.2.3.11 MessageFlag

The *MessageFlag* enumeration is used in both the **SendMailMessage** and **ComposeMailMessage** methods of the MailInterface, to specify a flag to attach to the message.

Value	Meaning
<b>mfNone</b>	Do not flag the message.
<b>mfComplete</b>	Flag the message as 'Complete'.
<b>mfPurple</b>	Flag the message with a 'Purple' flag.
<b>mfOrange</b>	Flag the message with a 'Orange' flag.
<b>mfGreen</b>	Flag the message with a 'Green' flag.
<b>mfYellow</b>	Flag the message with a 'Yellow' flag.
<b>mfBlue</b>	Flag the message with a 'Blue' flag.
<b>mfRed</b>	Flag the message with a 'Red' flag.

### 20.2.2.3.12 *ObjectType*

The **ObjectType** enumeration identifies Enterprise Architect object types even when referenced through a Dispatch interface. For example:

```

var treeSelectedType = Repository.GetTreeSelectedItemType();

switch ( treeSelectedType )
{
    case otElement :
    {
        // Code for when an element is selected
        var theElement as EA.Element;
        theElement = Repository.GetTreeSelectedObject();

        break;
    }
    case otPackage :
    {
        // Code for when a package is selected
        var thePackage as EA.Package;
        thePackage = Repository.GetTreeSelectedObject();

        break;
    }
}

```

#### Valid Enumeration Values

```

otAttribute
otAttributeConstraint
otAttributeTag
otAuthor
otClient
otCollection
otConnector
otConnectorConstraint
otConnectorEnd
otConnectorTag
otConstraint
otCustomProperty
otDatatype
otDiagram
otDiagramLink
otDiagramObject
otEffort
otElement
otEventProperties
otEventProperty
otFile
otIssue
otMailInterface
otMethod
otMethodConstraint
otMethodTag
otMetric
otModel
otNone
otPackage
otParameter
otParamTag
otPartition
otProject
otProjectIssues
otProjectResource
otProperties
otProperty

```

```

ot PropertyType
ot Reference
ot Repository
ot Requirement
ot Resource
ot Risk
ot RoleTag
ot Scenario
ot ScenarioExtension
ot ScenarioStep
ot Stereotype
ot Swimlane
ot SwimlaneDef
ot Swimlanes
ot TaggedValue
ot Task
ot Term
ot Test
ot Transition

```

### 20.2.2.3.13 PropType

The *PropType* enumeration gives the automation programmer an indication of what sort of data is going to be stored by this property.

Value	Meaning
<i>ptArray</i>	An array containing values of any type.
<i>ptBoolean</i>	<b>True</b> or <b>False</b> .
<i>ptEnum</i>	A string being an entry in the semi-colon separated list specified in the validation field of the Property.
<i>ptFloatingPoint</i>	4 or 8 byte floating point value.
<i>ptInteger</i>	16-bit or 32-bit signed integer.
<i>ptString</i>	Unicode string.

### 20.2.2.3.14 ReloadType

This enumeration represents values returned from the *GetReloadItem* and *PeekReloadItem* methods of the *ModelWatcher* Class. It has four possible values, which define the type of change that was made to a model.

Value	Meaning
<i>rtElement</i>	The <i>Item</i> parameter represents a particular element that must be reloaded.
<i>rtEntireModel</i>	Entire model must be reloaded to ensure that all changes are reloaded.

Value	Meaning
<i>rtNone</i>	No change in the model.
<i>rtPackage</i>	The <i>Item</i> parameter represents a particular package that must be reloaded.

#### 20.2.2.3.15 ScenarioDiagramType

The *ScenarioDiagramType* enumeration provides the following enumeration values to the **Project.GenerateDiagramFromScenario()** method. They specify the type of diagram to generate.

Value	Meaning	See also
<i>sdActivity</i>	Generate an Activity diagram.	<a href="#">Generated Activity diagram</a> <sup>[978]</sup>
<i>sdActivityWithAction</i>	Generate an Activity diagram with an Action.	<a href="#">Generated Activity diagram</a> <sup>[978]</sup>
<i>sdActivityWithActionPin</i>	Generate an Activity diagram with an ActionPin.	<a href="#">Generated Activity diagram</a> <sup>[978]</sup>
<i>sdActivityWithActivityParameter</i>	Generate an Activity diagram with an ActivityParameter.	<a href="#">Generated Activity diagram</a> <sup>[978]</sup>
<i>sdRobustness</i>	Generate a Robustness diagram.	<a href="#">Generated Robustness diagram</a> <sup>[984]</sup>
<i>sdRuleFlow</i>	Generate a RuleFlow diagram	<a href="#">Generated RuleFlow diagram</a> <sup>[980]</sup>
<i>sdSequence</i>	Generate a Sequence diagram.	<a href="#">Generated Sequence diagram</a> <sup>[982]</sup>
<i>sdState</i>	Generate a State Machine diagram.	<a href="#">Generated State Machine diagram</a> <sup>[980]</sup>

#### Learn more

- [Project.GenerateDiagramFromScenario\(\) Method](#) <sup>[2968]</sup>



### 20.2.2.3.16 ScenarioStepType

The *ScenarioStepType* enumeration is used to identify the steps of a scenario, and the entity performing the step.

Value	Meaning	See also
<i>stActor</i>	Identify that the step is an action performed by an actor.	
<i>stSystem</i>	Identify that the step is an action performed by the system.	

#### Learn more

- [Scenario Step Class](#) <sup>[2905]</sup>

### 20.2.2.3.17 ScenarioTestType

The *ScenarioTestType* enumeration provides the following enumeration values to the **Project.GenerateTestFromScenario()** method. They specify the type of test to generate.

Value	Meaning	See also
<i>stExternal</i>	Generate an external Test Case element.	
<i>stInternal</i>	Generate an internal test.	

#### Learn more

- [Project.GenerateTestFromScenario\(\) Method](#) <sup>[2971]</sup>
- [Generate Test Classes](#) <sup>[986]</sup>

### 20.2.2.3.18 XMIType

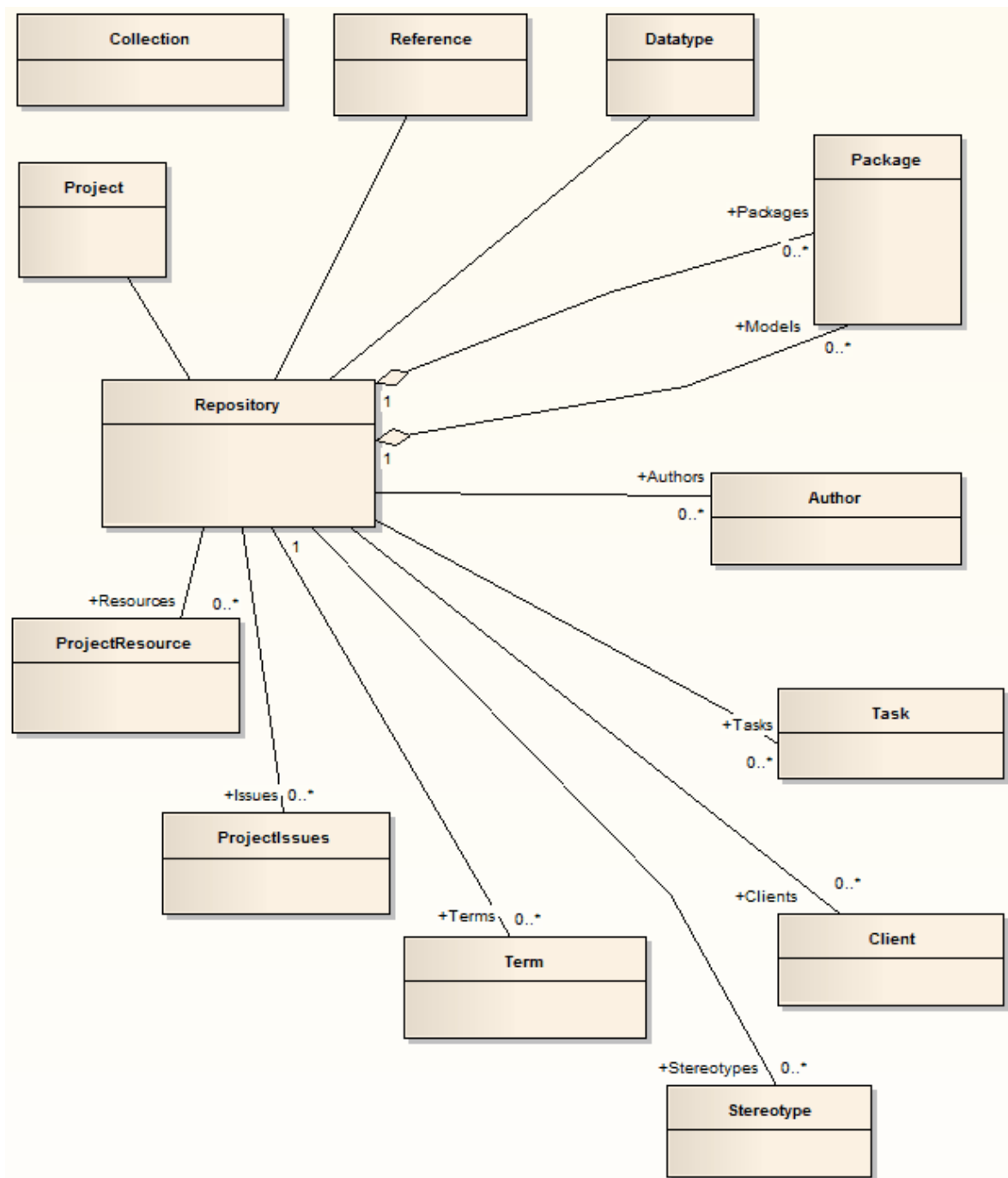
The following enumeration values are used in the *Project.ExportPackageXML()* method. They enable specification of the XML export type.

*xmiEADefault*  
*xmiRoseDefault*  
*xmiEA10*  
*xmiEA11*  
*xmiEA12*  
*xmiEA20*  
*xmiEA21*  
*xmiMOF13*  
*xmiMOF14*  
*xmiRose10*  
*xmiRose11*  
*xmiRose12*

### 20.2.2.4 Repository Package

The *Repository* package contains the high level system objects and entry point into the model itself using the *Models* collection and the other system level collections.

This diagram illustrates the **Repository** and its first level functions and collections.



[Learn more](#)

- [Repository Class](#) 2850

Learning Center topics

- Alt+F1 | [Enterprise Architect](#) | [Automation](#) | [Automation](#) | [Introduction to Automation](#)

**20.2.2.4.1 Author Class**

An **Author** object represents a named model author. Authors can be accessed using the Repository **Authors** collection.

Associated table in .EAP file    *t\_authors*

Author Attributes

Attribute	Type	Notes
<b>Name</b>	<i>String</i>	Read/Write The Author name.
<b>Notes</b>	<i>String</i>	Read/Write Notes about the author.
<b>ObjectType</b>	<a href="#">ObjectTy</a> <a href="#">pe</a> <sup>[2822]</sup>	Read only Distinguishes objects referenced through a Dispatch interface.
<b>Roles</b>	<i>String</i>	Read/Write Roles the author might play in this project.

Author Methods

Method	Type	Notes
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.
<b>Update ()</b>	<i>Boolean</i>	Updates the current Author object after modification or appending a new item. If <b>false</b> is returned, check the <b>GetLastError</b> function for more information.

#### 20.2.2.4.2 Client Class

A **Client** represents one or more people or organizations related to the project. Clients can be accessed using the Repository *Clients* collection.

**Associated table in .EAP file** *t\_clients*

##### Client Attributes

Attribute	Type	Notes
<b>Email</b>	<i>String</i>	Read/Write The client's email address.
<b>Fax</b>	<i>String</i>	Read/Write The client's fax number.
<b>Mobile</b>	<i>String</i>	Read/Write The client's mobile phone number, if available.
<b>Name</b>	<i>String</i>	Read/Write The client's name.
<b>Notes</b>	<i>String</i>	Read/Write Notes about the client.
<b>ObjectType</b>	<a href="#">Object Type</a> e[2822]	Read only Distinguishes objects referenced through the <i>Dispatch</i> interface.
<b>Organization</b>	<i>String</i>	Read/Write The client's associated organization.
<b>Phone1</b>	<i>String</i>	Read/Write The client's main phone number.
<b>Phone2</b>	<i>String</i>	Read/Write The client's second phone number.
<b>Roles</b>	<i>String</i>	Read/Write Roles this client might play in the project.

Attribute	Type	Notes

#### Client Methods

Method	Type	Notes
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.
<b>Update ()</b>	<i>Boolean</i>	Updates the current Client object after modification or appending a new item. If <b>false</b> is returned, check the <b>GetLastError</b> function for more information.

#### 20.2.2.4.3 Collection Class

**Collection** is the main collection Class used by all elements within the Automation Interface. It contains methods to iterate through the collection, refresh the collection and delete an item from the collection.

It is important to realize that when **AddNew** is called, the item is not automatically added to the current collection. The typical steps are:

- Call **AddNew** to add a new item
- Modify the item as required
- Call **Update** on the item to save it to the database
- Call **Refresh** on the collection to include it in the current set

**Delete** is the same; until **Refresh** is called, the collection still contains a reference to the deleted item, which should not be called.

Each method can be used to iterate through the collection for languages that support this type of construct.

#### Collection Attributes

Attribute	Type	Notes	
<b>Count</b>	<i>Short</i>	Read only The number of objects referenced by this list.	
<b>ObjectType</b>	<i>ObjectType</i>	Read only Distinguishes objects referenced through a Dispatch interface.	<a href="#">ObjectType</a> <sup>[2822]</sup>

#### Collection Methods

Method	Type	Notes	See also
<b>AddNew (string Name, string Type)</b>	<i>Object</i>	<p>Adds a new item to the current collection.</p> <p>Note that the interface is the same for all collections; you must provide a <i>Name</i> and <i>Type</i> argument. What these arguments are used for depends on the actual collection being accessed. For example, when adding a new element to the Elements collection, the Type string can be either a basic UML element type or a fully qualified element type (stereotype) defined by a profile, such as SysML::Requirement, differentiating it from a standard requirement.</p> <p>Also note that you must call <b>Update()</b> on the returned object to complete the <b>AddNew</b>. If <b>Update()</b> is not called the object is left in an indeterminate state.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• Name: String</li> <li>• Type: String (up to 30 characters long)</li> </ul>	
<b>Delete (short index)</b>	<i>Void</i>	<p>Deletes the item at the selected reference.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• index: Short</li> </ul>	
<b>DeleteAt (short index, boolean Refresh)</b>	<i>Void</i>	<p>Deletes the item at the selected index. The second parameter is currently unused.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• index: Short</li> <li>• Refresh: Boolean</li> </ul>	
<b>GetAt (short index)</b>	<i>Object</i>	<p>Retrieves the array object using a numerical index. If the index is out of bounds, an error occurs.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• index: Short</li> </ul>	
<b>GetByName (string Name)</b>	<i>Object</i>	<p>Gets an item in the current collection by name.</p> <p>If the collection does not contain any items, the method returns a null value. If the collection contains items, but it was unable to find an object with the specified name, the method raises an exception.</p> <p>Only supported for the following collections: <i>Model</i>, <i>Package</i>, <i>Element</i>, <i>Diagram</i> and element <i>TaggedValue</i>.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• Name: String</li> </ul>	<a href="#">Package Class</a> <a href="#">Element Class</a> <a href="#">Diagram Class</a> <a href="#">TaggedValue Class</a>
<b>GetLastError ()</b>	<i>String</i>	<p>Returns a string value describing the most recent error that occurred in relation to this object.</p>	

Method	Type	Notes	See also
<b>Refresh ()</b>	<i>Void</i>	Refreshes the collection by re-querying the model and reloading the collection. Should be called after adding a new item or after deleting an item.	
<b>Update ()</b>	<i>Boolean</i>	Updates the current Collection object after modification or appending a new item.  If <b>false</b> is returned, check the <b>GetLastError ()</b> function for more information.	

#### 20.2.2.4.4 Datatype Class

A **Datatype** is a named type that can be associated with attribute or method types. It typically is related to either code engineering or database modeling. Datatypes also indicate which language or database system they relate to. Datatypes can be accessed using the Repository **Datatypes** collection.

Associated table in .EAP file    *t\_datatypes*

#### Datatype Attributes

Attribute	Type	Notes
<b>DatatypeID</b>	<i>Long</i>	Read/Write  The instance ID for this datatype within the current model; this is system maintained.
<b>DefaultLen</b>	<i>Long</i>	Read/Write  The default length (DDL only).
<b>DefaultPrec</b>	<i>Long</i>	Read/Write  The default precision (DDL only).
<b>DefaultScale</b>	<i>Long</i>	Read/Write  The default scale (DDL only).
<b>GenericType</b>	<i>String</i>	Read/Write  The associated generic type for this data type.
<b>HasLength</b>	<i>String</i>	Read/Write  Indicates whether the datatype has a length component.

Attribute	Type	Notes
<b>MaxLen</b>	<i>Long</i>	Read/Write The maximum length (DDL only).
<b>MaxPrec</b>	<i>Long</i>	Read/Write The maximum precision (DDL only).
<b>MaxScale</b>	<i>Long</i>	Read/Write The maximum scale (DDL only).
<b>Name</b>	<i>String</i>	Read/Write The datatype name (such as <i>integer</i> ). This appears in the related drop-down datatype lists where appropriate.
<b>ObjectType</b>	<a href="#">ObjectType</a> <small>e[2822]</small>	Read only Distinguishes objects referenced through a Dispatch interface.
<b>Product</b>	<i>String</i>	Read/Write The datatype product, such as Java, C++ or Oracle.
<b>Size</b>	<i>Long</i>	Read/Write The datatype size.
<b>Type</b>	<i>String</i>	Read/Write The type can be <i>DDL</i> for database datatypes or <i>Code</i> for language datatypes.
<b>UserDefined</b>	<i>Long</i>	Read/Write Indicates if the datatype is a user defined type or system generated. Datatypes distributed with Enterprise Architect are all system generated. Datatypes created in the Datatype dialog are marked <b>1 (true)</b> .

#### Datatype Methods

Method	Type	Notes
<b>GetLastError() ( )</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.



Method	Type	Notes
<b>Update ()</b>	<i>Boolean</i>	Updates the current Datatype object after modification or appending a new item.  If <b>false</b> is returned, check the <b>GetLastError</b> function for more information.

#### 20.2.2.4.5 EventProperties Class

An **EventProperties** object is passed to *BroadcastFunctions* to facilitate parameter passing.

##### EventProperties Attributes

Attribute	Type	Notes
<b>Count</b>	<i>Long</i>	Read only  The number of parameters being passed to this broadcast event.
<b>ObjectType</b>	<a href="#">ObjectType</a> [2822]	Read only  Distinguishes objects referenced through a Dispatch interface.

##### EventProperties Methods

Method	Type	Notes
<b>Get (object Index)</b>	<a href="#">EventProperty</a> [2833]	Read only  Returns an <i>EventProperty</i> in the list, raising an error if <i>Index</i> is out of range.  Parameters: <ul style="list-style-type: none"><li>Index: Variant - can either be a number representing a zero-based index into the array, or a string representing the name of the <i>EventProperty</i>: for example, <i>Props.Get(3)</i> or <i>Props.Get("ObjectID")</i></li></ul>

#### 20.2.2.4.6 EventProperty Class

**EventProperty** objects are always part of an **EventProperties** collection, and are passed to Add-In methods responding to broadcast events.

##### EventProperty Attributes

Attribute	Type	Notes
<b>Description</b>	<i>String</i>	An explanation of what this property represents.
<b>Name</b>	<i>String</i>	A string distinguishing this property from others in the list.

Attribute	Type	Notes
<b>ObjectType</b>	<a href="#">ObjectTyp</a> <a href="#">e</a> <sup>[2822]</sup>	Distinguishes objects referenced through a Dispatch interface.
<b>Value</b>	<i>Variant</i>	A string, number or object reference representing the property value.

#### Learn more

- [EventProperties Class](#)<sup>[2833]</sup>
- [Broadcast Events](#)<sup>[3029]</sup>

#### 20.2.2.4.7 *ModelWatcher Class*

The **ModelWatcher** object enables an automation client to track changes in a particular model.

#### ModelWatcher Attributes

Attribute	Type	Notes
<b>ObjectType</b>	<a href="#">ObjectTyp</a> <a href="#">e</a> <sup>[2822]</sup>	Read only Distinguishes objects referenced through a Dispatch interface.

#### ModelWatcher Methods

Methods	Type	Notes
<b>GetReloadItem</b> ( <b>object Item</b> )	<a href="#">ReloadTyp</a> <a href="#">e</a> <sup>[2823]</sup>	The object that must be reloaded in order to see all changes is returned through the <i>Item</i> parameter. If there are no changes or the entire model must be reloaded, this value is returned as <b>null</b> (C#) or <b>Nothing</b> (VB).  Calling this method clears the records so that the next time it is called the return values refer only to new changes.  Returns a value from the <b>ReloadType</b> enumeration that specifies which type of change, if any, has occurred.  Parameters: <ul style="list-style-type: none"> <li>Item: Object</li> </ul>
<b>PeekReloadItem</b>	<a href="#">ReloadTyp</a> <a href="#">e</a> <sup>[2823]</sup>	This method behaves identically to <b>GetReloadItem()</b> but does not clear the change record.

#### Notes

- After your model has been loaded, you only create the ModelWatcher once; if you **reload** the model, or load another model, the created ModelWatcher is still valid

#### Learn more

- [Refresh View of Shared Project](#)<sup>[308]</sup>

#### 20.2.2.4.8 Package Class

A **Package** object corresponds to a Package element in the Enterprise Architect Project Browser. Packages can be accessed either through the **Repository Models** collection (a Model is a special form of Package) or through the Package **Packages** collection.

Note that a Package has an Element object as an attribute; this corresponds to an Enterprise Architect Package element in the *t\_object* table and is used to associate additional information (such as scenarios and constraints) with the logical package.

To set additional information for a package, reference the Element object directly. Also note that if you add a Package to a diagram, you should add an instance of the element (not the Package itself) to the **DiagramObject** Class for a diagram.

**Associated table in .EAP file**    *t\_package*

#### Package Attributes

Attribute	Type	Notes
<b>Alias</b>	<i>String</i>	Read only Alias
<b>BatchLoad</b>	<i>Long</i>	Read/Write Flag to indicate that the package is batch loaded during batch import from controlled packages. <b>Not currently used.</b>
<b>BatchSave</b>	<i>Long</i>	Read/Write Boolean value to indicate whether the package is included in the batch XML export list or not.
<b>CodePath</b>	<i>String</i>	Read/Write The path where associated source code is found. <b>Not currently used.</b>
<b>Connectors</b>	<a href="#">Collection</a> <sup>[2829]</sup>	Read only The collection of connectors.

Attribute	Type	Notes
<b>Created</b>	<i>Date</i>	Read/Write Date the package was created.
<b>Diagrams</b>	<a href="#">Collection</a> <small>[2829]</small>	Read only A collection of diagrams contained in this package.
<b>Element</b>	<a href="#">Element</a> <small>[2881]</small>	Read only The associated element object; use to get/set common information such as Stereotype, Complexity, Alias, Author, Constraints, Tagged Values and Scenarios.
<b>Elements</b>	<a href="#">Collection</a> <small>[2829]</small>	Read only A collection of elements that belong to this package.
<b>Flags</b>	<i>String</i>	Read/Write Extended information about the package.
<b>IsControlled</b>	<i>Boolean</i>	Read/Write Indicates if the package has been marked as <i>Controlled</i> .
<b>IsModel</b>	<i>Boolean</i>	Read only Indicates if the package is a model or a package.
<b>IsNamespace</b>	<i>Boolean</i>	Read/Write <b>True</b> indicates that 'package is a Namespace root'. Use <b>0</b> and <b>1</b> to set <b>False</b> and <b>True</b> .
<b>IsProtected</b>	<i>Boolean</i>	Read/Write Indicates if the package has been marked as <i>Protected</i> .
<b>IsVersionControlled</b>	<i>Boolean</i>	Read only Indicates whether or not this package is under version control.
<b>LastLoadDate</b>	<i>Date</i>	Read/Write The date XML was last loaded for the package.

Attribute	Type	Notes
<b>LastSaveDate</b>	<i>Date</i>	Read/Write The date XML was last saved from the package.
<b>LogXML</b>	<i>Boolean</i>	Read/Write Indicates if XML export information is to be logged.
<b>Modified</b>	<i>Date</i>	Read/Write Date the package was last modified.
<b>Name</b>	<i>String</i>	Read/Write The name of the package.
<b>Notes</b>	<i>String</i>	Read/Write Notes about this package.
<b>ObjectType</b>	<a href="#">Object Type</a> <small>[2822]</small>	Read only Distinguishes objects referenced through a Dispatch interface.
<b>Owner</b>	<i>String</i>	Read/Write. The package owner when using controlled packages.
<b>PackageGUID</b>	<i>Variant</i>	Read only The global Package ID; valid across models.
<b>PackageID</b>	<i>Long</i>	Read only The local Package ID number. Valid only in this model file.
<b>Packages</b>	<a href="#">Collection</a> <small>[2829]</small>	Read only A collection of contained packages that can be walked through.
<b>ParentID</b>	<i>Long</i>	Read/Write The ID of the package that is the parent of this one. <b>0</b> indicates this package is a <i>model</i> (that is, it has no parent).
<b>TreePos</b>	<i>Long</i>	Read/Write

Attribute	Type	Notes
		The relative position in the tree compared to other packages (use to sort packages).
<b>UMLVersion</b>	<i>String</i>	Read/Write The UML version for XML export purposes.
<b>UseDTD</b>	<i>Boolean</i>	Read/Write Indicates if a DTD is to be used when exporting XML.
<b>Version</b>	<i>String</i>	Read/Write The version of the package.
<b>XMLPath</b>	<i>String</i>	Read/Write The path to which the XML is saved when using controlled packages.

#### Package Methods

Method	Type	Notes
<b>ApplyGroupLock (string aGroupName)</b>	<i>Boolean</i>	Applies a group lock to the package object, for the specified group, on behalf of the current user.  Returns <b>true</b> if the operation is successful; returns <b>false</b> if the operation is unsuccessful. Use <b>GetLastError ()</b> to retrieve error information.  Parameters: <ul style="list-style-type: none"> <li>aGroupName: String - The name of the security group for which to apply the lock</li> </ul>
<b>ApplyGroupLockRecursive (string aGroupName)</b>	<i>Boolean</i>	Applies a group lock to the package object, object, and all of the package, diagrams and elements contained within that package, for the specified group, on behalf of the current user.  Returns <b>true</b> if the operation is successful; returns <b>false</b> if the operation is unsuccessful. Use <b>GetLastError ()</b> to retrieve error information.  Parameters: <ul style="list-style-type: none"> <li>aGroupName: String - The name of the security group for which to apply the lock</li> </ul>
<b>ApplyUserLock ()</b>	<i>Boolean</i>	Applies a user lock to the package object for the current user.

Method	Type	Notes
		Returns <b>true</b> if the operation is successful; returns <b>false</b> if the operation is unsuccessful. Use <b>GetLastError ()</b> to retrieve error information.
<b>ApplyUserLockRecursive ()</b>	<i>Boolean</i>	Applies user locks to the package object, and all of the packages, diagrams and elements contained within that package.  Returns <b>true</b> if the operation is successful; returns <b>false</b> if the operation is unsuccessful. Use <b>GetLastError ()</b> to retrieve error information.
<b>Clone</b>	<i>LDISPAT CH</i>	Inserts a copy of the package into the same parent as the original package.  Returns the newly-created package.
<b>FindObject (string DottedID)</b>	<i>LPDISPAT CH</i>	Returns a package, element, attribute or operation matching the parameter <i>DottedID</i> .  If the <i>DottedID</i> is not found, an error is returned: <i>Can't find matching object</i> .  Parameters: <ul style="list-style-type: none"> <li>DottedID: String - Is in the form <i>object.object.object</i> where <i>object</i> is replaced by the name of a package, element attribute or operation; examples include <i>MyNamespace.Class1</i>, <i>CStudent.m_Name</i>, <i>MathClass.DoubleIt(int)</i></li> </ul>
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.
<b>ReleaseUserLock ()</b>	<i>Boolean</i>	Removes an existing User or Group lock from the package object.  Returns <b>true</b> if the operation is successful; returns <b>false</b> if the operation is unsuccessful. Use <b>GetLastError ()</b> to retrieve error information.
<b>ReleaseUserLockRecursive ()</b>	<i>Boolean</i>	Releases user locks and group locks from the package object, and all of the packages, diagrams and elements contained within that package.  Returns <b>true</b> if the operation is successful; returns <b>false</b> if the operation is unsuccessful. Use <b>GetLastError ()</b> to retrieve error information
<b>SetReadOnly (boolean ReadOnly,</b>	<i>Void</i>	Sets a Package Flag to mark a package as <b>ReadOnly=1</b> .  If Project Security is enabled, the user must have <b>Configure</b>

Method	Type	Notes
<b>boolean IncludeSubPkgs)</b>		<p><b>Packages</b> permission to use this method.</p> <p>Throws an exception if the operation fails due to the user not having <b>Configure Packages</b> permission; use <b>GetLastError ()</b> to retrieve error information.</p> <p>Parameters;</p> <ul style="list-style-type: none"> <li>ReadOnly: Boolean - Sets or clears the <i>Read Only</i> flag on the package(s); if: <ul style="list-style-type: none"> <li><b>False</b>, any <i>Read Only</i> flag is removed from the package</li> <li><b>True</b>, a <i>Read Only</i> flag is applied to the package</li> </ul> </li> <li>IncludeSubPkgs: Boolean - Indicates whether to set/reset the <i>Read Only</i> flag on just the object package, or on the object package and all of the nested sub-packages that it contains; if: <ul style="list-style-type: none"> <li><b>False</b>, only the flag on the object package is set or cleared</li> <li><b>True</b>, flags are set (or cleared, according to the <i>ReadOnly</i> parameter) for the object package plus all of the nested sub-packages that it contains</li> </ul> </li> </ul> <p>When working with version controlled packages, the <i>Read Only</i> flag can be applied to packages whether they are checked-in or checked-out.</p> <p>User Security applies to setting this flag - if you are prevented from editing the package, you are also prevented from setting the flag.</p>
<b>Update ()</b>	<i>Boolean</i>	<p>Updates the current package object after modification or appending a new item.</p> <p>If <b>false</b> is returned, check the <b>GetLastError ()</b> function for more information.</p> <p>Note that a package object also has an <i>element</i> component that must be taken into account; the package object contains information about the package attributes such as hierarchy or contents.</p> <p>The element attribute contains information about, for example, Stereotypes, Constraints or Files - all the attributes of a typical element.</p>
<b>VersionControlAdd (string ConfigGuid, string XMLFile, string Comment, boolean KeepCheckedOut)</b>	<i>Void</i>	<p>Places the package under version control, using the specified Version Control Configuration and the specified XML filename.</p> <p>Throws an exception if the operation fails; use <b>GetLastError ()</b> to retrieve error information.</p> <p>It is recommended that the package be saved using <i>Update()</i> before calling <b>VersionControlAdd ()</b>, so that any outstanding changes are not lost.</p>



Method	Type	Notes
		<p>Parameters:</p> <ul style="list-style-type: none"> <li>• ConfigGuid: String - Name corresponding to the Unique ID of the version control configuration to use</li> <li>• XMLFile: String - Name of the XML file to use for this package; this filename is relative to the Working Copy folder specified for the Config</li> <li>• Comment: String - Log message that is added to the version controlled file's history (where applicable)</li> <li>• KeepCheckedOut: Boolean - Specify <b>True</b> to add to version control and keep package checked-out</li> </ul>
<b>VersionControlCheckin (string Comment)</b>	<i>Void</i>	<p>Perform checkin of the version controlled package (also see <b>VersionControlCheckinEx</b>, below).</p> <p>Throws an exception if the operation fails; use <b>GetLastError ( )</b> to retrieve error information.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• Comment: String - Log message that is added to the version controlled file's history (where applicable)</li> </ul>
<b>VersionControlCheckinEx (string Comment, boolean PreserveCrossPkgRefs)</b>	<i>Void</i>	<p>Perform checkin of the version controlled package.</p> <p>Throws an exception if the operation fails; use <b>GetLastError ( )</b> to retrieve error information.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• Comment: String - Log message that is added to the version controlled file's history (where applicable)</li> <li>• PreserveCrossPkgRefs: Boolean - Flag to indicate whether to preserve or discard pre-existing Cross Package References when checking-in; this parameter overrides the setting in the Options dialog, XML Specifications page</li> </ul> <p>Unsatisfied cross package references are preserved or discarded according to this setting, without prompting the user; see <i>Learn more</i> below</p>
<b>VersionControlCheckout (string Comment)</b>	<i>Void</i>	<p>Perform checkout of the version controlled package.</p> <p>Throws an exception if the operation fails; use <b>GetLastError ( )</b> to retrieve error information.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• Comment: String - Log message that is added to the version controlled file's history (where applicable)</li> </ul> <p>When working in an environment that uses a Private Model deployment and your model contains a significant number of cross-package references, it is recommended that you invoke</p>

Method	Type	Notes
		the <b>Repository.ScanXMIAndReconcile ( )</b> method from time to time, following the re-importation of controlled packages - for example, after using <b>Package.VersionControlGetLatest ( )</b> to update a number of packages, or after performing a number of package check-outs.
<b>VersionControlGetLatest ( boolean ForceImport )</b>	<i>Void</i>	<p>Updates the local working copy of the package file associated with the object package, before re-importing the package data from the package file.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>ForceImport: Boolean - Used if the package data in the model is found to be up-to-date with respect to the version controlled package file; if: <ul style="list-style-type: none"> <li><b>False</b>, the package data that exists in the model is accepted as being up-to-date and no attempt is made to re-import data from the package file</li> <li><b>True</b>, Enterprise Architect re-imports the package from the package file regardless</li> </ul> </li> </ul> <p>See also the version control menu option <b>Get Latest</b>.</p> <p>When working in an environment that uses a Private Model deployment and your model contains a significant number of cross-package references, it is recommended that you invoke the <b>Repository.ScanXMIAndReconcile ( )</b> method from time to time, following the re-importation of controlled packages - for example, after using <b>Package.VersionControlGetLatest ( )</b> to update a number of packages, or after performing a number of package check-outs.</p>
<b>VersionControlGetStatus ( )</b>	<i>Long</i>	<p>Returns the version control status of the package, as recorded in the current Enterprise Architect project database.</p> <p>Throws an exception if the operation fails; use <b>GetLastError ( )</b> to retrieve error information.</p> <p>Return value maps to the following enumerated type:</p> <pre>enum EnumCheckOut Status {     csUncontrolled = 0,     csCheckedIn,     csCheckedOut ToThisUser,     csReadOnlyVersion,     csCheckedOut ToAnotherUser,     csOfflineCheckedIn,     csCheckedOut OfflineByUser,     csCheckedOut OfflineByOther,     csDeleted, };</pre> <p><i>csUncontrolled</i> - Either unable to communicate with the version control provider associated with the package or the package file</p>

Method	Type	Notes
		<p>is unknown to the provider.</p> <p><i>csCheckedIn</i> - The package is not checked-out to anybody in the current project database.</p> <p><i>csCheckedOutToThisUser</i> - The package is marked as checked-out to the current user, in the current project database.</p> <p><i>csReadOnlyVersion</i> - Package is marked as read-only; an earlier revision of the package has been retrieved from version control.</p> <p><i>csCheckedOutToAnotherUser</i> - The package is marked as checked-out in the current project database, by a user other than the current user.</p> <p><i>csOfflineCheckedIn</i> - The package is not checked-out to anybody in the current project database; however, the version control configuration associated with the package was unable to connect to the VC server.</p> <p><i>csCheckedOutOfflineByUser</i> - Indicates that the package was 'checked out' in this database, by this user, whilst disconnected from version control.</p> <p><i>csCheckedOutOfflineByOther</i> - Indicates that the package was checked out in this project database, by another user, whilst disconnected from version control.</p> <p><i>csDeleted</i> - The package file has been deleted from version control.</p>
<b>VersionControlPutLatest (string CheckInComment)</b>	<i>Void</i>	<p>Perform a checkin of the version controlled package, whilst keeping the package checked-out.</p> <p>Throws an exception if the operation fails; use <b>GetLastError ( )</b> to retrieve error information.</p> <p>When a package that was previously marked as <i>Checked Out Offline</i>, is successfully 'Put' (checkedin) to version control, that package's flags are updated to clear the <i>Checked Out Offline</i> indicator.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>Comment: String - Log message added to the version controlled file's history (where applicable)</li> </ul>
<b>VersionControlRemove ( )</b>	<i>Void</i>	<p>Removes version control from the package.</p> <p>Throws an exception if the operation fails; use <b>GetLastError ( )</b> to retrieve error information.</p>
<b>VersionControlResynchPkg Status (boolean ClearSettings)</b>		<p>Synchronizes the version control status of the single object package recorded in your current model with the package status reported by your version control provider.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>ClearSettings: Boolean - used if the package file associated with the specified package is reported by the</li> </ul>

Method	Type	Notes
		<p>version control provider as uncontrolled; if ClearSettings is:</p> <ul style="list-style-type: none"> <li>• <b>True</b>, the version control settings are cleared from the package</li> <li>• <b>False</b>, the version control settings remain unchanged</li> </ul>

#### Learn more

- [Repository Models](#) <sup>[2852]</sup>
- [Package Packages](#) <sup>[2837]</sup>
- [Version Control - Get Latest](#) <sup>[428]</sup>
- [Resynchronize the Status of Version Controlled Packages](#) <sup>[444]</sup>
- [Preserving Cross Package References](#) <sup>[496]</sup>

#### 20.2.2.4.9 *ProjectIssues Class*

A **ProjectIssue** is a system-level Issue that indicates a problem or risk associated with the system as a whole. ProjectIssues can be accessed using the Repository **Issues** collection.

Associated table in .EAP file *t\_issues*

#### ProjectIssues Attributes

Attribute	Type	Notes
<b>Category</b>	<i>String</i>	Read/Write The category this issue belongs to.
<b>Date</b>	<i>Date</i>	Read/Write The date the issue item was created.
<b>DateResolved</b>	<i>Date</i>	Read/Write The date the issue was resolved.
<b>Name</b>	<i>String</i>	Read/Write The issue name (that is, the issue itself).
<b>IssueID</b>	<i>Long</i>	Read only The ID of this issue.

Attribute	Type	Notes
<b>Notes</b>	<i>String</i>	Read/Write The associated description of the issue.
<b>ObjectType</b>	<a href="#">ObjectTyp</a> <small>e<sub>[2822]</sub></small>	Read only Distinguishes objects referenced through a Dispatch interface.
<b>Owner</b>	<i>String</i>	Read/Write The owner of the issue.
<b>Priority</b>	<i>String</i>	Read/Write The issue priority - <b>Low</b> , <b>Medium</b> or <b>High</b> .
<b>Resolution</b>	<i>String</i>	Read/Write A description of the resolution.
<b>Resolver</b>	<i>String</i>	Read/Write The name of the person resolving the issue.
<b>Severity</b>	<i>String</i>	Read/Write The issue severity - <b>Low</b> , <b>Medium</b> or <b>High</b> .
<b>Status</b>	<i>String</i>	Read/Write The current status of the issue.

**ProjectIssues Methods**

Method	Type	Notes
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.
<b>Update ()</b>	<i>Boolean</i>	Updates the current Issue object after modification or appending a new item. If <b>false</b> is returned, check the <b>GetLastError</b> function for more information.

#### 20.2.2.4.10 ProjectResource Class

A **Project Resource** is a named person who is available to work on the current project in any capacity. ProjectResources can be accessed using the Repository **Resources** collection.

Associated table in .EAP file    *t\_resources*

##### ProjectResource Attributes

Attribute	Type	Notes
<b>Email</b>	<i>String</i>	The resource's email address.
<b>Fax</b>	<i>String</i>	The resource's fax number.
<b>Mobile</b>	<i>Variant</i>	The resource's mobile number, if available.
<b>Name</b>	<i>String</i>	The name of the resource.
<b>Notes</b>	<i>String</i>	A description of the resource, if appropriate.
<b>ObjectType</b>	<a href="#">ObjectType</a> <small>[2822]</small>	Read only Distinguishes objects referenced through a Dispatch interface.
<b>Organization</b>	<a href="#">Package</a> <small>[2835]</small> : <i>String</i>	The organization the resource is associated with.
<b>Phone1</b>	<i>Variant</i>	The resource's main telephone number.
<b>Phone2</b>	<i>Variant</i>	The resource's alternative telephone number.
<b>Roles</b>	<i>String</i>	The roles this resource can play in the current project.

##### ProjectResource Methods

Method	Type	Notes
<b>GetLastError() ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.

Method	Type	Notes
<b>Update ()</b>	<i>Boolean</i>	Updates the current Resource object after modification or appending a new item.  If <b>false</b> is returned, check the <b>GetLastError</b> function for more information.

#### 20.2.2.4.11 ProjectRole Class

A **ProjectRole** object represents a named project role. ProjectRoles can be accessed using the Repository **ProjectRole** collection.

Associated table in .EAP file    *t\_projectroles*

##### ProjectRole Attributes

Attribute	Type	Notes
<b>Description</b>	<i>String</i>	Read/Write The project role item description.
<b>Notes</b>	<i>String</i>	Read/Write Notes about the project role item.
<b>ObjectType</b>	<a href="#">ObjectTy pe</a> <sup>[2822]</sup>	Read only Distinguishes objects referenced through a Dispatch interface.
<b>Role</b>	<i>String</i>	Read/Write The project role item name.

##### ProjectRole Methods

Method	Type	Notes
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.
<b>Update ()</b>	<i>Boolean</i>	Updates the current ProjectRole object after modification or appending a new item.  If <b>false</b> is returned, check the <b>GetLastError</b> function for more information.

#### 20.2.2.4.12 PropertyType Class

A **PropertyType** object represents a defined property that can be applied to UML elements as a Tagged Value. PropertyTypes can be accessed using the Repository **PropertyTypes** collection.

Each PropertyType corresponds to one of the predefined Tagged Values for the model.

**Associated table in .EAP file** `t_propertytypes`

##### PropertyType Attributes

Attribute	Type	Notes
<b>Description</b>	<i>String</i>	Read/Write A short description of the property.
<b>Detail</b>	<i>String</i>	Read/Write Configuration information for the property.
<b>ObjectType</b>	<a href="#">ObjectType</a> <small>[2822]</small>	Read only Distinguishes objects referenced through a Dispatch interface.
<b>Tag</b>	<i>String</i>	Read/Write The name of the property (Tag Name).

##### PropertyType Methods:

Method	Type	Notes
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.
<b>Update ()</b>	<i>Boolean</i>	Updates the current <b>PropertyType</b> object after modification or appending a new item.  If <b>false</b> is returned, check the <b>GetLastError</b> function for more information.

#### 20.2.2.4.13 Reference Class

This **Interface** provides access to the various lookup tables within Enterprise Architect. Use the Repository **GetReferenceList()** method to get a handle to a list.

Valid lists are:

- Diagram



- Element
- Constraint
- Requirement
- Connector
- Status
- Cardinality
- Effort
- Metric
- Scenario
- Status
- Test

#### Reference Attributes

Attribute	Type	Notes
<b>Count</b>	<i>Short</i>	A count of items in the list.
<b>ObjectType</b>	<a href="#">ObjectTyp</a> <a href="#">e<sub>[2822]</sub></a>	Read only Distinguishes objects referenced through a Dispatch interface.
<b>Type</b>	<i>String</i>	The list type (for example, <i>Diagram Types</i> ).

#### Reference Methods

Method	Type	Notes
<b>GetAt (short Index)</b>	<i>String</i>	Get the item at the specified index. Parameters: <ul style="list-style-type: none"><li>• Index: Short - The index of the item to retrieve from the list</li></ul>
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.
<b>Refresh ()</b>	<i>Short</i>	Refresh the current list and return the count of items.

#### 20.2.2.4.14 Repository Class

The **Repository** is the main container of all structures such as models, packages and elements. You can iteratively begin accessing the model using the **Models** collection. It also has some convenient methods to directly access the structures without having to locate them in the hierarchy first.

**Associated table in .EAP file** <none>

#### Repository Attributes

Attribute	Type	Notes	See also
<b>Authors</b>	<a href="#">Collection</a> [2829]	Read only  This is the system <b>Authors</b> collection. This contains <b>0</b> or more <b>Author objects</b> , each of which can be associated with, for example, elements or diagrams as the item author or owner.  Use <b>AddNew ()</b> , <b>Delete ()</b> and <b>GetAt ()</b> to manage Authors.	
<b>BatchAppend</b>	<i>Boolean</i>	Read/Write  Set this property to <b>true</b> when your automation client has to rapidly insert many elements, operations, attributes and/or operation parameters.  Set to <b>false</b> when work is complete.  This can result in 10- to 20-fold improvement in adding new elements in bulk.	
<b>Clients</b>	<a href="#">Collection</a> [2829]	Read only  A list of <b>Clients</b> associated with the project. You can modify, delete and add new <b>Client objects</b> using this collection.	
<b>ConnectionString</b>	<i>String</i>	Read only  The filename/connection string of the current Repository.  For a connection string, the DBMS repository type is identified by "DBType=n;" where n is a number corresponding to the DBMS type, as follows:  <b>0</b> - MYSQL <b>1</b> - SQLSVR <b>2</b> - ADOJET <b>3</b> - ORACLE <b>4</b> - POSTGRES	

		<b>5</b> - ASA <b>7</b> - OPENEDGE <b>8</b> - ACCESS2007	
<b>Datatypes</b>	<a href="#">Collection</a> <small>[2829]</small>	Read only  The <b>Datatypes</b> collection. This contains a list of <b>Datatype objects</b> , each representing a data type definition for either data modeling or code generation purposes.	
<b>EAEdition</b>	<a href="#">EAEdition Types</a> <small>[2819]</small>	Read only  Returns the current level of core licensed functionality available.  This property returns <b>Corporate</b> when the edition is <i>Business and Software Engineering</i> , <i>Systems Engineering</i> or <i>Ultimate</i> .  Use <b>EAEditionEx</b> to identify which of these extended editions is available.	
<b>EAEditionEx</b>	<a href="#">EAEdition Types</a> <small>[2819]</small>	Read only  Returns the current level of extended licensed functionality available.	
<b>EnableCache</b>	<i>Boolean</i>	Read/Write  An optimization for pre-loading package objects when dealing with large sets of automation objects.	
<b>EnableUIUpdates</b>	<i>Boolean</i>	Read/Write  Set this property to <b>false</b> to improve the performance of changes to the model; for example, bulk addition of elements to a package. To reveal changes to the user, call <b>Repository.RefreshModelView ()</b> .	
<b>FlagUpdate</b>	<i>Boolean</i>	Read/Write  Instructs Enterprise Architect to update the Repository with the <b>LastUpdate</b> value.	
<b>InstanceGUID</b>	<i>String</i>	Read only  The identifier string identifying the Enterprise Architect runtime session.	

<b>IsSecurityEnabled</b>	<i>Boolean</i>	Read only  Checks whether User Security is enabled for the current repository.	
<b>Issues</b>	<a href="#">Collection</a> <small>[2829]</small>	Read only  The <b>System Issues</b> list. Contains <b>ProjectIssues objects</b> , each detailing a particular issue as it relates to the project as a whole.	
<b>LastUpdate</b>	<i>String</i>	Read only  The identifier string identifying the Enterprise Architect runtime session and the timestamp for when it was set.	
<b>LibraryVersion</b>	<i>Long</i>	Read only  The build number of the Enterprise Architect runtime.	
<b>Models</b>	<a href="#">Collection</a> <small>[2829]</small> of type <a href="#">Package</a> <small>[2835]</small>	Read only  <b>Models</b> are of type <b>package</b> and belong to a collection of packages. This is the top level entry point to an Enterprise Architect project file. Each model is a <b>root node</b> in the Project Browser and can contain items such as Views and packages.  A model is a special form of a package; it has a <i>ParentID</i> of <b>0</b> . By iterating through all models, you can access all the elements within the project hierarchy.  You can also use the <b>AddNew ()</b> function to create a new model. A model can be deleted, but remember that everything contained in the model is deleted as well.	
<b>ObjectType</b>	<a href="#">ObjectType</a> <small>[2822]</small>	Read only  Distinguishes objects referenced through the Dispatch interface.	
<b>ProjectGUID</b>	<i>String</i>	Read only  Returns a unique ID for the project.	
<b>ProjectRoles</b>	<a href="#">Collection</a> <small>[2829]</small>	Read only  The system Roles collection. Contains <b>0</b> or more Role objects, each of which can be	

		<p>associated with, for example, elements or diagrams as the item author or owner.</p> <p>Use <b>AddNew ()</b>, <b>Delete ()</b> and <b>GetAt ()</b> to manage Roles.</p>	
<b>PropertyTypes</b>	<a href="#">Collection</a> [2829]	<p>Read only</p> <p>Collection of <b>Property Types</b> available to the Repository.</p>	<a href="#">Property Type Class</a> [2848]
<b>Resources</b>	<a href="#">Collection</a> [2829]	<p>Read only</p> <p>Contains available <i>ProjectResource</i> objects to assign to work items within the project.</p> <p>Use the <b>add new</b>, <b>modify</b> and <b>delete</b> functions to manage resources.</p>	
<b>Stereotypes</b>	<a href="#">Collection</a> [2829]	<p>Read only</p> <p>The <b>Stereotype</b> collection. A list of <i>Stereotype objects</i> that contain information on a stereotype and which elements it can be applied to.</p>	<a href="#">Stereotype Class</a> [2873]
<b>SuppressEADialogs</b>	<i>Boolean</i>	<p>Read/Write</p> <p>Set this property in the <b>EA_OnPostNewElement</b> broadcast event to control whether Enterprise Architect should suppress showing the default Properties dialog to the user when an element is created.</p>	<a href="#">EA_OnPostNewElement</a> [3061] <a href="#">EA_OnPostNewConnector</a> [3062]
<b>SuppressSecurityDialog</b>	<i>Boolean</i>	<p>Read/Write</p> <p>Suppress the login prompt dialog that appears by default when username and password parameters passed to <i>OpenFile2</i> are invalid. For use by external automation clients only.</p>	
<b>Tasks</b>	<a href="#">Collection</a> [2829]	<p>Read only</p> <p>A list of system tasks (to do list). Each entry is a <b>Task</b> item; you can modify, delete and add new tasks.</p>	<a href="#">Task Class</a> [2874]
<b>Terms</b>	<a href="#">Collection</a> [2829]	<p>Read only</p> <p>The project <i>Glossary</i>. Each <b>Term</b> object is an entry in the Glossary. Add, modify and delete Terms to maintain the Glossary.</p>	<a href="#">Term Class</a> [2876]

--	--	--	--

### Repository Methods

Method	Type	Notes	See also
<b>ActivateDiagram (long DiagramID)</b>		Activates an already open diagram (that is, makes it the active tab) in the main Enterprise Architect user interface.  Parameters: <ul style="list-style-type: none"> <li>DiagramID: Long - the ID of the diagram to make active</li> </ul>	
<b>ActivatePerspective (string long)</b>	<i>Boolean</i>	<b>Deprecated</b> - no longer in use.	
<b>ActivateTab (string Name)</b>		Activates an open Enterprise Architect tabbed view.  Parameters: <ul style="list-style-type: none"> <li>Name: String - the name of the view to activate</li> </ul>	
<b>ActivateTechnology (string TechnologyID)</b>		Activates an enabled MDG Technology.  Parameters: <ul style="list-style-type: none"> <li>TechnologyID: String - the ID of the Technology to activate, as assigned in the MDG Technology Wizard</li> </ul>	<a href="#">Create MDG Technology File</a> <sup>[1545]</sup>
<b>ActivateToolbox (string Toolbox, long Options)</b>	<i>Boolean</i>	Activates a Toolbox page in the GUI.  The returned value is reserved for future use.  Parameters: <ul style="list-style-type: none"> <li>Toolbox: String - the name of the Toolbox page to activate</li> <li>Options: Long - reserved for future use</li> </ul>	
<b>AddDefinedSearches (string sXML)</b>		Enables you to enter a set of defined searches that last in Enterprise Architect for the life of the application; when Enterprise Architect loads again they must be inserted again by your Add-In.  Parameters: <ul style="list-style-type: none"> <li>sXML: String - the XML of the defined searches; you can get this XML by performing an <i>export</i> of the searches from the Manage Searches dialog in Enterprise Architect</li> </ul>	<a href="#">Create and Manage Searches</a> <sup>[709]</sup>

<b>AddDocumentationPath</b> ( string Name, string Path, long Type)		<p>Provides an Add-In with the ability to insert a book path into the Enterprise Architect installation directory, to display Learning Center pages on user-authored subjects (such as use of the Add-In).</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>Name: String - the top-level (root) name for the Learning Center documentation hierarchy for the Add-In (for example, Enterprise Architect)</li> <li>Path: String - the directory path to the folder to contain the Learning Center documentation structure (for example, C:\Program Files (86)\Sparx Systems\EA\Books</li> <li>Type: Long - reserved for future use; set to 0</li> </ul>	<a href="#">Learning Center</a> <sup>[74]</sup>  <a href="#">Create Learning Center Pages</a> <sup>[74]</sup>
<b>AddPerspective</b> ( string Perspective, long Options)	<i>Boolean</i>	<b>Deprecated</b> - no longer in use.	
<b>AddTab</b> ( string TabName, string ControlID)	<i>activeX custom control</i>	<p>Adds an ActiveX custom control as a tabbed window. Enterprise Architect creates a control and, if successful, returns its Unknown pointer, which can be used by the caller to manipulate the control.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>TabName: String - used as the tab caption</li> <li>ControlID: String - the ProgID of the control; for example, "CS_AddinFramework.UserControl1"</li> </ul>	<a href="#">Create a Custom View</a> <sup>[3095]</sup>
<b>AddWindow</b> ( string WindowName, string ControlID)	<i>activeX custom control</i>	<p>Adds an ActiveX custom control as a window to the Add-Ins docked window. Enterprise Architect creates a control and, if successful, returns its Unknown pointer, which can be used by the caller to manipulate the control.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>WindowName: String - used as the window title</li> <li>ControlID: String - the ProgID of the control; for example, "CS_AddinFramework.UserControl1"</li> </ul>	<a href="#">Custom Docked Window</a> <sup>[3096]</sup>  <a href="#">RemoveWindow</a> <sup>[2869]</sup>
<b>AdviseConnectorChange</b> ( 		Provides an Add-In or automation client with the ability to advise the Enterprise Architect	

<b>long ConnectorID)</b>		<p>user interface that a particular connector has changed and, if it is visible in any open diagram, to reload and refresh that connector for the user.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>ConnectorID: Long - the ID of the connector</li> </ul>	
<b>AdviseElementChange (long ObjectID)</b>		<p>Provides an Add-In or automation client with the ability to advise the Enterprise Architect user interface that a particular element has changed and, if it is visible in any open diagram, to reload and refresh that element for the user.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>ObjectID: Long - the ID of the element</li> </ul>	
<b>ChangeLoginUser (string Name, string Password)</b>	<i>Boolean</i>	<p>Sets the currently logged on user to be that specified by a name and password; this logs the user into the repository when security is enabled.</p> <p>If security is not enabled an exception (<i>Security not enabled</i>) is thrown.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>Name: String - the name of the user</li> <li>Password: String - the password of the user</li> </ul>	
<b>ClearAuditLogs (Object StartDateTime, Object EndDateTime)</b>	<i>Boolean</i>	<p>Clears all Audit Logs from the model.</p> <p>If <i>StartDateTime</i> and <i>EndDateTime</i> are not null then only log items that fall into this period are cleared.</p> <p>Returns <b>true</b> for success, <b>false</b> for failure.</p> <ul style="list-style-type: none"> <li>This method cannot be undone. It is strongly advised that you call <b>SaveAuditLogs</b> first to backup the logs</li> <li>This method might fail if the user logged into the model does not have the correct access permission</li> </ul> <p>Parameters:</p> <ul style="list-style-type: none"> <li>StartDateTime: Variant ( DateTime ) - the earliest date and time of log entries to clear</li> <li>EndDateTime; Variant ( DateTime ) - the latest date and time of log entries to clear</li> </ul>	



<b>ClearOutput (string Name)</b>		Removes all the text from a tab in the Output window.  Parameters: <ul style="list-style-type: none"> <li>Name: String - the name of the tab to remove text from</li> </ul>	<a href="#">CreateOutputTab</a> <sup>[2857]</sup> <a href="#">EnsureOutput Visible</a> <sup>[2858]</sup> <a href="#">WriteOutput</a> <sup>[2872]</sup>
<b>CloseAddins ()</b>		Called by automation controllers to ensure that Add-Ins created in .NET do not linger after all controller references to Enterprise Architect have been cleared.	
<b>CloseDiagram (long DiagramID)</b>		Closes a diagram in the current list of diagrams that Enterprise Architect has open.  Parameters: <ul style="list-style-type: none"> <li>DiagramID: Long - the ID of the diagram to close</li> </ul>	
<b>CloseFile ()</b>		Closes any open file.	
<b>CreateDocumentGenerator()</b>	<a href="#">DocumentGenerator</a> <sup>[2965]</sup>	Returns a pointer to the <b>EA.DocumentGenerator</b> interface.	
<b>CreateModel (CreateModelType CreateType, string FilePath, long ParentWnd)</b>	<i>Boolean</i>	Creates a new .eap model file based on the standard Enterprise Architect Base model, or a shortcut .eap based on a provided SQL connection.  Returns <b>true</b> when the new file is created, otherwise returns <b>false</b> .  Parameters: <ul style="list-style-type: none"> <li>CreateType: CreateModelType - Specify whether to make a new copy of the EABase.eap model, or create a .eap file shortcut to a DBMS repository; the latter option requires a dialog to be opened for the user to provide SQL connection details</li> <li>FilePath: String - Destination for new .eap file</li> <li>ParentWnd: Long - Window handle to act as the parent for the SQL connection dialog; only required when using <i>cmEAPFromSQLRepository</i></li> </ul>	<a href="#">CreateModelType</a> <sup>[2818]</sup>
<b>CreateOutputTab (string Name)</b>		Creates a tab in the Output window.  Parameters:	<a href="#">CreateOutputTab</a> <sup>[2857]</sup> <a href="#">EnsureOutput Visible</a> <sup>[2858]</sup>

		<ul style="list-style-type: none"> <li>Name: String - the name of the tab to create</li> </ul>	<a href="#">WriteOutput</a> <sup>[2872]</sup>
<b>DeletePerspective (string Perspective, long Options)</b>	<i>Boolean</i>	<b>Deprecated</b> - no longer in use.	
<b>DeleteTechnology (string ID)</b>	<i>Boolean</i>	<p>Removes a specified MDG Technology resource from the repository.</p> <p>Returns <b>true</b>, if the technology is successfully removed from the model. Returns <b>false</b> otherwise.</p> <ul style="list-style-type: none"> <li>This applies to technologies imported into pre-7.0 versions of Enterprise Architect (imported technologies), not to technologies referenced in version 7.0 and later (referenced technologies)</li> </ul> <p>Parameters:</p> <ul style="list-style-type: none"> <li>ID: String - the ID of the technology</li> </ul>	<a href="#">Deploy an MDG Technology</a> <sup>[1581]</sup>
<b>EnsureOutputVisible (string Name)</b>		<p>Ensures that a specified tab in the Output window is visible to the user. The Output window is made visible if it is hidden.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>Name: String - the name of the tab to make visible</li> </ul>	<a href="#">CreateOutputTab</a> <sup>[2857]</sup> <a href="#">EnsureOutput Visible</a> <sup>[2858]</sup> <a href="#">WriteOutput</a> <sup>[2872]</sup>
<b>ExecutePackageBuildScript (long ScriptOptions, string PackageGuid)</b>		<p>Enables you to run the active package build script based on your current selection in the Project Browser. You can also run a script by passing in the package GUID.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>ScriptOptions: Long - the script type; can be any one of these numerical values: <ul style="list-style-type: none"> <li>1 = Build</li> <li>2 = Test</li> <li>3 = Run</li> <li>4 = Create Workbench Instance</li> <li>5 = Debug</li> </ul> </li> <li>PackageGuid: String - the ID of the package for which to run the script</li> </ul>	
<b>Exit</b>		Shuts down Enterprise Architect immediately. Used by .NET programmers where the	

		garbage collector does not immediately release all referenced COM objects.	
<b>GetActivePerspective ()</b>	<i>String</i>	<b>Deprecated</b> - no longer in use.	
<b>GetAttributeByGuid (string Guid)</b>	<a href="#">Attribute</a> [2911]	Returns a pointer to an attribute in the repository, located by its GUID. This is usually found using the <i>AttributeGUID</i> property of an attribute.  Parameters: <ul style="list-style-type: none"><li>Guid: String - the GUID of the attribute to locate</li></ul>	
<b>GetAttributeByID (string Id)</b>	<a href="#">Attribute</a> [2911]	Returns a pointer to an attribute in the repository, located by its ID. This is usually found using the <i>AttributeID</i> property of an attribute.  Parameters: <ul style="list-style-type: none"><li>Id: String - the ID of the attribute to locate</li></ul>	
<b>GetConnectorByGuid (string Guid)</b>	<a href="#">Connector</a> [2933]	Returns a pointer to a connector in the repository, located by its GUID. This is usually found using the <i>ConnectorGUID</i> property of a connector.  Parameters: <ul style="list-style-type: none"><li>Guid: String - the GUID of the connector to locate</li></ul>	
<b>GetConnectorByID (long ConnectorID)</b>	<a href="#">Connector</a> [2933]	Searches the repository for a connector with a specific ID.  Parameters: <ul style="list-style-type: none"><li>ConnectorID: Long - the ID of the connector to locate</li></ul>	
<b>GetContextItem (object Item)</b>	<a href="#">ObjectType</a> [2822]	Sets a pointer to an item in context within Enterprise Architect.  Also returns the corresponding <i>ObjectType</i> .  For additional information about <i>ContextItems</i> and the supported <i>ObjectTypes</i> see the <b>GetContextItemType</b> method (below).  Parameters: <ul style="list-style-type: none"><li>Item: Object - the item to point to</li></ul>	
<b>GetContextItemType ()</b>	<a href="#">ObjectType</a>	Returns the <i>ObjectType</i> of an item in context	

	<a href="#">ype</a> <sup>[2822]</sup>	<p>within Enterprise Architect. A <i>ContextItem</i> is defined as an item selected anywhere within the Enterprise Architect GUI including:</p> <ul style="list-style-type: none"> <li>• An item selected in the Project Browser</li> <li>• An item selected in an open diagram</li> <li>• An item selected in certain dialogs, such as the attribute Properties dialog</li> </ul> <p>The supported <i>ObjectTypes</i> can be any one of the following values:</p> <ul style="list-style-type: none"> <li>• <i>otElement</i></li> <li>• <i>otPackage</i></li> <li>• <i>otDiagram</i></li> <li>• <i>otAttribute</i></li> <li>• <i>otMethod</i></li> <li>• <i>otConnector</i></li> </ul>	
<b>GetCurrentContextObject ()</b>	<i>Object</i>	Returns the current context Object.	
<b>GetCounts ()</b>	<i>String</i>	Returns a set of counts from a number of tables within the base Enterprise Architect repository. These can be used to determine whether records have been added or deleted from the tables for which information is retrieved.	
<b>GetCurrentDiagram ()</b>	<a href="#">Diagram</a> <sup>[2948]</sup>	Returns a selected diagram.	
<b>GetCurrentLoginUser (boolean GetGuid)</b>	<i>String</i>	<p>If security is not enabled in the repository, an error is generated.</p> <p>If <i>GetGuid</i> is <b>True</b>, a GUID generated by Enterprise Architect representing the user is returned; otherwise the text as entered in <i>System Users/User Details/Login</i> is returned.</p>	
<b>GetDiagramByGuid (string Guid)</b>	<a href="#">Diagram</a> <sup>[2948]</sup>	<p>Returns a pointer to a diagram using the global reference ID (global ID). This is usually found using the diagram <i>GUID</i> property of an element, and stored for later use to open an diagram without using the collection <i>GetAt()</i> function.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• Guid: String - the GUID of the diagram to locate</li> </ul>	

<b>GetDiagramByID (long DiagramID)</b>	<a href="#">Diagram</a> [2948]	<p>Gets a pointer to a diagram using an absolute reference number (local ID). This is usually found using the <b>DiagramID</b> property of an element, and stored for later use to open a diagram without using the collection <b>GetAt()</b> function.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>DiagramID: Long - the ID of the diagram to locate</li> </ul>	
<b>GetElementByGuid (string Guid)</b>	<a href="#">Element</a> [2887]	<p>Returns a pointer to an element in the repository, using the element's GUID reference number (global ID). This is usually found using the <b>ElementGUID</b> property of an element, and stored for later use to open an element without using the collection <b>GetAt ()</b> function.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>Guid: String - the GUID of the element to locate</li> </ul>	
<b>GetElementByID (long ElementID)</b>	<a href="#">Element</a> [2887]	<p>Gets a pointer to an element using an absolute reference number (local ID). This is usually found using the <b>ElementID</b> property of an element, and stored for later use to open an element without using the collection <b>GetAt ()</b> function.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>ElementID: Long - the ID of the element to locate</li> </ul>	
<b>GetElementsByQuery (string QueryName, string SearchTerm)</b>	<a href="#">Collection</a> [2829] (of type <i>Element</i> )	<p>Enables the user to run a search in Enterprise Architect, returning the result as a collection.</p> <p>For example <i>GetElementsByQuery ('Simple','Class1')</i>, where results contain elements with <i>Class1</i> in the <b>Name</b> and <b>Notes</b> fields.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>QueryName: String - the name of the search to run, for example 'Simple'</li> <li>SearchTerm: String - the term to search for</li> </ul>	
<b>GetElementSet (string IDList, long Options)</b>	<a href="#">Collection</a> [2829] (of type <i>Element</i> )	<p>Returns a set of elements as a collection based on a comma-separated list of <b>ElementID</b> values. By default, if no values are provided in the IDList parameter, all objects for the entire project are returned.</p> <p>Parameters</p>	

		<ul style="list-style-type: none"> <li>• IDList: String - a comma-separated list of <i>ElementID</i> values</li> <li>• Options: Long - modifies default behaviour of this method <ul style="list-style-type: none"> <li>• 1 - Returns empty collection when empty IDList parameter is given</li> <li>• 2 - Use IDList string as an SQL query to populate this collection</li> </ul> </li> </ul>	
<b>GetFieldFromFormat (string Format, string Text)</b>	<i>String</i>	<p>Converts a field from your preferred format to Enterprise Architect's internal format; returns the field in that format.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• Format: String - The format to convert the field from; valid formats are: <ul style="list-style-type: none"> <li>• HTML - Full HTML</li> <li>• RTF - Rich Text Format</li> <li>• TXT - Plain text</li> </ul> </li> <li>• Text: String - The field to be converted</li> </ul>	
<b>GetFormatFromField (string Format, string Text)</b>	<i>String</i>	<p>After accessing a field that contains formatting, use this method to convert it to your preferred format; returns the field in the format specified.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• Format: String - The format to convert the field to; valid formats are: <ul style="list-style-type: none"> <li>• HTML - Full HTML</li> <li>• RTF - Rich Text Format</li> <li>• TXT - Plain text</li> </ul> </li> <li>• Text: String - The field to be converted</li> </ul>	
<b>GetGapAnalysisMatrix ()</b>	<i>String</i>	<p>Read Only.</p> <p>Returns all Gap Analyses as an XML document.</p>	<a href="#">Gap Analysis Matrix</a> [745]
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.	
<b>GetMailInterface()</b>	<i>MailInterface</i>	Returns an instance of the <b>EA.MailInterface</b> ; use this interface to automate the process of creating and sending model mail messages.	<a href="#">MailInterface Package</a> [2990]

<b>GetMethodByGuid (string Guid)</b>	<a href="#">Method</a> [2919]	<p>Returns a pointer to a method in the repository; this is usually found using the <i>MethodGUID</i> property of a method.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>Guid: String - the GUID of the method to look for</li> </ul>	
<b>GetMethodById (string Id)</b>	<a href="#">Method</a> [2919]	<p>Returns a pointer to a method in the repository; this is usually found using the <i>MethodID</i> property of a method.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>Id: String - the ID of the method to look for</li> </ul>	
<b>GetPackageByGuid (string Guid)</b>	<a href="#">Package</a> [2835]	<p>Returns a pointer to a package in the repository using the package's GUID reference number (global ID). This is usually found using the <i>PackageGUID</i> property of the package.</p> <p>Each package in the model also has an associated element with the same GUID, so if you have an element with <i>Type="Package"</i> then you can load the package by calling:</p> <p style="text-align: center;"><i>GetPackageByGuid(ElementGUID)</i></p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>Guid: String - the GUID of the package to look for</li> </ul>	
<b>GetPackageById (long PackageID)</b>	<a href="#">Package</a> [2835]	<p>Get a pointer to a package using an absolute reference number (local ID). This is usually found using the <i>PackageID</i> property of an package, and stored for later use to open a package without using the collection <b>GetAt ()</b> function.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>PackageID: Long - the ID of the package to locate</li> </ul>	
<b>GetProjectInterface ()</b>	<a href="#">Project</a> [2962]	<p>Return a pointer to the <b>EA.Project</b> interface (the XML-based automation server for Enterprise Architect). Use this interface to work with Enterprise Architect using XML, and also to access utility functions for loading diagrams, running reports and so on.</p>	<a href="#">EA Project interface</a> [2962]
<b>GetReferenceList (string Type)</b>	<a href="#">Reference</a> [2848]	<p>Uses the list type to get a pointer to a <b>Reference List</b> object.</p> <p>Parameters:</p>	

		<ul style="list-style-type: none"> <li>Type: String - specifies the list type to get; valid list types are:   <i>Diagram</i>  <i>Element</i>  <i>Constraint</i>  <i>Requirement</i>  <i>Connector</i>  <i>Status</i>  <i>Cardinality</i>  <i>Effort</i>  <i>Metric</i>  <i>Scenario</i>  <i>Status and</i>  <i>Test</i> </li> </ul>	
<b>GetRelationshipMatrix()</b>	<i>String</i>	Returns all Relationship Matrices as an XML document.	<a href="#">Relationship Matrix</a> [727]
<b>GetTechnologyVersion (string ID)</b>	<i>String</i>	Returns the version of a specified MDG Technology resource.  Parameters: <ul style="list-style-type: none"> <li>ID: String - the specified technology ID</li> </ul>	
<b>GetTreeSelectedElements()</b>	<a href="#">Collection</a> [2825]	Returns the set of elements currently selected in the Project Browser as a collection.	
<b>GetTreeSelectedItem (object SelectedItem)</b>	<a href="#">Object</a> <a href="#">Type</a> [2822]	Gets an object variable and type corresponding to the currently selected item in the tree view.  To use this function, create a generic object variable and pass this as the parameter. Depending on the return type, cast it to a more specific type.  The object passed back through the parameter can be a package, element, diagram, attribute or operation object.  Parameters: <ul style="list-style-type: none"> <li>SelectedItem: Object - the object to get the variable and type for</li> </ul>	
<b>GetTreeSelectedItemType ()</b>	<a href="#">Object</a> <a href="#">Type</a> [2822]	Returns the type of the object currently selected in the tree. One of: <ul style="list-style-type: none"> <li><i>otDiagram</i></li> <li><i>otElement</i></li> <li><i>otPackage</i></li> <li><i>otAttribute</i></li> </ul>	






		<ul style="list-style-type: none"> <li><i>otMethod</i></li> </ul>	
<b>GetTreeSelectedObject ()</b>	<i>Object</i>	The related method <b>GetTreeSelectedItem ()</b> has an output parameter that is inaccessible by some scripting languages. As an alternative, this method provides the selected item through the return value.	<a href="#">GetTreeSelectedItem</a> <sup>[2864]</sup>
<b>GetTreeSelectedPackage ()</b>	<a href="#">Package</a> <sup>[2835]</sup>	Returns the package in which the currently selected tree view object is contained.	
<b>HasPerspective (string Perspective)</b>	<i>String</i>	<b>Deprecated</b> - no longer in use.	
<b>HideAddinWindow ()</b>		Hides the docked Add-In window.	<a href="#">AddWindow</a> <sup>[2855]</sup> <a href="#">Custom Docked Window</a> <sup>[3096]</sup> <a href="#">RemoveWindow</a> <sup>[2869]</sup>
<b>ImportPackageBuildScripts (string PackageGuid, string BuildScriptXML)</b>		Imports build scripts into a package in Enterprise Architect.  Parameters: <ul style="list-style-type: none"> <li>PackageGuid: String - the GUID of the package into which to import the build scripts</li> <li>BuildScriptXML: String - the build script XML data, which you can export from within Enterprise Architect</li> </ul>	
<b>ImportTechnology (string Technology)</b>	<i>Boolean</i>	Installs a given MDG Technology resource into the repository.  Returns <b>True</b> , if the technology is successfully loaded into the model. Otherwise returns <b>False</b> .  This applies to technologies imported into pre-7.0 versions of Enterprise Architect (imported technologies), not to technologies referenced in version 7.0 and later (referenced technologies).  Parameters: <ul style="list-style-type: none"> <li>Technology: String - the contents of the technology resource file</li> </ul>	<a href="#">Deploy an MDG Technology</a> <sup>[1581]</sup>
<b>InvokeConstructPicker (string ElementFilter)</b>	<i>String</i>	Invokes the Select <Item> dialog with filters on the object type and, optionally, stereotype. Returns the <i>ElementID</i> of the selected object, or <b>0</b> if no object was selected when the	<a href="#">Select &lt;Item&gt; Dialog</a> <sup>[994]</sup>




		<p>dialog was closed.</p> <p>For example:</p> <pre>elementid=Repository. InvokeConstructPicker ("IncludedTypes=Class,Component; StereoType=foo,bar")</pre> <p>In this example, the Select &lt;item&gt; dialog will allow the user to select any Class or Component element in the model that has a stereotype of <i>foo</i> or <i>bar</i>.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• ElementFilter: String - specifies which elements or packages are to be made available for selection, based on element types and stereotypes identified by the <b>IncludedTypes</b> and <b>StereoType</b> filters <ul style="list-style-type: none"> <li>• IncludedTypes - (mandatory) comma separated list of element types that can be selected in this dialog; example: Package, Class, Component</li> <li>• StereoType - (optional) comma separated list of stereotypes that can be selected in this dialog</li> </ul> </li> <li>• The IncludedTypes and StereoType filters are separated by a semi-colon</li> <li>• Do not use leading or trailing spaces between element type or stereotype values</li> <li>• StereoType and IncludedTypes must be written with the correct case; element type names are also case sensitive</li> </ul>	
<b>InvokeFileDialog (</b> <b>string FilterString,</b> <b>long Filterindex,</b> <b>long Flags)</b>	String	<p>Opens a standard Open File Dialog and returns a string containing the full path to the selected file on success. Returns an empty string if the dialog was canceled.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• FilterString: String - list of file type filters.</li> <li>• Filterindex: Long - One-based index of the filter to be used by default</li> <li>• Flags: Long - additional bit flags used to initialize the file dialog. See OPENFILENAME structure in MSDN documentation for accepted values.</li> </ul>	<a href="#">Project.</a> <a href="#">GetFileNameDialog()</a> <small>[2974]</small>

<b>IsTabOpen (string TabName)</b>	<i>String</i>	<p>Checks whether a named Enterprise Architect tabbed view is open and active. This includes open diagram windows or custom controls added using <b>Repository.AddTab ()</b>.</p> <p>Returns:</p> <ul style="list-style-type: none"> <li>• <b>2</b> to indicate that a tab is open and active (top-most)</li> <li>• <b>1</b> to indicate that it is open but not top-most, or</li> <li>• <b>0</b> to indicate that it is not visible at all</li> </ul> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• TabName: String - the name of the tab to check for; TabName is case sensitive</li> </ul>	<a href="#">Repository.AddTab()</a> <sup>[2855]</sup>
<b>IsTechnologyEnabled (string ID)</b>	<i>Boolean</i>	<p>Checks whether a specified technology is enabled in Enterprise Architect.</p> <p>Returns <b>True</b> if the MDG Technology resource is enabled. Otherwise returns <b>False</b>.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• ID: String - the technology ID to check for</li> </ul>	
<b>IsTechnologyLoaded (string ID)</b>	<i>Boolean</i>	<p>Checks whether a specified technology is loaded into the repository.</p> <p>Returns <b>True</b> if the MDG Technology resource is loaded into the repository. Otherwise returns <b>False</b>.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• ID: String - the technology ID to check for</li> </ul>	
<b>LoadAddins ()</b>		Loads all Add-Ins from a repository.	
<b>OpenDiagram (long DiagramID)</b>		<p>Provides a method for an automation client or Add-In to open a diagram. The diagram is added to the tabbed list of open diagrams in the main Enterprise Architect view.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• DiagramID: Long - the ID of the diagram to open</li> </ul>	
<b>OpenFile (string Filename)</b>	<i>Boolean</i>	This is the main point for opening an Enterprise Architect project file from an automation client, and working with the contained objects.	<a href="#">Open a Project</a> <sup>[202]</sup>

		<p>If the required project is a DBMS repository, and you have created a shortcut .EAP file containing the database connection string, you can call this shortcut file to access the DBMS repository.</p> <p>You can also connect to a SQL database by passing in the connection string itself instead of a filename. A valid connection string can be obtained from the <b>Open Project</b> dialog by selecting a recently opened SQL repository.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>Filename: String - the filename of the Enterprise Architect project to open</li> </ul>	
<b>OpenFile2 (string FilePath, string Username, string Password)</b>	<i>Boolean</i>	<p>As for <b>OpenFile ()</b> except this enables the specification of a password.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>Filepath: String - the file path of the Enterprise Architect project to open</li> <li>Username: String - the user login ID</li> <li>Password: String - the user password</li> </ul>	
<b>RefreshModelView (long PackageID)</b>		<p>Reloads a package or the entire model, updating the user interface.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>PackageID: Long - the ID of the package to reload: if <b>0</b>, the entire model is reloaded; if a valid package ID, only that package is reloaded</li> </ul>	
<b>RefreshOpenDiagrams (boolean FullReload)</b>		<p>Reloads the diagram contents for all open diagrams from the repository.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>FullReload: Boolean - if <b>false</b> only the contents of element compartments are reloaded; if <b>true</b> the full content of each diagram is reloaded</li> </ul>	
<b>ReloadDiagram (long DiagramID)</b>		<p>Reloads a specified diagram. This would commonly be used to refresh a visible diagram after code import/export or other batch process where the diagram requires complete refreshing.</p> <ul style="list-style-type: none"> <li>Calling this method within a call to <b>EA_OnNotifyContextItemModified</b> is not supported</li> </ul> <p>Parameters:</p> <ul style="list-style-type: none"> <li>DiagramID: Long - the ID of the</li> </ul>	<a href="#">EA_OnNotifyContextItemModified</a> <small>[3039]</small>

		diagram to be reloaded	
<b>RemoveOutputTab (string Name)</b>		Removes a specified tab from the Output window.  Parameters: <ul style="list-style-type: none"> <li>Name: String - the name of the tab to be removed</li> </ul>	
<b>RemoveWindow (string WindowName)</b>	<i>Boolean</i>	Removes an Add-In window that matches the specified WindowName.  Parameters: <ul style="list-style-type: none"> <li>WindowName: String - the name of the window to remove</li> </ul>	<a href="#">AddWindow</a> 
<b>RepositoryType ()</b>	<i>String</i>	Returns the currently open database/repository type.  May return one of the following values: <ul style="list-style-type: none"> <li>JET ( .EAP file, MS Access 97 to 2003 format )</li> <li>ACCESS2007 ( .accdb file, MS Access 2007+ format )</li> <li>ASA ( Sybase SQL Anywhere )</li> <li>SQLSVR ( Microsoft SQL Server )</li> <li>MYSQL ( MySQL )</li> <li>ORACLE ( Oracle )</li> <li>POSTGRES ( PostgreSQL )</li> </ul>	
<b>RunModelSearch (string sQueryName, string sSearchTerm, string sSearchOptions, string sSearchData)</b>		Runs a search, displaying the results in Enterprise Architect's Model Search window.  Parameters: <ul style="list-style-type: none"> <li>sQueryName: String - the name of the search to run, for example <i>Simple</i></li> <li>sSearchTerm: String - the term to search for</li> <li>sSearchOptions: String - currently not being used</li> <li>sSearchData: String - enables you to supply a list of results in the form of XML, which is appended onto the result list in Enterprise Architect - see <b>XML Format</b>; this parameter is not mandatory so pass in an empty string to run the search as per normal</li> </ul>	<a href="#">XML Format</a>  <a href="#">Project Class</a>  (a different call to Run Model Search)
<b>SaveAllDiagrams ()</b>		Saves all open diagrams.	

<b>SaveAuditLogs (</b> <b>string FilePath,</b> <b>object StartDateTime,</b> <b>object EndDateTime)</b>	<i>Boolean</i>	<p>Saves the Audit Logs contained within a model to a specified file.</p> <p>If <i>StartDateTime</i> and <i>EndDateTime</i> are not null then only log items that fall into this period are saved.</p> <p>Returns <b>true</b> for success, <b>false</b> for failure.</p> <ul style="list-style-type: none"> <li>This might fail if the user logged into the model does not have the correct access permission</li> </ul> <p>Parameters:</p> <ul style="list-style-type: none"> <li>FilePath: String - the file to save the Audit Logs to</li> <li>StartDateTime: Variant ( DateTime ) - the earliest date and time of log entries to save</li> <li>EndDateTime; Variant ( DateTime ) - the latest date and time of log entries to save</li> </ul>	
<b>SaveDiagram (</b> <b>long DiagramID)</b>		<p>Saves an open diagram; assumes the diagram is open in the main user interface Tab list.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>DiagramID: Long - the ID of the diagram to save</li> </ul>	
<b>ScanXMIAndReconcile</b> <b>()</b>		<p>Scans the package XMI files associated with each of the project's controlled packages and restores any diagram objects or cross-references that are detected as missing from the project.</p> <p>This function is useful in team environments where each user maintains their own private copy of the model database (i.e. multiple private EAP files) and model updates are propagated through the use of controlled packages; it provides no benefit when the model is hosted in a single shared database that is accessed by all team members.</p> <p>Each controlled package is compared with its associated XMI file and, if the cross-reference information in the model does not match the XMI, Enterprise Architect updates the model with the information from the XMI and records the update in the Output window.</p> <p>You can roll back such updates by right-clicking on the entry in the Output window and selecting the context menu option <b>Rollback Update</b> (or <b>Rollback Selected Updates</b> if multiple entries are selected).</p>	

		<p>Closing the model clears the entries in the Output window; an entry in the Output window is also cleared as and when you roll-back the update for it.</p> <p>This functionality is invoked automatically as part of the <b>Get All Latest</b> operation.</p> <p>When working in an environment that uses a Private Model deployment and your model contains a significant number of cross-package references, it is recommended that you invoke this function from time to time, following the re-importation of controlled packages - for example, after using <b>Get Latest</b> to update a number of packages, or after performing a number of package check-outs.</p> <ul style="list-style-type: none"> <li>As a general rule, avoid running this function while you have uncommitted changes in your model</li> <li>Generally, you should: <ul style="list-style-type: none"> <li>Check-out a number of packages</li> <li>Invoke <b>ScanXMIAndReconcile</b></li> <li>Make your modifications</li> <li>Commit any outstanding changes before you check-out more packages and run <b>ScanXMIAndReconcile</b> again</li> </ul> </li> </ul>	
<b>ShowAddinWindow (string TabName)</b>	<i>Boolean</i>	<p>Shows the docked Add-In window on the specified page. Returns <b>true</b> if a tab of the specified name is now displayed.</p> <p>Parameters</p> <ul style="list-style-type: none"> <li>TabName: String - specifies the tab page</li> </ul>	<a href="#">AddWindow</a>  <sup>[2855]</sup> <a href="#">Custom Docked Window</a>  <sup>[3098]</sup> <a href="#">RemoveWindow</a>  <sup>[2869]</sup>
<b>ShowDynamicHelp (string Topic)</b>		<p>Shows a help topic as a view.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>Topic: String - specifies the help topic</li> </ul>	
<b>ShowInProjectView (object Item)</b>		<p>Selects a specified object in the Project Browser.</p> <p>Accepted object types are <i>Package</i>, <i>Element</i>, <i>Diagram</i>, <i>Attribute</i>, and <i>Method</i>; an exception is thrown if the object is of an invalid type.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>Item: Object - the object to highlight</li> </ul>	
<b>ShowWindow (long Show)</b>		Shows or hides Enterprise Architect.	

		Parameters: <ul style="list-style-type: none"> <li>Show: Long</li> </ul>	
<b>SQLQuery (string SQL)</b>	<i>String</i>	Enables execution of a SQL <i>select</i> statement against the current repository.  Returns an XML formatted string value of the resulting recordset.  Parameters: <ul style="list-style-type: none"> <li>SQL: String - contains the SQL Select statement</li> </ul>	
<b>SynchProfile (string Profile, string Stereotype)</b>	<i>Boolean</i>	Synchronizes Tagged Values and constraints of a UML Profile item using the Synch Profile Elements dialog.  Parameters: <ul style="list-style-type: none"> <li>Profile: String - the name of the profile that contains the stereotype</li> <li>Stereotype: String - the name of the profile stereotype for which the default tags and constraints are to be synchronized</li> </ul>	<a href="#">Synchronize Tagged Values and Constraints</a> <sup>[1473]</sup>
<b>VersionControlResynch PkgStatuses (boolean ClearSettings)</b>		Synchronizes the version control status of each version controlled package within the current model with the status reported by your version control provider.  Parameters: <ul style="list-style-type: none"> <li>ClearSettings: Boolean <ul style="list-style-type: none"> <li>- if <b>true</b>, clear the version control settings from packages that are reported by the version control provider as <i>uncontrolled</i></li> <li>- if <b>false</b>, leave the version control settings unchanged for packages reported as <i>uncontrolled</i></li> </ul> </li> </ul>	<a href="#">Resynchronize the Status of Version Controlled Packages</a> <sup>[444]</sup>
<b>WriteOutput (string Name, string Output, long ID)</b>		Writes text to a specified tab in the Output window, and associates the text with an ID.  Parameters: <ul style="list-style-type: none"> <li>Name: String - specifies the tab on which to display the text</li> <li>Output: String - specifies the text to display</li> <li>ID: Long - specifies a numeric ID value to associate with this output item for further handling by Add-Ins; can be set to <b>0</b> if no handling is required</li> </ul>	<a href="#">ClearOutput</a> <sup>[2857]</sup> <a href="#">CreateOutputTab</a> <sup>[2857]</sup> <a href="#">EnsureOutput Visible</a> <sup>[2858]</sup> <a href="#">EA_OnOutputItemClicked</a> <sup>[3027]</sup> <a href="#">EA_OnOutputItemDoubleClicked</a> <sup>[3028]</sup>




--	--	--	--

#### 20.2.2.4.15 Stereotype Class

The **Stereotype** element corresponds to a UML stereotype, which is an extension mechanism for varying the behavior and type of a model element. Use the Repository **Stereotypes** collection to add new elements and delete existing ones.

**Associated table in .EAP file** *t\_stereotypes*

#### Stereotype Attributes

Attribute	Type	Notes
<b>AppliesTo</b>	<i>String</i>	Read/Write A reference to the stereotype <i>Base Class</i> , that is, which element it applies to.
<b>MetafileLoadPath</b>	<i>String</i>	Read/Write The path to an associated metafile. The automation interface does not yet support loading metafiles. To do this you must use the Stereotype tab of the UML Types dialog in Enterprise Architect.
<b>Notes</b>	<i>String</i>	Read/Write. Notes about the stereotype.
<b>Name</b>	<i>String</i>	Read/Write The stereotype name, which appears in the <b>Stereotype</b> drop list for elements that match the <b>AppliesTo</b> attribute.
<b>ObjectType</b>	<a href="#">Object Type</a> 	Read only Distinguishes objects referenced through a Dispatch interface.
<b>StereotypeGUID</b>	<i>String</i>	Read/Write A unique identifier for stereotype, generally set and maintained by Enterprise Architect.
<b>Style</b>	<i>String</i>	Read/Write An additional style specifier for the stereotype.
<b>VisualType</b>	<i>String</i>	Read/Write Indicates an inbuilt visual style associated with a stereotype.  <b>Not currently implemented.</b>

Attribute	Type	Notes

#### Stereotype Methods

Method	Type	Notes
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.
<b>Update ()</b>	<i>Boolean</i>	Updates the current stereotype object after modification or appending a new item.  If <b>false</b> is returned, check the <b>GetLastError ()</b> function for more information.

#### 20.2.2.4.16 Task Class

A **Task** is an entry in the System ToDo list. Tasks can be accessed using the Repository **Tasks** collection.

Associated table in .EAP file    *t\_tasks*

#### Task Attributes

Attribute	Type	Notes
<b>ActualTime</b>	<i>Long</i>	Read/Write The time already expended on the task, in hours, days or other units.
<b>AssignedTo</b>	<i>String</i>	Read/Write The person this task is assigned to; that is, the responsible resource.
<b>EndDate</b>	<i>Date</i>	Read/Write The date the task is scheduled to finish.
<b>History</b>	<i>String</i>	Read/Write A memo field to hold, for example, task history or notes.
<b>Name</b>	<i>Variant</i>	Read/Write The task name.
<b>Notes</b>	<i>Variant</i>	Read/Write

Attribute	Type	Notes
		A description of the task.
<b>ObjectType</b>	<a href="#">ObjectTyp</a> <small>e[2822]</small>	Read only Distinguishes objects referenced through a Dispatch interface.
<b>Owner</b>	<i>String</i>	Read/Write The task owner.
<b>Percent</b>	<i>Long</i>	Read/Write The percentage completion of the task.
<b>Phase</b>	<i>String</i>	Read/Write The phase of the project the task relates to.
<b>Priority</b>	<i>String</i>	Read/Write The priority of this task.
<b>StartDate</b>	<i>Date</i>	Read/Write The date the task is to start.
<b>Status</b>	<i>Variant</i>	Read/Write The current status of the task.
<b>TaskID</b>	<i>Long</i>	Read only The local ID of the task.
<b>TotalTime</b>	<i>Long</i>	Read/Write The total expected time the task might run, in hours, days or some other unit.
<b>Type</b>	<i>String</i>	Read/Write Sets or returns a string representing the type.

#### Task Methods


Method	Type	Notes
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.
<b>Update ()</b>	<i>Boolean</i>	Updates the current Task object after modification or appending a new item. If <b>false</b> is returned, check the <b>GetLastError ()</b> function for more information.

#### 20.2.2.4.17 Term Class

A **Term object** represents one entry in the system glossary. Terms can be accessed using the Repository *Terms* collection.

Associated table in .EAP file    *t\_glossary*

##### Term Attributes

Attribute	Type	Notes
<b>Meaning</b>	<i>String</i>	Read/Write The description of the term; its meaning.
<b>ObjectType</b>	<a href="#">ObjectType</a> 	Read only Distinguishes objects referenced through a Dispatch interface.
<b>Term</b>	<i>String</i>	Read/Write The glossary item name.
<b>TermID</b>	<i>Long</i>	Read only A local ID number to identify the term in the model.
<b>Type</b>	<i>String</i>	Read/Write The type this term applies to (for example, business or technical).

##### Term Methods

Method	Type	Notes
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.

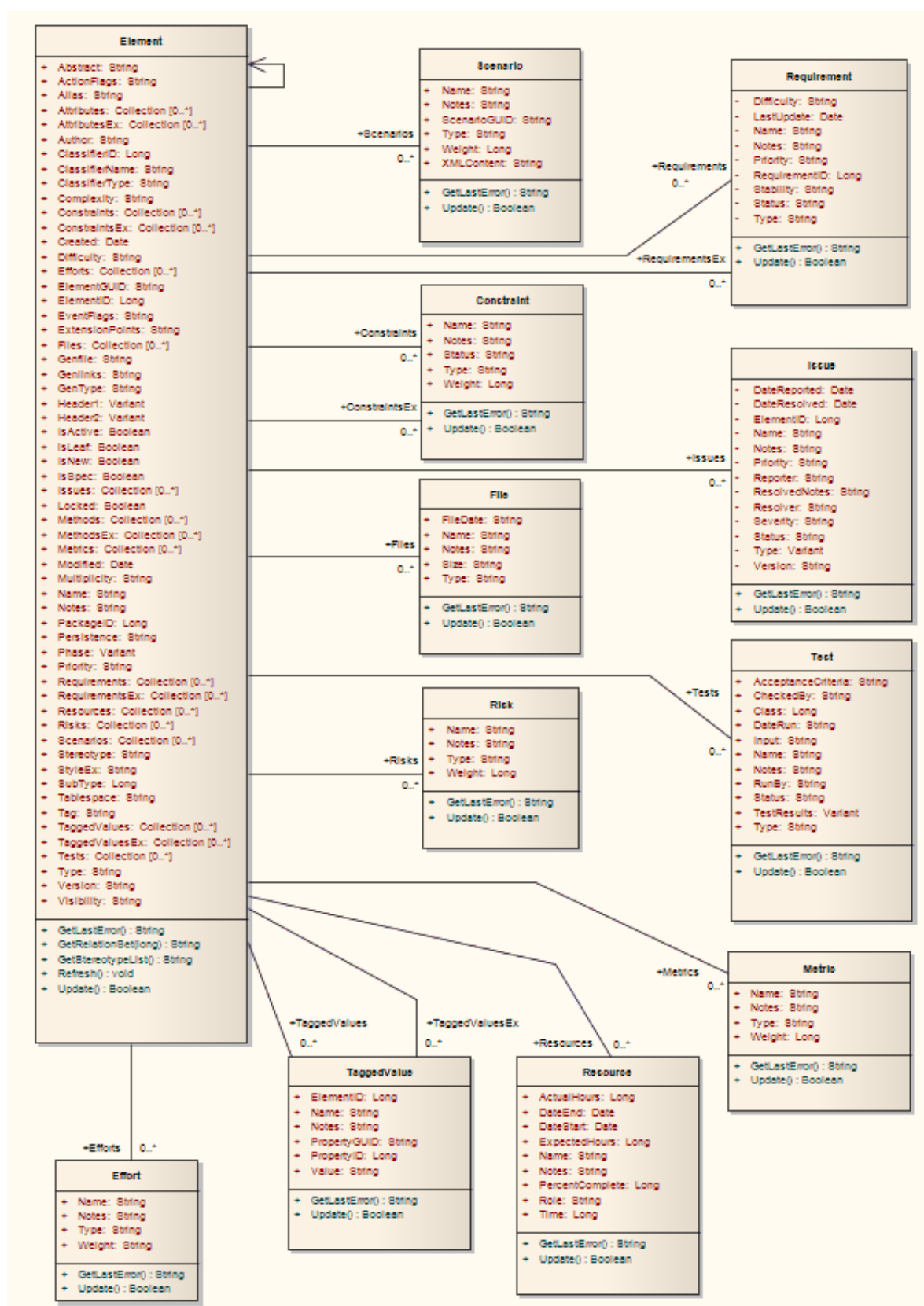
Method	Type	Notes
<b>Update ()</b>	<i>Boolean</i>	Updates the current Term object after modification or appending a new item. If <b>false</b> is returned, check the <b>GetLastError ()</b> function for more information.

#### 20.2.2.5 Element Package

The **Element Package** contains information about an element and its associated extended properties such as testing and project management information. An element is the basic item in an Enterprise Architect model. Classes, Use Cases and Components are all different types of UML element.

The diagram below illustrates the relationships between an *element* and its associated extended information. The related information is accessed through the collections owned by the element (for example, Scenarios and Tests). It also includes a full description of the element object (the basic model structural unit).

#### Example



### 20.2.2.5.1 Constraint Class

A **Constraint** is a condition imposed on an element. Constraints are accessed through the Element **Constraints** collection.

**Associated table in .EAP file** *t\_objectconstraints*

#### Constraint Attributes

Attribute	Type	Notes
<b>Name</b>	<i>String</i>	Read/Write The name of the constraint (that is, the constraint).
<b>Notes</b>	<i>String</i>	Read/Write Notes about the constraint.
<b>ObjectType</b>	<a href="#">ObjectTyp</a> e <sup>[2822]</sup>	Read only Distinguishes objects referenced through a Dispatch interface.
<b>ParentID</b>	<i>Long</i>	Read only The <b>ElementID</b> of the element to which this constraint applies.
<b>Status</b>	<i>String</i>	Read/Write The current status of the constraint.
<b>Type</b>	<i>String</i>	Read/Write The constraint type.
<b>Weight</b>	<i>Long</i>	Read/Write A weighting factor.

#### Constraint Methods

Method	Type	Notes
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.
<b>Update ()</b>	<i>Boolean</i>	Update the current <i>Constraint</i> object after modification or appending a new

Method	Type	Notes
		<p>item.</p> <p>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.</p>

#### 20.2.2.5.2 Effort Class

An **Effort** is a named item with a weighting that can be associated with an element for purposes of building metrics about the model. Efforts are accessed through the Element **Efforts** collection.

Associated table in .EAP file    *t\_objecteffort*

##### Effort Attributes

Attribute	Type	Notes
<b>Name</b>	<i>String</i>	<p>Read/Write</p> <p>The name of the effort.</p>
<b>Notes</b>	<i>String</i>	<p>Read/Write</p> <p>Notes about the effort.</p>
<b>ObjectType</b>	<a href="#">ObjectTyp</a> <small>e<sub>2822</sub></small>	<p>Read only</p> <p>Distinguishes objects referenced through a Dispatch interface.</p>
<b>Type</b>	<i>String</i>	<p>Read/Write</p> <p>The effort type.</p>
<b>Weight</b>	<i>Long</i>	<p>Read/Write</p> <p>A weighting factor.</p>
<b>Weight2</b>	<i>Float</i>	<p>Read/Write</p> <p>A weighting factor.</p>

##### Effort Methods

Method	Type	Notes
<b>GetLastError() ( )</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.



Method	Type	Notes
<b>Update ()</b>	<i>Boolean</i>	Update the current <b>Effort</b> object after modification or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.

### 20.2.2.5.3 Element Class

An **Element** is the main modeling unit, corresponding to (for example) a Class, Use Case, Node or Component. You create new elements by adding to the Package *Elements* collection. Once you have created an element, you can add it to the *DiagramObjectClass* of a diagram to include it in the diagram.

Elements also have a collection of connectors. Each entry in this collection indicates a relationship to another element.

There are also some extended collections for managing additional information about the element, including properties such as Tagged Values, Issues, Constraints and Requirements.





**Associated table in .EAP file** *t\_object*

### Element Attributes

Attribute	Type	Notes	See also
<b>Abstract</b>	<i>String</i>	Read/Write Indicates if the element is <b>Abstract (1)</b> or <b>Concrete (0)</b> .	
<b>ActionFlags</b>	<i>String</i>	Read/Write A structure to hold flags concerned with Action semantics.	
<b>Alias</b>	<i>String</i>	Read/Write An optional alias for this element.	
<b>AssociationClassConnectorID</b>	<i>Long</i>	Read only If the element is an <i>AssociationClass</i> , <i>AssociationClassConnectorID</i> contains the Connector ID of the respective Association connector.	<a href="#">CreateAssociationClass()</a> <sup>[2892]</sup> (Information on creating <i>AssociationClasses</i> from elements)
<b>Attributes</b>	<i>Collection</i>	Read only A collection of <b>Attribute</b> objects for the current	<a href="#">Attribute Class</a> <sup>[2911]</sup> <a href="#">Collection Class</a>

Attribute	Type	Notes	See also
		element; use the <b>AddNew</b> and <b>Delete</b> functions to manage attributes.	<a href="#">[2829]</a>
<b>AttributesEx</b>	<i>Collection</i>	Read only A collection of <b>Attribute</b> objects belonging to the current element and its parent elements.	<a href="#">Attribute Class</a> <a href="#">[2911]</a> <a href="#">Collection Class</a> <a href="#">[2829]</a>
<b>Author</b>	<i>String</i>	Read/Write The element author.	<a href="#">Repository: Authors</a> <a href="#">[2850]</a>
<b>BaseClasses</b>	<i>Collection</i>	Read only A list of Base Classes for this element, presented as a collection for convenience.	<a href="#">Collection Class</a> <a href="#">[2829]</a>
<b>ClassifierID</b>	<i>Long</i>	<b>Deprecated</b> See <i>ClassifierID</i> .	
<b>ClassifierID</b>	<i>Long</i>	Read/Write The <b>ElementID</b> of a Classifier associated with this element; that is, the base type. Only valid for instance type elements (such as Object or Sequence).	
<b>ClassifierName</b>	<i>String</i>	Read/Write Name of associated Classifier (if any).	
<b>ClassifierType</b>	<i>String</i>	Read only Type of associated Classifier.	
<b>Complexity</b>	<i>String</i>	Read/Write A complexity value indicating how complex the element is; used for metric reporting and estimation. Valid values are: <b>1</b> for Easy, <b>2</b> for Medium, <b>3</b> for Difficult.	
<b>CompositeDiagram</b>	<i>Diagram</i>	Read only If the element is Composite, returns its associated diagram; otherwise returns null.	<a href="#">Diagram Class</a> <a href="#">[2948]</a>

Attribute	Type	Notes	See also
<b>Connectors</b>	<i>Collection</i>	Read only Returns a collection containing the connectors to other elements.	<a href="#">Connector Class</a> <sup>[2933]</sup> <a href="#">Collection Class</a> <sup>[2829]</sup>
<b>Constraints</b>	<i>Collection</i>	Read only A collection of Constraint objects.	<a href="#">Constraint Class</a> <sup>[2879]</sup> <a href="#">Collection Class</a> <sup>[2829]</sup>
<b>ConstraintsEx</b>	<i>Collection</i>	Read only Collection of Constraint objects belonging to the current element and its parent elements.	<a href="#">Constraint Class</a> <sup>[2879]</sup> <a href="#">Collection Class</a> <sup>[2829]</sup>
<b>Created</b>	<i>Date</i>	Read/Write The date the element was created.	
<b>CustomProperties</b>	<i>Collection</i>	Read only List of advanced properties for an element. The collection of advanced properties differs depending on element type; for example, an Action and an Activity have different advanced properties. Currently only editable from the user interface.	<a href="#">Custom Properties Collection</a> <sup>[2918]</sup>
<b>Diagrams</b>	<i>Collection</i>	Read only Returns a collection of sub-diagrams (child diagrams) attached to this element as seen in the tree view.	<a href="#">Diagram Class</a> <sup>[2948]</sup> <a href="#">Collection Class</a> <sup>[2829]</sup>
<b>Difficulty</b>	<i>String</i>	Read/Write A difficulty level associated with this element for estimation/metrics; only useable for Requirement, Change and Issue element types, otherwise ignored. Valid values are: <b>Low</b> , <b>Medium</b> , <b>High</b> .	
<b>Efforts</b>	<i>Collection</i>	Read only A collection of Effort objects.	<a href="#">Effort Class</a> <sup>[2880]</sup> <a href="#">Collection Class</a> <sup>[2829]</sup>
<b>ElementGUID</b>	<i>String</i>	Read only A globally unique ID for this element; that is, unique across all model files.	

Attribute	Type	Notes	See also
<b>ElementID</b>	<i>Long</i>	Read only The local ID of the Element; valid for this file only.	
<b>Elements</b>	<i>Collection</i>	Read only Returns a collection of child elements (sub-elements) attached to this element as seen in the tree view.	<a href="#">Collection Class</a> 
<b>EmbeddedElements</b>	<i>Collection</i>	Read only A list of elements that are embedded into this element, such as Ports, Parts, Pins and Parameter Sets.	<a href="#">EmbeddedElements Collection</a> 
<b>EventFlags</b>	<i>String</i>	Read/Write A structure to hold a variety of flags to do with signals or events.	
<b>ExtensionPoints</b>	<i>String</i>	Read/Write Optional extension points for a Use Case as a comma-separated list.	
<b>Files</b>	<i>Collection</i>	Read only A collection of <b>File</b> objects.	<a href="#">File Class</a>  <a href="#">Collection Class</a> 
<b>GenFile</b>	<i>String</i>	Read/Write The file associated with this element for code generation and synchronization purposes; can include macro expansion tags for local conversion to full path.	
<b>Genlinks</b>	<i>String</i>	Read/Write Links to other Classes discovered at code reversing time; Parents and Implements connectors only.	
<b>GenType</b>	<i>String</i>	Read/Write The code generation type; for example, Java, C++, C#, VBNNet, Visual Basic, Delphi.	
<b>Header1</b>	<i>Variant</i>	Read/Write	


Attribute	Type	Notes	See also
		A user defined string for inclusion as header in the source files generated.	
<b>Header2</b>	<i>Variant</i>	Read/Write  Same as for <b>Header1</b> , but used in the CPP source file.	
<b>IsActive</b>	<i>Boolean</i>	Read/Write  Boolean value indicating whether the element is active or not.  <b>1</b> = True, <b>0</b> = False.	
<b>IsComposite</b>	<i>Boolean</i>	Read/Write  Indicates whether the element is composite or not.  <b>1</b> = True, <b>0</b> = False.	
<b>IsLeaf</b>	<i>Boolean</i>	Read/Write  Boolean value indicating whether the element is in leaf node or not.  <b>1</b> = True, <b>0</b> = False.	
<b>IsNew</b>	<i>Boolean</i>	Read/Write  Boolean value indicating whether the element is new or not.  <b>1</b> = True, <b>0</b> = False.	
<b>IsSpec</b>	<i>Boolean</i>	Read/Write  Boolean value indicating whether the element is a specification or not.  <b>1</b> = True, <b>0</b> = False.	
<b>Issues</b>	<i>Collection</i>	Read only  Collection of Issue objects.	<a href="#">Issue (Maintenance) Class</a> <sup>[2896]</sup>  <a href="#">Collection Class</a> <sup>[2829]</sup>
<b>Locked</b>	<i>Boolean</i>	Read/Write  Indicates if the element has been locked against further change.	

Attribute	Type	Notes	See also
<b>MetaType</b>	<i>String</i>	Read only  The element's domain-specific meta type, as defined by an applied stereotype from an MDG Technology.	
<b>Methods</b>	<i>Collection</i>	Read only  Collection of Method objects for current element.	<a href="#">Method Class</a> <sup>[2919]</sup> <a href="#">Collection Class</a> <sup>[2829]</sup>
<b>MethodsEx</b>	<i>Collection</i>	Read only  Collection of Method objects belonging to the current element and its parent elements.	<a href="#">Method Class</a> <sup>[2919]</sup> <a href="#">Collection Class</a> <sup>[2829]</sup>
<b>Metrics</b>	<i>Collection</i>	Read only  Collection of Metric elements for current element.	<a href="#">Metric Class</a> <sup>[2898]</sup> <a href="#">Collection Class</a> <sup>[2829]</sup>
<b>MiscData</b>	<i>String</i>	Read only  This low-level property provides information about the contents of the <b>PData<sub>x</sub></b> fields.  These database fields are not documented, and developers must gain understanding of these fields through their own endeavors to use this property.  <b>MiscData</b> is zero based, therefore: <ul style="list-style-type: none"> <li>• <b>MiscData(0)</b> corresponds to <b>PData1</b></li> <li>• <b>MiscData(1)</b> to <b>PData2</b></li> </ul> and so on.	
<b>Modified</b>	<i>Date</i>	Read/Write  The date the element was last modified.	
<b>Multiplicity</b>	<i>String</i>	Read/Write  Multiplicity value for this element.	
<b>Name</b>	<i>String</i>	Read/Write  The element name; should be unique within the current package.	
<b>Notes</b>	<i>String</i>	Read/Write	

Attribute	Type	Notes	See also
		Further descriptive text about the element.	
<b>ObjectType</b>	<i>ObjectType</i>	Read only Distinguishes objects referenced through a Dispatch interface.	<a href="#">ObjectType</a> <sup>[2822]</sup>
<b>PackageID</b>	<i>Long</i>	Read/Write A local ID for the Package containing this element.	
<b>ParentID</b>	<i>Long</i>	Read/Write If this element is a child of another, used to set or retrieve the <i>ElementID</i> of the other element; if not, returns <b>0</b> .	
<b>Partitions</b>	<i>Collection</i>	Read only List of logical partitions into which an element can be divided.  Only valid for elements that support partitions, such as Activities and States.	<a href="#">Partitions Collection</a> <sup>[2928]</sup>
<b>Persistence</b>	<i>String</i>	Read/Write The persistence associated with this element; can be <b>Persistent</b> or <b>Transient</b> .	
<b>Phase</b>	<i>String</i>	Read/Write The phase this element is scheduled to be constructed in; any string value.	
<b>Priority</b>	<i>String</i>	Read/Write The priority of this element as compared to other project elements; only applies to Requirement, Change and Issue types, otherwise ignored.  Valid values are: <b>Low</b> , <b>Medium</b> and <b>High</b> .	
<b>Properties</b>	<i>Properties</i>	Returns a list of specialized properties that apply to the element that might not be available using the automation model.  The properties are purposely undocumented because of their obscure nature and because they are subject to change as progressive enhancements are made to them.	<a href="#">Properties</a> <sup>[2929]</sup>

Attribute	Type	Notes	See also
<b>PropertyType</b>	<i>Long</i>	Read/Write The ElementID of a Type associated with this element; only valid for Port and Part elements.	
<b>PropertyTypeName</b>	<i>String</i>	Read The name of a Type associated with this element; only valid for <b>Port</b> and <b>Part</b> elements.	
<b>Realizes</b>	<i>Collection</i>	Read only List of Interfaces realized by this element for convenience.	<a href="#">Collection Class</a> <sup>[2829]</sup>
<b>Requirements</b>	<i>Collection</i>	Read only Collection of <b>Requirement</b> objects.	<a href="#">Requirement Class</a> <sup>[2899]</sup> <a href="#">Collection Class</a> <sup>[2829]</sup>
<b>RequirementsEx</b>	<i>Collection</i>	Read only Collection of <b>Requirement</b> objects belonging to the current element and its parent elements.	<a href="#">Requirement Class</a> <sup>[2899]</sup> <a href="#">Collection Class</a> <sup>[2829]</sup>
<b>Resources</b>	<i>Collection</i>	Read only Collection of <b>Resource</b> objects for current element.	<a href="#">Resource Class</a> <sup>[2900]</sup> <a href="#">Collection Class</a> <sup>[2829]</sup>
<b>Risks</b>	<i>Collection</i>	Read only Collection of <b>Risk</b> objects.	<a href="#">Risk Class</a> <sup>[2902]</sup> <a href="#">Collection Class</a> <sup>[2829]</sup>
<b>RunState</b>	<i>String</i>	Read/Write The object's runstate list as a string.	
<b>Scenarios</b>	<i>Collection</i>	Read only Collection of <b>Scenario</b> objects for current element.	<a href="#">Scenario Class</a> <sup>[2903]</sup> <a href="#">Collection Class</a> <sup>[2829]</sup>
<b>StateTransitions</b>	<i>Collection</i>	Read only List of State Transitions that an element can support; applies in particular to Timing elements.	<a href="#">Collection Class</a> <sup>[2829]</sup> <a href="#">Transitions Collection</a> <sup>[2932]</sup>
<b>Status</b>	<i>String</i>	Read/Write	



Attribute	Type	Notes	See also
		Sets or gets the status, such as <b>Proposed</b> or <b>Approved</b> .	
<b>Stereotype</b>	<i>String</i>	Read/Write  The primary element stereotype; the first of the list of stereotypes you can access using the <i>StereotypeEx</i> attribute.	
<b>StereotypeEx</b>	<i>String</i>	Read/Write  All the applied stereotypes of the element in a comma-separated list.	
<b>StyleEx</b>	<i>String</i>	Read/Write  Advanced style settings; reserved for the use of Sparx Systems.	
<b>Subtype</b>	<i>Long</i>	Read/Write  A numeric subtype that qualifies the <b>Type</b> of the main element <ul style="list-style-type: none"> <li>For Event: <b>0</b> = Receiver, <b>1</b> = Sender</li> <li>For Class: <b>1</b> = Parameterised, <b>2</b> = Instantiated, <b>3</b> = Both, <b>0</b> = Neither, <b>17</b> = Association Class</li> </ul> If <b>17</b> , because an Association Class has been created through the user interface, <b>MiscData(3)</b> contains the ID of the related Association; as MiscData is read-only, you cannot create an Association Class through the Automation Interface: <ul style="list-style-type: none"> <li>For Note: <b>1</b> = Note linked to connector, <b>2</b> = Constraint linked to connector</li> <li>For StateNode: <b>100</b> = ActivityInitial, <b>101</b> = ActivityFinal</li> <li>For Activity: <b>0</b> = Activity, <b>8</b> = composite Activity (also set to <b>8</b> for other composite elements such as Use Cases)</li> <li>For Synchronization: <b>0</b> = Horizontal, <b>1</b> = Vertical</li> </ul> Note that there are many more Types than indicated in the above examples.	<a href="#">Type</a> 
<b>Tablespace</b>	<i>String</i>	Read/Write	

Attribute	Type	Notes		See also
		Associated tablespace for a Table element.		
<b>Tag</b>	<i>String</i>	Read/Write Corresponds to the <b>Keywords</b> field in the Enterprise Architect user interface.		<a href="#">General Settings</a> [958]
<b>TaggedValues</b>	<i>Collection</i>	Read only Returns a collection of <b>TaggedValue</b> objects.		<a href="#">TaggedValue Class</a> [2907] <a href="#">Collection Class</a> [2829]
<b>TaggedValuesEx</b>	<i>Collection</i>	Read only Returns a collection of <b>TaggedValue</b> objects belonging to the current element and the elements specialized or realized by the current element.		<a href="#">TaggedValue Class</a> [2907] <a href="#">Collection Class</a> [2829]
<b>TemplateParameters</b>	<i>Collection</i>	Read Only A collection of <b>TemplateParameter</b> objects.		<a href="#">Parameterized Classes (Templates)</a> [1365] <a href="#">TemplateParameter Class</a> [2931] <a href="#">Collection Class</a> [2829]
<b>Tests</b>	<i>Collection</i>	Read only A collection of Test objects for the current element.		<a href="#">Test Class</a> [2908] <a href="#">Collection Class</a> [2829]
<b>TreePos</b>	<i>Long</i>	Read/Write Sets or gets the tree position.		
<b>Type</b>	<i>String</i>	Read/Write The element type (such as Class, Component). Note that <b>Type</b> is case sensitive inside Enterprise Architect and should be provided with an initial capital (proper case); valid types are:		
		<b>Action</b> <b>Activity</b> <b>ActivityPartition</b> <b>ActivityRegion</b> <b>Actor</b> <b>Artifact</b> <b>Association</b> <b>Boundary</b>	<b>InteractionFragment</b> <b>InteractionOccurrence</b> <b>InteractionState</b> <b>Interface</b> <b>InterruptibleActivityRegion</b> <b>Issue</b> <b>Node</b>	

Attribute	Type	Notes		See also
		<b>Change</b> <b>Class</b> <b>Collaboration</b> <b>Component</b> <b>Constraint</b> <b>Decision</b> <b>DeploymentSpecification</b> <b>DiagramFrame</b> <b>EmbeddedElement</b> <b>Entity</b> <b>EntryPoint</b> <b>Event</b> <b>ExceptionHandler</b> <b>ExitPoint</b> <b>ExpansionNode</b> <b>ExpansionRegion</b> <b>Feature</b> <b>GUIElement</b>	<b>Note</b> <b>Object</b> <b>Package</b> <b>Parameter</b> <b>Part</b> <b>Port</b> <b>ProvidedInterface</b> <b>Report</b> <b>RequiredInterface</b> <b>Requirement</b> <b>Screen</b> <b>Sequence</b> <b>State</b> <b>StateNode</b> <b>Synchronization</b> <b>Text</b> <b>TimeLine</b> <b>UMLDiagram</b> <b>UseCase</b>	
<b>Version</b>	<i>String</i>	Read/Write  The version of the element.		
<b>Visibility</b>	<i>String</i>	Read/Write  The Scope of this element within the current package.  Valid values are: <b>Public</b> , <b>Private</b> , <b>Protected</b> or <b>Package</b> .		

### Element Methods

Method	Type	Notes	See also
<b>ApplyGroupLock (string aGroupName)</b>	<i>Boolean</i>	Applies a group lock to the element object, for the specified group, on behalf of the current user. Returns <b>true</b> if the operation is successful; returns <b>false</b> if the operation is unsuccessful. Use <i>GetLastError()</i> to retrieve error information. Parameters: <ul style="list-style-type: none"> <li>aGroupName: String - the name of the user group for which to set the group lock</li> </ul>	
<b>ApplyUserLock ()</b>	<i>Boolean</i>	Applies a user lock to the element object for the current user. Returns <b>true</b> if the operation is successful; returns	

Method	Type	Notes	See also
		<b>false</b> if the operation is unsuccessful. Use <i>GetLastError()</i> to retrieve error information.	
<b>CreateAssociationClass (long ConnectorID)</b>	<i>Boolean</i>	<p>Makes this element an AssociationClass of the Association with the provided <b>Connector ID</b>; the return value indicates whether the function succeeded in converting the element to an AssociationClass.</p> <p>AssociationClasses are created only where:</p> <ul style="list-style-type: none"> <li>• The current element is valid</li> <li>• The current element is a Class</li> <li>• The current element is not already an AssociationClass</li> <li>• The specified connector exists</li> <li>• The specified connector is an Association</li> <li>• The specified connector is not already in an AssociationClass pair</li> <li>• The current element is not at either end of the specified connector</li> </ul> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• ConnectorID: Long - the Connector ID of an Association connector</li> </ul>	<a href="#">UnlinkFromAssociation</a> <sup>[2895]</sup>
<b>DeleteLinkedDocument()</b>	<i>Boolean</i>	<p>Removes the Linked Document for the element. This method does not display a confirmatory prompt.</p> <p>Returns <b>true</b> if a document was deleted.</p>	
<b>GetBusinessRules()</b>	<i>String</i>	<p>Read Only.</p> <p>Returns all the Business Rules for the element.</p>	<a href="#">Model Business Rules</a> <sup>[1821]</sup>
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.	
<b>GetLinkedDocument ()</b>	<i>String</i>	<p>Returns a string value containing the element's linked document contents, in RTF format.</p> <p>If the element contains no linked document, an empty string is returned.</p>	
<b>GetRelationSet (EnumRelationSetType Type)</b>	<i>String</i>	Returns a string containing a comma-separated list of ElementIDs of directly- and indirectly-related elements based on the given type.	<a href="#">EnumRelationSetType</a> <sup>[2820]</sup> <a href="#">Connector</a> <sup>[2933]</sup>

Method	Type	Notes	See also
		<p>Recurses using the same relation type on all elements it finds, retrieving all dependencies and sub-dependencies of the current element; for example, <i>Object1</i> depends on <i>Object2</i>, which depends on <i>Object3</i>, therefore this method returns <i>Object2</i> and <i>Object3</i>.</p> <p>To obtain only the direct relationships of the element, use the <i>Connector</i> collection instead.</p>	
<b>GetStereotypeList ()</b>	<i>String</i>	Returns a comma-separated list of stereotypes allied to this element.	
<b>IsAssociationClass</b>	<i>Boolean</i>	Returns whether or not the current element is an AssociationClass.	
<b>LoadLinkedDocument (string Filename)</b>	<i>Boolean</i>	<p>Loads the RTF document from the specified file into the element's linked document.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• <b>FileName:</b> String - the name of the file from which to load the RTF document</li> </ul>	
<b>Refresh ()</b>	<i>Void</i>	<p>Refreshes the element features in the Project Browser.</p> <p>Usually called after adding or deleting attributes or methods, when the user interface is required to be updated as well.</p>	
<b>ReleaseUserLock ()</b>	<i>Boolean</i>	<p>Releases a user lock or group lock on the element object.</p> <p>Returns <b>true</b> if the operation is successful; returns <b>false</b> if the operation is unsuccessful. Use <i>GetLastError()</i> to retrieve error information.</p>	
<b>SaveLinkedDocument (string Filename)</b>	<i>Boolean</i>	<p>Saves the linked document for this element to the specified RTF file.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• <b>FileName:</b> String - the name of the RTF file to which to save the linked document</li> </ul>	
<b>SetAppearance (long Scope, long Item, long Value)</b>	<i>Void</i>	<p>Sets the visual appearance of the element.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• <b>Scope:</b> Long - Scope of appearance set to modify 1 – Base (Default appearance across entire</li> </ul>	<a href="#">DiagramObject Class</a> <sup>[2958]</sup>

Method	Type	Notes	See also
		<p>model)</p> <p>To set appearance for the element (diagram object) in a selected diagram only, see <i>Setting The Style</i> in the <i>Diagram Object Class</i> topic</p> <ul style="list-style-type: none"> <li>Item: Long - Appearance feature to modify <ul style="list-style-type: none"> <li>0 – Background color</li> <li>1 – Font Color</li> <li>2 – Border Color</li> <li>3 – Border Width</li> </ul> </li> <li>Value: Long - Value to set appearance to</li> </ul>	
<b>SetCompositeDiagram ()</b>	<i>Boolean</i>	<p>Sets the composite diagram of the element.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>GUID: String - the GUID of the composite diagram. A blank GUID will remove the link to the composite diagram</li> </ul>	<a href="#">Composite Elements</a> <sup>[936]</sup>
<b>SetCreated (Date NewVal)</b>	<i>Void</i>	<p><b>Deprecated</b></p> <p>This method is no longer supported.</p>	
<b>SetModified (Date NewVal)</b>	<i>Void</i>	<p><b>Deprecated</b></p> <p>This method is no longer supported.</p>	
<b>SynchConstraints (string Profile, string Stereotype)</b>	<i>Boolean</i>	<p>Synchronizes the constraints of a UML Profile item for this element, only if the specified stereotype has been applied.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>Profile: String - Name of the profile that contains the stereotype</li> <li>Stereotype: String - Name of the profile <i>stereotype</i> for which the default constraints are to be synchronized</li> </ul>	
<b>SynchTaggedValues (string Profile, string Stereotype)</b>	<i>Boolean</i>	<p>Synchronizes the Tagged Values of a UML Profile item for this element, only if the specified stereotype has been applied.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>Profile: String - Name of the profile that contains the stereotype</li> <li>Stereotype: String - Name of the profile <i>stereotype</i> for which the default tags are to</li> </ul>	

Method	Type	Notes	See also
		be synchronized	
<b>UnlinkFromAssociation</b>	<i>Boolean</i>	Performs the opposite of <b>CreateAssociationClass()</b> .	<a href="#">CreateAssociationClass()</a> <sup>[2892]</sup>
<b>Update ()</b>	<i>Boolean</i>	Updates the current element object after modification or appending a new item.  If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.	

#### 20.2.2.5.4 File Class

A **File** represents an associated file for an element. Files are accessed through the Element **Files** collection.

Associated table in .EAP file *t\_objectfiles*

##### File Attributes

Attribute	Type	Notes
<b>FileDate</b>	<i>String</i>	Read/Write The file date when the entry was created.
<b>Name</b>	<i>String</i>	Read/Write The file name can be a logical file or a reference to a web address (using <i>http://</i> ).
<b>Notes</b>	<i>String</i>	Read/Write Notes about the file.
<b>ObjectType</b>	<a href="#">Object Type</a> <sup>[2822]</sup>	Read only Distinguishes objects referenced through a Dispatch interface.
<b>Size</b>	<i>String</i>	Read/Write The file size.
<b>Type</b>	<i>String</i>	Read/Write The file type.

**File Methods:**

Method	Type	Notes
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.
<b>Update ()</b>	<i>Boolean</i>	Updates the current File object after modification or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.

**20.2.2.5.5 Issue (Maintenance) Class**

An **Issue** is either a **Change** or a **Defect**, is associated with the containing element, and is accessed through the **Issues** collection of an element.

**Associated table in .EAP file** *t\_objectproblems*

**Issue Attributes**

Attribute	Type	Notes
<b>DateReported</b>	<i>Date</i>	Read/Write The date the issue was reported.
<b>DateResolved</b>	<i>Date</i>	Read/Write The date the issue was resolved.
<b>ElementID</b>	<i>Long</i>	Read/Write The ID of the element associated with this issue.
<b>Name</b>	<i>String</i>	Read/Write The Issue name; that is, the Issue itself.
<b>Notes</b>	<i>String</i>	Read/Write The Issue description.
<b>ObjectType</b>	<a href="#">ObjectTyp</a> <a href="#">e</a> <sub>[2822]</sub>	Read only Distinguishes objects referenced through a Dispatch interface.
<b>Priority</b>	<i>String</i>	Read/Write



Attribute	Type	Notes
		The priority of the Issue - <b>Low</b> , <b>Medium</b> or <b>High</b> .
<b>Reporter</b>	<i>String</i>	Read/Write The user ID of the person reporting the issue.
<b>Resolver</b>	<i>String</i>	Read/Write The user ID of the person resolving the issue.
<b>ResolverNotes</b>	<i>String</i>	Read/Write Notes entered by the resolver about resolution of the Issue.
<b>Severity</b>	<i>String</i>	Read/Write The Issue severity - <b>Low</b> , <b>Medium</b> or <b>High</b> .
<b>Status</b>	<i>String</i>	Read/Write The current status of the issue.
<b>Type</b>	<i>Variant</i>	Read/Write The Issue type - <b>Defect</b> , <b>Change</b> , <b>Issue</b> or <b>ToDo</b> .
<b>Version</b>	<i>String</i>	Read/Write The version associated with the issue. Note that this method is only available through a Dispatch interface.  Object ob = Issue; Print ob.Version;

**Issue Methods**

Method	Type	Notes
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.
<b>Update ()</b>	<i>Boolean</i>	Updates the current Issue object after modification or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.

#### 20.2.2.5.6 Metric Class

A **Metric** is a named item with a weighting that can be associated with an element for purposes of building metrics about the model. Metrics are accessed through the Element **Metrics** collection.

Associated table in .EAP file    *t\_objectmetrics*

##### Metric Attributes

Attribute	Type	Notes
<b>Name</b>	<i>String</i>	Read/Write The name of the metric.
<b>Notes</b>	<i>String</i>	Read/Write Notes about this metric.
<b>ObjectType</b>	<a href="#">ObjectTyp</a> <a href="#">e</a> <sub>[2822]</sub>	Read only Distinguishes objects referenced through a Dispatch interface.
<b>Type</b>	<i>String</i>	Read/Write The metric type.
<b>Weight</b>	<i>Long</i>	Read/Write A user-defined weighting for estimation or metric purposes.

##### Metric Methods

Method	Type	Notes
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.
<b>Update ()</b>	<i>Boolean</i>	Updates the current Metric object after modification or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.

### 20.2.2.5.7 Requirement Class

An **Element Requirement** object holds information about the responsibilities of an element in the context of the model. Requirements can be accessed using the Element **Requirements** collection.

Associated table in .EAP file *t\_objectrequires*

#### Requirement Attributes

Attribute	Type	Notes
<b>Difficulty</b>	<i>String</i>	Read/Write The estimated difficulty of implementing the requirement.
<b>LastUpdate</b>	<i>Date</i>	Read/Write The date the requirement was last updated.
<b>Name</b>	<i>String</i>	Read/Write The requirement itself.
<b>Notes</b>	<i>String</i>	Read/Write Further notes on the requirement.
<b>ObjectType</b>	<a href="#">ObjectTyp</a> <small>e<sub>2822</sub></small>	Read only Distinguishes objects referenced through a Dispatch interface.
<b>ParentID</b>	<i>Long</i>	Read only The <b>ElementID</b> of the element to which this requirement applies.
<b>Priority</b>	<i>String</i>	Read/Write The assigned priority of the requirement.
<b>RequirementID</b>	<i>Long</i>	Read only A local ID for this requirement.
<b>Stability</b>	<i>String</i>	Read/Write The estimated stability of the requirement.  This is an indication of the probability of the requirement - or understanding of the requirement - changing. High stability indicates a low probability of the requirement changing.

Attribute	Type	Notes
<b>Status</b>	<i>String</i>	Read/Write The current status of the requirement.
<b>Type</b>	<i>String</i>	Read/Write The requirement type.

#### Requirement Methods

Method	Type	Notes
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.
<b>Update ()</b>	<i>Boolean</i>	Updates the current Requirement object after modification or appending a new item.  If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.

#### 20.2.2.5.8 Resource Class

An Element **Resource** is a named person/task pair with timing constraints and percent complete indicators. Use this to manage the work associated with delivering an Element.

Associated table in .EAP file *t\_objectresources*

#### Resource Attributes

Attribute	Type	Notes
<b>ActualHours</b>	<i>Long</i>	Read/Write The time already expended on the task, in hours, days or other units.
<b>DateEnd</b>	<i>Date</i>	Read/Write The expected end date.
<b>DateStart</b>	<i>Date</i>	Read/Write The date to start work.

Attribute	Type	Notes
<b>ExpectedHours</b>	<i>Long</i>	Read/Write The total expected time the task might run, in hours, days or other units.
<b>History</b>	<i>String</i>	Read/Write Gets or sets history text.
<b>Name</b>	<i>String</i>	Read/Write The name of the resource (for example, a person's name).
<b>Notes</b>	<i>String</i>	Read/Write Descriptive notes.
<b>ObjectType</b>	<a href="#">ObjectType</a> [2822]	Read only Distinguishes objects referenced through a Dispatch interface.
<b>PercentComplete</b>	<i>Long</i>	Read/Write The current percent complete figure.
<b>Role</b>	<i>String</i>	Read/Write The role the resource plays in implementing the element.
<b>Time</b>	<i>Long</i>	Read/Write The time expected to complete the task; a numeric indicating the number of days.

#### Resource Methods

Method	Type	Notes
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.  This function is rarely used as an exception is thrown when an error occurs.
<b>Update ()</b>	<i>Boolean</i>	Update the current Resource object after modification or appending a new item.  If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.

Method	Type	Notes

#### 20.2.2.5.9 Risk Class

A **Risk** object represents a named risk associated with an element, it is used for project management purposes. Risks can be accessed through the Element **Risks** collection.

Associated table in .EAP file    *t\_objectrisks*

##### Risk Attributes

Attribute	Type	Notes
<b>Name</b>	<i>String</i>	Read/Write The name of the risk.
<b>Notes</b>	<i>String</i>	Read/Write Further notes describing the risk.
<b>ObjectType</b>	<a href="#">ObjectTyp</a> <small>e[2822]</small>	Read only Distinguishes objects referenced through a Dispatch interface.
<b>Type</b>	<i>String</i>	Read/Write The risk type associated with this element.
<b>Weight</b>	<i>Long</i>	Read/Write A weighting for estimation or metric purposes.

##### Risk Methods

Method	Type	Notes
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.
<b>Update ()</b>	<i>Boolean</i>	Update the current Risk object after modification or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.

### 20.2.2.5.10 Scenario Class

A **Scenario** corresponds to a Collaboration or Use Case instance. Each Scenario is a path of execution through the logic of a Use Case. Scenarios can be added to using the Element *Scenarios* collection.

**Associated table in .EAP file** `t_objectscenarios`

#### Scenario Attributes

Attribute	Type	Notes	See also
<b>Name</b>	<i>String</i>	Read/Write The Scenario name.	
<b>Notes</b>	<i>String</i>	Read/Write A description of the Scenario, usually containing the steps to execute the scenario.	
<b>ObjectType</b>	<a href="#">ObjectType</a> [2822]	Read only Distinguishes objects referenced through a Dispatch interface.	
<b>ScenarioGUID</b>	<i>String</i>	Read/Write A unique ID for the Scenario, used to identify the Scenario unambiguously within a model.	
<b>Steps</b>	<a href="#">Collection of ScenarioStep</a> [2905]	Read only A collection of step objects for this Scenario.  Use the <i>AddNew</i> and <i>Delete</i> functions to manage steps. <i>AddNew</i> passes the step name and "1" as the type for an actor step.	<a href="#">Collection Class</a> [2823]
<b>Type</b>	<i>String</i>	Read/Write The scenario type (for example, <i>Basic Path</i> ).	
<b>Weight</b>	<i>Long</i>	Read/Write Currently used to position scenarios in the scenario list (that is, <i>List Position</i> ).	
<b>XMLContent</b>	<i>String</i>	Read/Write A structured field that can contain scenario details in XML format. It is recommended that you use the <i>Steps</i> collection to read or modify this field.	

Attribute	Type	Notes	See also

### Scenario Methods

Method	Type	Notes
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.
<b>Update ()</b>	<i>Boolean</i>	Update the current Scenario object after modification or appending a new item.  If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.

#### 20.2.2.5.11 ScenarioExtension Class

### ScenarioExtension Attributes

Attribute	Type	Notes
<b>ExtensionGUID</b>	<i>String</i>	Read/Write A unique GUID for this Extension.
<b>Join</b>	<i>String</i>	Read/Write The GUID of the step where this Extension rejoins the Scenario.
<b>JoiningStep</b>	<a href="#">ScenarioStep</a> [2905]	Read only The actual step where this Extension rejoins the Scenario, if any.
<b>Level</b>	<i>String</i>	Read only The number of this Extension as shown in the scenario editor. This is derived from the value of <i>Pos</i> for this object and the owning step.
<b>Name</b>	<i>String</i>	Read/Write The Extension name. This should match the name of the linked scenario.
<b>ObjectType</b>	<a href="#">ObjectType</a> [2822]	Read only Distinguishes objects referenced through a Dispatch interface.
<b>Pos</b>	<i>Long</i>	Read/Write



Attribute	Type	Notes
		The position of the Extension in the Extensions list.
<b>Scenario</b>	<a href="#">Scenario</a> <sup>[2903]</sup>	Read only The scenario that is executed as an alternative path for this Extension.

#### ScenarioExtension Methods

Method	Type	Notes
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.
<b>Update ()</b>	<i>Boolean</i>	Updates the current <b>ScenarioExtension</b> object after modification or appending a new item. If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.

#### 20.2.2.5.12 ScenarioStep Class

##### ScenarioStep Attributes

Attribute	Type	Notes	See also
<b>Extensions</b>	<i>Collection of ScenarioExtension</i>	Read only A collection of <b>ScenarioExtension</b> objects that specify how the scenario is extended from this step. The arguments to <b>AddNew</b> should match the name and GUID of the alternative scenario being linked to.	<a href="#">Collection</a> <sup>[2829]</sup> <a href="#">ScenarioExtension</a> <sup>[2904]</sup>
<b>Level</b>	<i>String</i>	Read only The number of this Step as shown in the scenario editor. This is derived from the value of <b>Pos</b> .	
<b>Link</b>	<i>String</i>	Read/Write The GUID of a Use Case that is relevant to this step.	
<b>LinkedElement</b>	<i>Element</i>	Read only The actual element specified by Link, if any.	<a href="#">Element</a> <sup>[2881]</sup>
<b>Name</b>	<i>String</i>	Read/Write	

Attribute	Type	Notes	See also
		The Step name.	
<b>ObjectType</b>	<i>ObjectType</i>	Read only Distinguishes objects referenced through a Dispatch interface.	<a href="#">ObjectType</a> <sup>[2822]</sup>
<b>Pos</b>	<i>Long</i>	Read/Write The position of the Step in the Scenario Step list.	
<b>Results</b>	<i>String</i>	Read/Write Any results that are given from this step.	
<b>State</b>	<i>String</i>	Read/Write A description of the state the system enters when this Step is executed.	
<b>StepGUID</b>	<i>String</i>	Read/Write A unique GUID for this Step.	
<b>StepType</b>	<i>ScenarioStepType</i>	Read/Write Identifies whether this step is being performed by a user or the system.	<a href="#">ScenarioStepType</a> <sup>[2825]</sup>
<b>Uses</b>	<i>String</i>	Read/Write The input and requirements that are relevant to this step.	
<b>UsesElement List</b>	<i>Collection of Element</i>	Read only Indicates that the Structured Specification tab <b>Uses</b> field is a linked element list.	<a href="#">Collection</a> <sup>[2829]</sup> <a href="#">Element</a> <sup>[2881]</sup> <a href="#">Set Up Scenario Specification</a> <sup>[968]</sup>

**ScenarioStep Methods:**

Method	Type	Notes
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.

Method	Type	Notes
<b>Update ()</b>	<i>Boolean</i>	Updates the current <b>ScenarioStep</b> object after modification or appending a new item.  If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.

#### 20.2.2.5.13 TaggedValue Class

A **TaggedValue** is a named property and value associated with an element. Tagged Values can be accessed through the **TaggedValues** collection.

Associated table in .EAP file    *t\_objectproperties*

#### TaggedValue Attributes

Attribute	Type	Notes
<b>ElementID</b>	<i>Long</i>	Read/Write The local ID of the associated element.
<b>FQName</b>	<i>String</i>	Read only The fully-qualified name of the tag.
<b>Name</b>	<i>String</i>	Read/Write The name of the tag.
<b>Notes</b>	<i>String</i>	Read/Write Further descriptive notes about this tag.  If <b>Value</b> (below) is set to "<memo>", then <b>Notes</b> should contain the actual Tagged Value content.
<b>ObjectType</b>	<a href="#">ObjectTyp</a> <small>e<sup>[2822]</sup></small>	Read only Distinguishes objects referenced through a Dispatch interface.
<b>PropertyGUID</b>	<i>String</i>	Read/Write The global ID of the tag.
<b>PropertyID</b>	<i>Long</i>	Read only The local ID of the tag.

Attribute	Type	Notes
<b>Value</b>	<i>String</i>	<p>Read/Write</p> <p>The value assigned to this tag.</p> <p>This field has a 255 character limit. If the value is greater than 255 characters long, set the value to "&lt;memo&gt;" and insert the body of text in the <b>Notes</b> attribute (above).</p> <p>When reading existing Tagged Values, if <b>Value</b> = "&lt;memo&gt;" then the developer should read the actual body of text from the <b>Notes</b> attribute.</p>

#### TaggedValue Methods

Method	Type	Notes
<b>GetAttribute(string propName)</b>	<i>String</i>	<p>Returns the text of a single named property within a structured Tagged Value.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>propName: String - the name of the property for which the text is being returned</li> </ul>
<b>GetLastError()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.
<b>HasAttributes()</b>	<i>Boolean</i>	Returns true if the Tagged Value is a structured Tagged Value with one or more properties.
<b>SetAttribute(string propName, string propValue)</b>	<i>Boolean</i>	<p>Sets the text of a single named property within a structured Tagged Value.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>propName: String - the name of the property for which the text is being set</li> <li>propValue: the value of the property</li> </ul>
<b>Update()</b>	<i>Boolean</i>	<p>Updates the current TaggedValue object after modification or appending a new item.</p> <p>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.</p>

#### 20.2.2.5.14 Test Class

A **Test** is a single Test Case applied to an element. Tests are added and accessed through the Element **Tests** collection.

Associated table in .EAP file    *t\_objecttests*

**Test Attributes**

Attribute	Type	Notes
<b>AcceptanceCriteria</b>	<i>String</i>	Read/Write The acceptance criteria for successful execution.
<b>CheckedBy</b>	<i>String</i>	Read/Write User ID of the person confirming the results.
<b>Class</b>	<i>Long</i>	Read/Write The test Class: 1 = Unit Test 2 = Integration Test 3 = System Test 4 = Acceptance Test 5 = Scenario Test
<b>DateRun</b>	<i>Date</i>	Read/Write The date the test was last run.
<b>Input</b>	<i>String</i>	Read/Write Input data for the test.
<b>Name</b>	<i>String</i>	Read/Write The test name.
<b>Notes</b>	<i>String</i>	Read/Write Detailed notes about test to be carried out.
<b>ObjectType</b>	<a href="#">ObjectType</a> <small>[2822]</small>	Read only Distinguishes objects referenced through a Dispatch interface.
<b>RunBy</b>	<i>String</i>	Read/Write The user ID of the person conducting the test.
<b>Status</b>	<i>String</i>	Read/Write Current status of test.

Attribute	Type	Notes
<b>TestResults</b>	<i>Variant</i>	Read/Write Results of test.
<b>Type</b>	<i>String</i>	Read/Write The test type, such as Load or Regression.

#### Test Methods

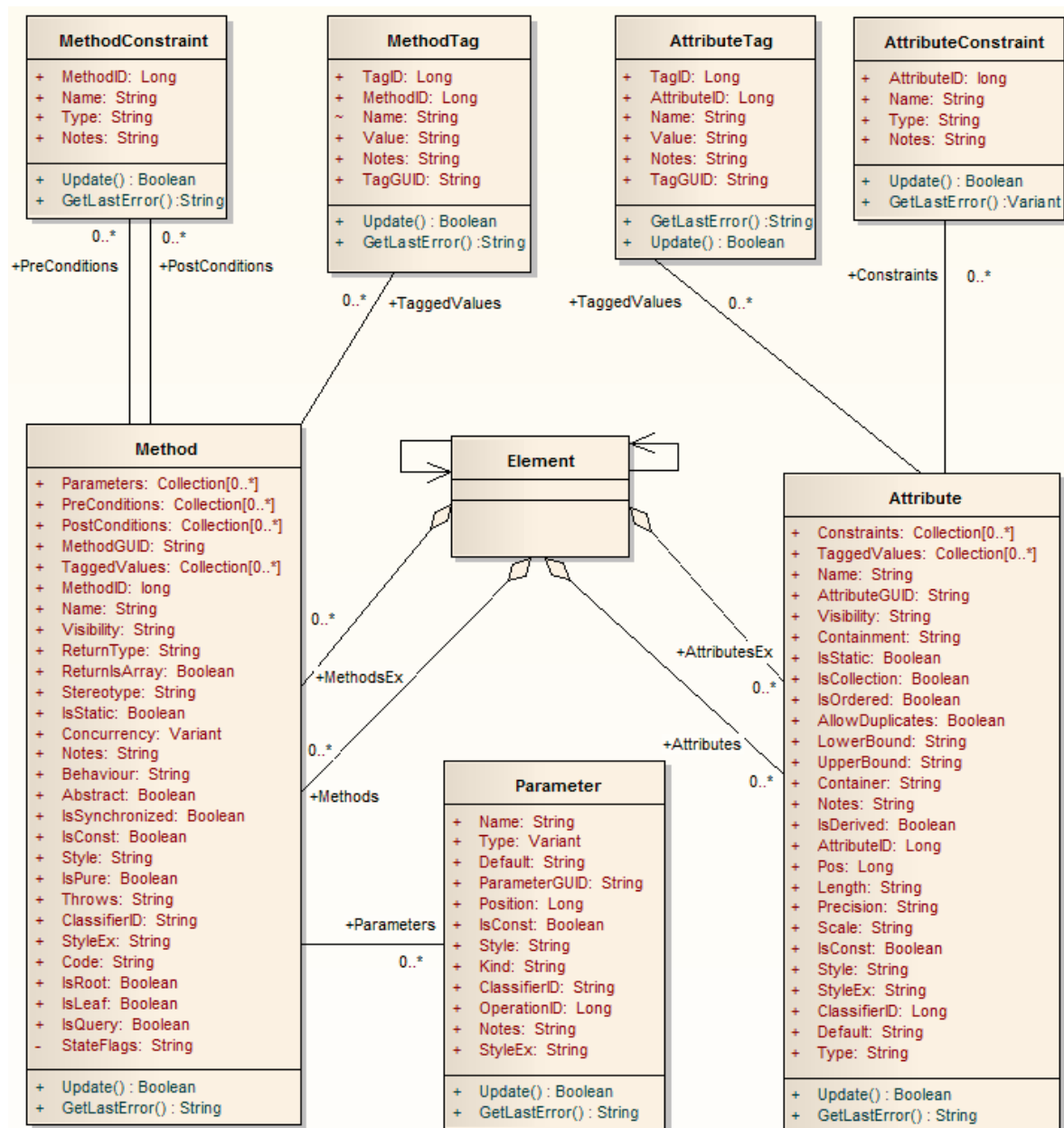
Method	Type	Notes
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.
<b>Update ()</b>	<i>Boolean</i>	Update the current Test object after modification or appending a new item.  If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.

### 20.2.2.6 Element Features Package

The *ElementFeatures* package contains descriptions of the model interfaces that enable access to operations and attributes, and their associated Tagged Values and constraints.

This diagram illustrates the components associated with element features. These include *Attributes* and *Methods*, and the associated constraints and Tagged Values related to them. It also includes the *Parameter* object that defines the arguments associated with an operation (method).

#### Example



#### 20.2.2.6.1 Attribute Class

An **attribute** corresponds to a UML Attribute. It contains further collections for constraints and Tagged Values. Attributes are accessed from the element **Attributes** collection.

Associated table in .EAP file *t\_attribute*

#### Attribute Attributes

Attribute	Type	Notes
<b>Alias</b>	<i>String</i>	Read/Write Contains the (optional) <b>Alias</b> property for this attribute. This can be used

Attribute	Type	Notes
		interchangeably with the <b>Style</b> attribute, below.
<b>AllowDuplicates</b>	<i>Boolean</i>	Read/Write Indicates if duplicates are allowed in the collection. If the attribute represents a database column this, when set, represents the <b>Not Null</b> option.
<b>AttributeGUID</b>	<i>String</i>	Read only A globally unique ID for the current attribute. This attribute is system generated.
<b>AttributeID</b>	<i>Long</i>	Read only The local ID number of the attribute.
<b>ClassifierID</b>	<i>Long</i>	Read/Write The classifier ID, if appropriate, indicating the base type associated with the attribute, if not a primitive type.
<b>Container</b>	<i>String</i>	Read/Write The container type.
<b>Containment</b>	<i>String</i>	Read/Write The type of containment - <b>Not Specified</b> , <b>By Reference</b> or <b>By Value</b> .
<b>Constraints</b>	<a href="#">Collection</a> <small>[2829]</small>	Read only A collection of <i>AttributeConstraint</i> objects, used to access and manage constraints associated with this attribute.
<b>Default</b>	<i>String</i>	Read/Write The initial value assigned to this attribute.
<b>IsCollection</b>	<i>Boolean</i>	Read/Write Indicates if the current feature is a collection or not. If the attribute represents a database column this, when set, represents a Foreign Key.
<b>IsConst</b>	<i>Boolean</i>	Read/Write A flag indicating if the attribute is <b>Const</b> or not.



Attribute	Type	Notes
<b>IsDerived</b>	<i>Boolean</i>	Read/Write Indicates if the attribute is derived (that is, a calculated value).
<b>IsID</b>	<i>Boolean</i>	Read/Write Indicates if the attribute uniquely identifies an instance of the containing Class, or not.
<b>IsOrdered</b>	<i>Boolean</i>	Read/Write Indicates if a collection is ordered or not. If the attribute represents a database column this, when set, represents a Primary Key.
<b>IsStatic</b>	<i>Boolean</i>	Read/Write Indicates if the current attribute is a static feature or not. If the attribute represents a database column this, when set, represents the <b>Unique</b> option.
<b>Length</b>	<i>String</i>	Read/Write The attribute length, where applicable.
<b>LowerBound</b>	<i>String</i>	Read/Write A value for the collection lower boundary.
<b>Name</b>	<i>String</i>	Read/Write The attribute name.
<b>Notes</b>	<i>String</i>	Read/Write Further notes on this attribute.
<b>ObjectType</b>	<a href="#">ObjectType</a> <small>[2822]</small>	Read only Distinguishes objects referenced through a Dispatch interface.
<b>ParentID</b>	<i>Long</i>	Read only Returns the <i>ElementID</i> of the element that this attribute is a part of.
<b>Pos</b>	<i>Long</i>	Read/Write The position of the attribute in the Class attribute list.

Attribute	Type	Notes
<b>Precision</b>	<i>String</i>	Read/Write The precision value.
<b>RedefinedProperty</b>	<i>String</i>	Read/Write Corresponds to the <b>Redefined Property</b> field on the Detail page of the attribute Properties dialog, or the UML <i>redefinedProperty</i> attribute. Contains a comma separated list of GUIDs.
<b>Scale</b>	<i>String</i>	Read/Write The scale value.
<b>Stereotype</b>	<i>String</i>	Read/Write Sets or gets the stereotype for this attribute.
<b>StereotypeEx</b>	<i>String</i>	Read/Write Provides all the applied stereotypes of the attribute, in a comma-separated list.
<b>Style</b>	<i>String</i>	Read/Write Contains the (optional) <b>Alias</b> property for this attribute. This can be used interchangeably with the <b>Alias</b> attribute, above.
<b>SubsettedProperty</b>	<i>String</i>	Read/Write Corresponds to the <b>Subsetted Property</b> field on the Detail page of the attribute Properties dialog, or the UML <i>subsettedProperty</i> attribute. Contains a comma separated list of GUIDs.
<b>StyleEx</b>	<i>String</i>	Read/Write Advanced style settings, reserved for the use of Sparx Systems.
<b>TaggedValues</b>	<a href="#">Collection</a> [2829] of type <a href="#">AttributeTag</a> [2916]	Read only A collection of <i>AttributeTag</i> objects, used to access and manage Tagged Values associated with this attribute.
<b>TaggedValuesEx</b>	<a href="#">Collection</a> [2829] of type <a href="#">TaggedValue</a> [2907]	Read only A collection of <i>TaggedValue</i> objects belonging to the current attribute and the <i>TaggedValuesEx</i> property of its classifier.

Attribute	Type	Notes
<b>Type</b>	<i>String</i>	Read/Write The attribute type (by name; also see <i>ClassifierID</i> ).
<b>UpperBound</b>	<i>String</i>	Read/Write A value for the collection upper boundary.
<b>Visibility</b>	<i>String</i>	Read/Write Identifies the scope of the attribute - <b>Private</b> , <b>Protected</b> , <b>Public</b> or <b>Package</b> .

#### Attribute Methods

Method	Type	Notes
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.
<b>Update ()</b>	<i>Boolean</i>	Updates the current attribute object after modifying or appending a new item.  If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.

#### **20.2.2.6.2 AttributeConstraint Class**

An **AttributeConstraint** is a constraint associated with the current Attribute.

Associated table in .EAP file    *t\_attributeconstraints*

#### AttributeConstraint Attributes

Attribute	Type	Notes
<b>AttributeID</b>	<i>Long</i>	Read/Write The ID of the attribute this constraint applies to.
<b>Name</b>	<i>String</i>	Read/Write The name of the constraint.
<b>Notes</b>	<i>String</i>	Read/Write

Attribute	Type	Notes
		Descriptive notes about the constraint.
<b>ObjectType</b>	<a href="#">ObjectType</a> [2822]	Read only Distinguishes objects referenced through a Dispatch interface.
<b>Type</b>	<i>String</i>	Read/Write The type of constraint.

#### AttributeConstraint Methods

Method	Type	Notes
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.
<b>Update ()</b>	<i>Boolean</i>	Update the current <b>AttributeConstraint</b> object after modification or appending a new item.  If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.

#### 20.2.2.6.3 *AttributeTag Class*

An *AttributeTag* represents a Tagged Value associated with an attribute.

**Associated table in .EAP file:** *t\_attributetag*

#### AttributeTag Attributes:

Attribute	Type	Notes
<b>AttributeID</b>	<i>Long</i>	Read/Write The local ID of the attribute associated with this Tagged Value.
<b>FQName</b>	<i>String</i>	Read only The fully-qualified name of the tag.
<b>Name</b>	<i>String</i>	Read/Write The name of the tag.

Attribute	Type	Notes
<b>Notes</b>	<i>String</i>	Read/Write Further descriptive notes about this tag. If <b>Value</b> (below) is set to "<memo>", then <b>Notes</b> should contain the actual Tagged Value content.
<b>ObjectType</b>	<a href="#">ObjectType</a> <small>[2822]</small>	Read only Distinguishes objects referenced through a Dispatch interface.
<b>TagGUID</b>	<i>String</i>	Read/Write A globally unique ID for this Tagged Value.
<b>TagID</b>	<i>Long</i>	Read only The local ID to identify the Tagged Value.
<b>Value</b>	<i>String</i>	Read/Write The value assigned to this tag. This field has a 255 character limit. If the value is greater than 255 characters long, set the value to "<memo>" and insert the body of text in the <b>Notes</b> attribute (above). When reading existing Tagged Values, if <b>Value</b> = "<memo>" then the developer should read the actual body of text from the <b>Notes</b> attribute.

**AttributeTag Methods:**

Method	Type	Notes
<b>GetAttribute</b> (string propName)	<i>String</i>	Returns the text of a single named property within a structured tagged value.
<b>GetLastError()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object. This function is rarely used as an exception is thrown when an error occurs.
<b>HasAttributes()</b>	<i>Boolean</i>	Returns true if the tagged values is a structured tagged value with one or more properties.
<b>SetAttribute</b> (string propName, string propValue)	<i>Boolean</i>	Sets the text of a single named property within a structured tagged value.

Method	Type	Notes
<b>Update ()</b>	<i>Boolean</i>	<p>Updates the current <i>AttributeTag</i> object after modification or appending a new item.</p> <p>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.</p>

#### 20.2.2.6.4 CustomProperties Collection

The *CustomProperties* collection contains 0 or more *Cust Properties* associated with the current element. These properties provide advanced UML configuration options, and must not be added to or deleted. The value of each property can be set.

##### CustomProperty:

Attribute	Type	Notes
<b>Name</b>	<i>String</i>	<p>Read only</p> <p>The CustomProperty name.</p>
<b>ObjectType</b>	<a href="#">ObjectType</a> <small>[2822]</small>	<p>Read only</p> <p>Distinguishes objects referenced through a Dispatch interface.</p>
<b>Value</b>	<i>String</i>	<p>Read/Write</p> <p>The value associated with this custom property. This can be:</p> <ul style="list-style-type: none"> <li>• A string</li> <li>• The boolean values <b>true</b> or <b>false</b>, or</li> <li>• An enumeration value from a defined list</li> </ul> <p>The UML 2.4.1 specification in general provides information on the kinds of enumeration relevant here.</p>

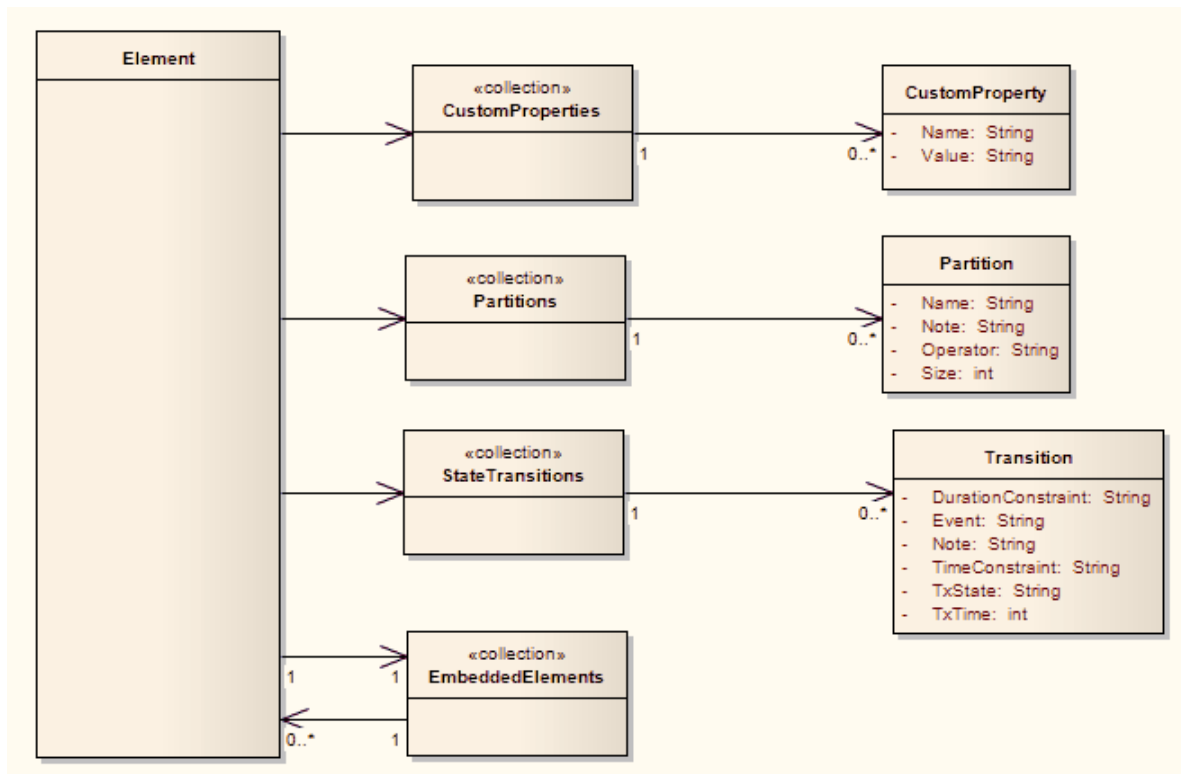
##### Notes

- The number and type of properties vary depending on the actual element

#### 20.2.2.6.5 EmbeddedElements Collection

In UML 2.4.1 an element can have one or more embedded elements such as Ports, Pins, Parameters or ObjectNodes. These are attached to the boundary of the host element and cannot be moved off the element. They are owned by their host element. This collection gives easy access to the set of elements embedded on the surface of an element. Note that some embedded elements can have their own embedded element collection (for example, Ports can have Interfaces embedded on them).

The *EmbeddedElements* collection contains Element objects.

Example**20.2.2.6.6 Method Class**

A *method* represents a UML *operation*. It is accessed from the *Element Methods* collection and includes collections for parameters, constraints and Tagged Values.

Associated table in .EAP file: *t\_operation*

Method Attributes:

Attribute	Type	Notes
<b>Abstract</b>	<i>Boolean</i>	Read/Write A flag indicating if the method is abstract (1) or not (0).
<b>Behavior</b>	<i>String</i>	Read/Write Some further explanatory behavior notes (for example, pseudocode). In earlier releases of Enterprise Architect this attribute had the UK/ Australian spelling 'Behaviour'; this is still present for backwards compatibility, but please now use the 'Behavior' attribute for consistency.
<b>ClassifierID</b>	<i>String</i>	Read/Write

Attribute	Type	Notes
		The Classifier ID that applies to the <i>ReturnType</i> .
<b>Code</b>	<i>String</i>	Read/Write An optional field to hold the method code (used for the <b>Initial Code</b> field).
<b>Concurrency</b>	<i>Variant</i>	Read/Write Indicates the concurrency type of the method.
<b>IsConst</b>	<i>Boolean</i>	Read/Write A flag indicating that the method is <b>Const</b> .
<b>IsLeaf</b>	<i>Boolean</i>	Read/Write A flag to indicate if the method is <i>Leaf</i> (cannot be overridden).
<b>IsPure</b>	<i>Boolean</i>	Read/Write A flag indicating that the method is defined as <b>Pure</b> in C++.
<b>IsQuery</b>	<i>Boolean</i>	Read/Write A flag to indicate if the method is a query (that is, does not alter Class variables).
<b>IsRoot</b>	<i>Boolean</i>	Read/Write A flag to indicate if the method is <i>Root</i> .
<b>IsStatic</b>	<i>Boolean</i>	Read/Write A flag to indicate a static method.
<b>IsSynchronized</b>	<i>Boolean</i>	Read/Write A flag indicating a Synchronized method call.
<b>MethodGUID</b>	<i>String</i>	Read/Write A globally unique ID for the current method. This is system generated.
<b>MethodID</b>	<i>Long</i>	Read only A local ID for the current method, only valid within this .EAP file.



Attribute	Type	Notes
<b>Name</b>	<i>String</i>	Read/Write The method name.
<b>Notes</b>	<i>String</i>	Read/Write Descriptive notes on the method.
<b>ObjectType</b>	<a href="#">ObjectType</a> [2822]	Read only Distinguishes objects referenced through a Dispatch interface.
<b>Parameters</b>	<a href="#">Collection</a> [2829]	Read only The <i>Parameters</i> collection for the current method, used to add and access parameter objects for the current method.
<b>ParentID</b>	<i>Long</i>	Read only Returns the <i>ElementID</i> of the element that this method belongs to.
<b>Pos</b>	<i>Long</i>	Read/Write Specifies the position of the method within the set of operations defined for a Class.
<b>PostConditions</b>	<a href="#">Collection</a> [2829]	Read only The PostConditions (constraints) as they apply to this method. This returns a <i>MethodConstraint</i> object of type <b>post</b> .
<b>PreConditions</b>	<a href="#">Collection</a> [2829]	Read only The PreConditions (constraints) as they apply to this method. This returns a <i>MethodConstraint</i> object of type <b>pre</b> .
<b>ReturnsArray</b>	<i>Boolean</i>	Read/Write A flag to indicate that the return value is an array.
<b>ReturnType</b>	<i>String</i>	Read/Write The return type for the method; this can be a primitive data type or a Class or Interface type.
<b>StateFlags</b>	<i>String</i>	Read/Write Some flags as applied to methods in State elements.

Attribute	Type	Notes
<b>Stereotype</b>	<i>String</i>	Read/Write The method stereotype (optional).
<b>StereotypeEx</b>	<i>String</i>	Read/Write All the applied stereotypes of the method in a comma-separated list.
<b>Style</b>	<i>String</i>	Read/Write Contains the <b>Alias</b> property for this method.
<b>StyleEx</b>	<i>String</i>	Read/Write Advanced style settings, reserved for the use of Sparx Systems.
<b>TaggedValues</b>	<a href="#">Collection</a> <sup>[2829]</sup> of type <a href="#">MethodTag</a> <sup>[2924]</sup>	Read only The <i>TaggedValues</i> collection for the current method. This accesses a list of <i>MethodTag</i> objects.
<b>Throws</b>	<i>String</i>	Read/Write Exception information.
<b>Visibility</b>	<i>String</i>	Read/Write The method scope - <b>Public</b> , <b>Protected</b> , <b>Private</b> or <b>Package</b> .

**Method Methods:**

Method	Type	Notes
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.
<b>Update ()</b>	<i>Boolean</i>	Update the current method object after modification or appending a new item.  If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.

### 20.2.2.6.7 MethodConstraint Class

A *MethodConstraint* is a condition imposed on a method. It is accessed through either the Method *PreConditions* or Method *PostConditions* collection.

**Associated table in .EAP file:** *t\_operationpres* and *t\_operationposts*

#### **MethodConstraint Attributes:**

Attribute	Type	Notes
<b>MethodID</b>	<i>Long</i>	Read/Write The local ID of the associated method.
<b>Name</b>	<i>String</i>	Read/Write The name of the constraint.
<b>Notes</b>	<i>String</i>	Read/Write Descriptive notes about this constraint.
<b>ObjectType</b>	<a href="#">ObjectType</a> [2822]	Read only Distinguishes objects referenced through a Dispatch interface.
<b>Type</b>	<i>String</i>	Read/Write The constraint type.

#### **MethodConstraint Methods:**

Method	Type	Notes
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.  This function is rarely used as an exception is thrown when an error occurs.
<b>Update ()</b>	<i>Boolean</i>	Update the current <i>MethodConstraint</i> object after modification or appending a new item.  If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.

### 20.2.2.6.8 MethodTag Class

A *MethodTag* is a Tagged Value associated with a method.

**Associated table in .EAP file:** *t\_operationtag*

#### MethodTag Attributes:

Attribute	Type	Notes
<b>FQName</b>	<i>String</i>	Read only The fully-qualified name of the tag.
<b>MethodID</b>	<i>Long</i>	Read/Write The ID of the associated method.
<b>Name</b>	<i>String</i>	Read/Write The tag or name of the property.
<b>Notes</b>	<i>String</i>	Read/Write Further descriptive notes about this tag. If <b>Value</b> (below) is set to "<memo>", then <b>Notes</b> should contain the actual Tagged Value content.
<b>ObjectType</b>	<a href="#">ObjectType</a> [2822]	Read only Distinguishes objects referenced through a Dispatch interface.
<b>TagGUID</b>	<i>String</i>	Read/Write A unique GUID for this Tagged Value.
<b>TagID</b>	<i>Long</i>	Read only A unique ID for this Tagged Value.
<b>Value</b>	<i>String</i>	Read/Write The value assigned to this tag. This field has a 255 character limit. If the value is greater than 255 characters long, set the value to "<memo>" and insert the body of text in the <b>Notes</b> attribute (above). When reading existing Tagged Values, if <b>Value</b> = "<memo>" then the developer should read the actual body of text from the <b>Notes</b> attribute.

**MethodTag Methods:**

Method	Type	Notes
<b>GetAttribute</b> (string propName)	<i>String</i>	Returns the text of a single named property within a structured tagged value.
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.  This function is rarely used as an exception is thrown when an error occurs.
<b>HasAttributes()</b>	<i>Boolean</i>	Returns true if the tagged values is a structured tagged value with one or more properties.
<b>SetAttribute</b> (string propName, string propValue)	<i>Boolean</i>	Sets the text of a single named property within a structured tagged value.
<b>Update ()</b>	<i>Boolean</i>	Updates the current <i>MethodTag</i> object after modification or appending a new item.  If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.

**20.2.2.6.9 Parameter Class**

A **Parameter** object represents a method argument and is accessed through the **Method Parameters** collection.

**Associated table in .EAP file** *t\_operationparams*

**Parameter Attributes**

Attribute	Type	Notes
<b>Alias</b>	<i>String</i>	Read/Write An optional alias for this parameter.
<b>ClassifierID</b>	<i>String</i>	Read/Write A ClassifierID for the parameter, if known.
<b>Default</b>	<i>String</i>	Read/Write A default value for this parameter.

Attribute	Type	Notes
<b>IsConst</b>	<i>Boolean</i>	Read/Write A flag indicating that the parameter is <i>Const</i> (cannot be altered).
<b>Kind</b>	<i>String</i>	Read/Write The parameter kind - <b>in</b> , <b>inout</b> , <b>out</b> , or <b>return</b> .
<b>Name</b>	<i>String</i>	Read/Write The parameter name; this must be unique for a single method.
<b>Notes</b>	<i>String</i>	Read/Write Descriptive notes.
<b>ObjectType</b>	<a href="#">ObjectType</a> <small>28221</small>	Read only Distinguishes objects referenced through a Dispatch interface.
<b>OperationID</b>	<i>Long</i>	Read only The ID of the method associated with this parameter.
<b>ParameterGUID</b>	<i>String</i>	Read/Write A system generated, globally unique ID for the current Parameter.
<b>Position</b>	<i>Long</i>	Read/Write The position of the parameter in the argument list.
<b>Stereotype</b>	<i>String</i>	Read/Write The first stereotype of the parameter.
<b>StereotypeEx</b>	<i>String</i>	Read/Write All the applied stereotypes of the parameter in a comma-separated list.
<b>Style</b>	<i>String</i>	Read/Write Some style information.
<b>StyleEx</b>	<i>String</i>	Read/Write

Attribute	Type	Notes
		Advanced style settings, reserved for the use of Sparx Systems.
<b>TaggedValues</b>	<a href="#">Collection</a> of type <a href="#">ParamTag</a>	Read/Write The GUID of the parameter with which this <i>ParamTag</i> is associated.
<b>Type</b>	<i>Variant</i>	Read/Write The parameter type; can be a primitive type or a defined classifier.

#### Parameter Methods

Method	Type	Notes
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.
<b>Update ()</b>	<i>Boolean</i>	Update the current Parameter object after modifying or appending a new item.  If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.

#### 20.2.2.6.10 ParamTag Class

A **ParamTag** is a Tagged Value associated with a method parameter.

Associated table in .EAP file    *t\_taggedvalue*

#### ParamTag Attributes

Attribute	Type	Notes
<b>ElementGUID</b>	<i>String</i>	Read/Write The GUID of the parameter with which this <i>ParamTag</i> is associated.
<b>FQName</b>	<i>String</i>	Read only The fully qualified name of the tag.
<b>ObjectType</b>	<a href="#">Object Type</a>	Read only Distinguishes objects referenced through a Dispatch interface.

Attribute	Type	Notes
<b>PropertyGUID</b>	<i>String</i>	Read/Write A system generated GUID to identify the Tagged Value.
<b>Tag</b>	<i>String</i>	Read/Write The actual tag name.
<b>Value</b>	<i>String</i>	Read/Write The value associated with this tag.

#### ParamTag Methods

Method	Type	Notes
<b>GetAttribute( string propName)</b>	<i>String</i>	Returns the text of a single named property within a structured tagged value.
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.
<b>HasAttributes()</b>	<i>Boolean</i>	Returns true if the tagged values is a structured tagged value with one or more properties.
<b>SetAttribute( string propName, string propValue)</b>	<i>Boolean</i>	Sets the text of a single named property within a structured tagged value.
<b>Update ()</b>	<i>Boolean</i>	Updates the current ParamTag object after modifying or appending a new item.  If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.

#### 20.2.2.6.11 Partitions Collection

A collection of internal element partitions (regions). This is commonly seen in Activity, State, Boundary, Diagram Frame and similar elements. Not all elements support partitions.

This collection contains a set of *Partition* elements. The set is read/write: information is not saved until the host element is saved, so ensure that you call the *Element.Save* method after making changes to a Partition.



**Partition Attributes**

Attribute	Type	Notes
<b>Name</b>	<i>String</i>	Read/Write The partition name; this can represent a condition or constraint in some cases.
<b>Note</b>	<i>String</i>	Read/Write A free text note associated with this partition.
<b>ObjectType</b>	<a href="#">ObjectType</a> [2822]	Read only Distinguishes objects referenced through a Dispatch interface.
<b>Operator</b>	<i>String</i>	Read/Write An optional operator value that specifies the partition type.
<b>Size</b>	<i>String</i>	Read/Write The vertical or horizontal width of the partition in pixels.

**Learn more**

- [Activity](#) [1279]
- [State](#) [1329]
- [Boundary](#) [1997]
- [Diagram Frame](#) [1296]

**20.2.2.6.12 Properties Class****Properties****Properties Attributes**

Attribute	Type	Notes
<b>Count</b>	<i>Long</i>	The number of properties that are available for this object.
<b>ObjectType</b>	<a href="#">ObjectType</a> [2822]	Read only Distinguishes objects referenced through a Dispatch interface.

**Properties Methods**

Method	Type	Notes
<b>Item (object Index)</b>	<i>Property</i>	<p>Returns a property either by name or by a zero-based integer offset into the list of properties.</p> <p>Parameter:</p> <ul style="list-style-type: none"> <li>Index: Variant - either a string representing the property name or an integer representing the zero-based offset into the property list</li> </ul>

**Property****Property Attributes**


Attribute	Type	Notes
<b>Name</b>	<i>String</i>	<p>Read only</p> <p>The name of the property.</p> <p>The object to which the properties list applies can have an automation property with the same name, in which case the data accessed through <b>Value</b> is identical to that obtained through the automation property.</p>
<b>ObjectType</b>	<a href="#">ObjectType</a> [2822]	<p>Read only</p> <p>Distinguishes objects referenced through a Dispatch interface.</p>
<b>Type</b>	<a href="#">PropType</a> [2823]	<p>Read only</p> <p>Provides an indication of what sort of data is going to be stored by this property. This restriction can be further defined by the <i>Validation</i> attribute.</p>
<b>Validation</b>	<i>String</i>	<p>Read only</p> <p>An optional string that is used to validate any data that is passed to the <i>Value</i> attribute. This string is used by the programmer at run time to provide an indication of what is expected, and by Enterprise Architect to ensure that the submitted data is appropriate.</p>
<b>Value</b>	<i>Variant</i>	<p>Read/write</p> <p>The value of the property as defined in the other fields.</p>

### 20.2.2.6.13 TemplateParameter Class

A **TemplateParameter** for a template signature specifies a formal parameter that will be substituted by an actual parameter (or the default) in a TemplateBinding relationship on a Class element.

**Associated table in .EAP file** *t\_xref*

#### TemplateParameter Attributes

Attribute	Type	Notes
<b>Constraint</b>	<i>String</i>	Read/Write The name of the Classifier that acts as the constraint value.
<b>Default</b>	<i>String</i>	Read/Write The name of the Classifier that acts as the default value.
<b>Name</b>	<i>String</i>	Read/Write The name of the Template Parameter.
<b>ObjectType</b>	<a href="#">Object Type</a> 	Read Only Distinguishes objects referenced through a Dispatch interface.
<b>TemplateParameterID</b>	<i>String</i>	Read Only The Enterprise Architect Globally Unique ID (GUID) of the current Template Parameter, in the <b>XrefID</b> column of <i>t_xref</i> .
<b>Type</b>	<i>String</i>	Read/Write The Template Parameter type.

#### TemplateParameter Methods

Method	Type	Notes
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.
<b>Update ()</b>	<i>Boolean</i>	Updates the current TemplateParameter object after modifying or appending a new item.  If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.

Learn more

- [Parameterized Classes \(Templates\)](#) <sup>[1365]</sup>
- [Template Binding](#) <sup>[1444]</sup>
- [TemplateBinding Class](#) <sup>[2946]</sup>

**20.2.2.6.14 Transitions Collection**

The Transitions collection applies only to *Timeline elements*.

A Timeline element displays **0** or more state transitions at set times on its extent. This collection enables you to access the transition set. You can also access additional information by referring to the connectors associated with the Timeline, and by referencing messages passed between timelines. Note that any changes made to elements in this collection are only saved when the main element is saved.

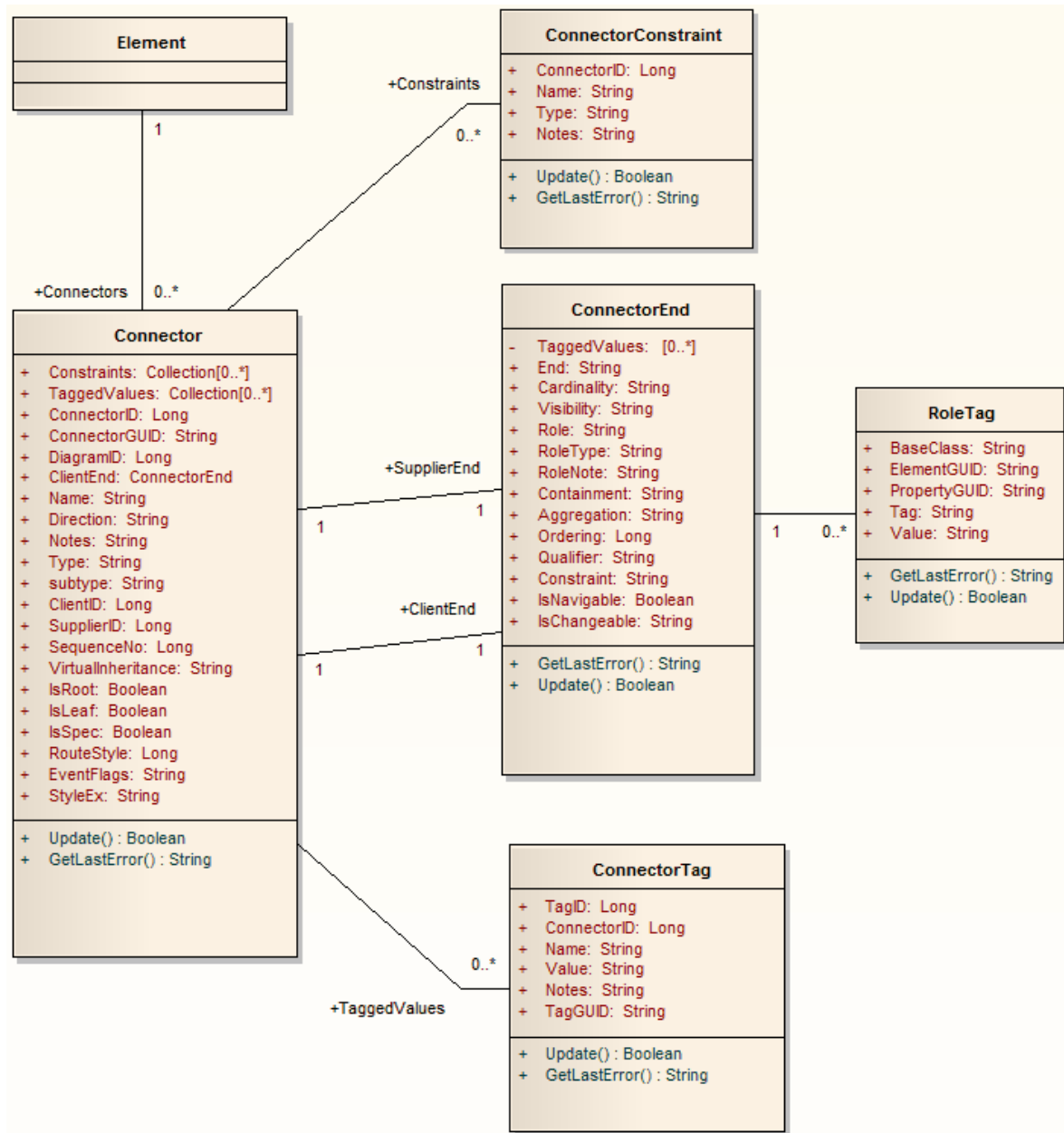
Transition Attributes

Attribute	Type	Notes
<b>DurationConstraint</b>	<i>String</i>	Read/Write A constraint on the time duration of the transition.
<b>Event</b>	<i>String</i>	Read/Write The event (optional) that initiated the transition.
<b>Note</b>	<i>String</i>	Read/Write A free text note.
<b>ObjectType</b>	<a href="#">ObjectType</a> <sup>[2822]</sup>	Read only Distinguishes objects referenced through a Dispatch interface.
<b>TimeConstraint</b>	<i>String</i>	Read/Write A constraint on when the transition has to be completed.
<b>TxState</b>	<i>String</i>	Read/Writ The state to transition to, as defined in the Timeline Properties dialog.
<b>TxTime</b>	<i>String</i>	Read/Write. The time that the transition occurs. The value depends on a range set in the diagram.

### 20.2.2.7 Connector Package

The **Connector** Package details how connectors between elements are accessed and managed.

#### Example



#### 20.2.2.7.1 Connector Class

To represent the various kinds of connectors between UML elements, you use a **Connector** object. You can access this from either the **Client** or **Supplier** element, using the **Connectors** collection of that element. When creating a new connector you assign to it a valid type from the following list:

- Aggregation
- Assembly
- Association

- Collaboration
- CommunicationPath
- Connector
- ControlFlow
- Delegate
- Dependency
- Deployment
- ERLink
- Generalization
- InformationFlow
- Instantiation
- InterruptFlow
- Manifest
- Nesting
- NoteLink
- ObjectFlow
- Package
- Realization
- Sequence
- StateFlow
- TemplateBinding
- UseCase

**Associated table in .EAP file** *t\_connector*

#### **Connector Attributes**

Attribute	Type	Notes	See also
<b>Alias</b>	<i>String</i>	Read/Write An optional alias for this connector.	
<b>ClientEnd</b>	<i>ConnectorEnd</i>	Read only A pointer to the <i>ConnectorEnd</i> object representing the source end of the relationship.	<a href="#">ConnectorEnd Class</a> <sup>[2946]</sup>
<b>ClientID</b>	<i>Long</i>	Read/Write The <i>ElementID</i> of the element at the source end of this connector.	
<b>Color</b>	<i>Long</i>	Read/Write	

Attribute	Type	Notes	See also
		Sets the color of the connector.	
<b>ConnectorGUID</b>	<i>Variant</i>	Read only A system generated, globally unique ID for the current connector.	
<b>ConnectorID</b>	<i>Long</i>	Read only A system generated local identifier for the current connector.	
<b>Constraints</b>	<i>Collection</i>	Read only A collection of <b>constraint</b> objects.	<a href="#">Constraint Class</a> <sup>[2879]</sup> <a href="#">Collection Class</a> <sup>[2829]</sup>
<b>ConveyedItems</b>	<i>Collection</i>	Read only Returns a collection of elements that have been conveyed.  To add another element to the conveyed Collection, use <i>AddNew (ElementGUID,NULL)</i> ; where <i>ElementGUID</i> is the GUID of the element to be added.	<a href="#">Collection Class</a> <sup>[2829]</sup>
<b>CustomProperties</b>	<i>Collection</i>	Read only Returns a collection of advanced properties associated with an element in the form of <b>CustomProperty</b> objects.	<a href="#">Collection Class</a> <sup>[2829]</sup> <a href="#">CustomProperties Collection</a> <sup>[2918]</sup>
<b>DiagramID</b>	<i>Long</i>	Read/Write The <i>DiagramID</i> of the connector.	
<b>Direction</b>	<i>String</i>	Read/Write  The connector direction, which can be set to one of the following: <ul style="list-style-type: none"> <li>• <b>Unspecified</b></li> <li>• <b>Bi-Directional</b></li> <li>• <b>Source -&gt; Destination</b></li> <li>• <b>Destination -&gt; Source</b></li> </ul> If the connector is non-navigable, set the <b>sourceNavigability</b> and/or <b>targetNavigability</b> attributes.	<a href="#">Special Attributes</a> <sup>[1503]</sup>
<b>EndPointX</b>	<i>Long</i>	Read/Write	

Attribute	Type	Notes	See also
		The x-coordinate of the connector's end point. Connector end points are specified in Cartesian coordinates with the origin to the top left of the screen.	
<b>EndPointY</b>	<i>Long</i>	Read/Write The y-coordinate of the connector's end point. Connector end points are specified in Cartesian coordinates with the origin to the top left of the screen.	
<b>EventFlags</b>	<i>String</i>	Read/Write A structure to hold a variety of flags concerned with event signaling on messages.	
<b>IsLeaf</b>	<i>Boolean</i>	Read/Write A flag indicating that the connector is a <i>leaf</i> .	
<b>IsRoot</b>	<i>Boolean</i>	Read/Write A flag indicating that the connector is a <i>root</i> .	
<b>IsSpec</b>	<i>Boolean</i>	Read/Write. A flag indicating that the connector is a specification.	
<b>MetaType</b>	<i>String</i>	Read only The connector's domain-specific meta type, as defined by an applied stereotype from an MDG Technology	
<b>MiscData</b>	<i>String</i>	Read only This low-level property returns an array providing information about the contents of the <b>PData</b> x fields; these database fields are not documented and developers must gain understanding of these fields through their own endeavors to use this property. <b>MiscData</b> is zero based, therefore: <ul style="list-style-type: none"> <li>• <b>MiscData(0)</b> corresponds to <b>PData1</b></li> <li>• <b>MiscData(1)</b> corresponds to <b>PData2</b>, and so on</li> </ul>	
<b>Name</b>	<i>String</i>	Read/Write The connector name.	



Attribute	Type	Notes	See also
<b>Notes</b>	<i>String</i>	Read/Write Descriptive notes about the connector.	
<b>ObjectType</b>	<i>ObjectType</i>	Read only Distinguishes objects referenced through a Dispatch interface.	<a href="#">ObjectType</a> <sup>[2822]</sup>
<b>Properties</b>	<i>Properties</i>	Returns a list of specialized properties applicable to the connector that might not be available using the automation model.  The properties are purposely undocumented because of their obscure nature and because they are subject to change as progressive enhancements are made to them.	<a href="#">Properties Class</a> <sup>[2925]</sup>
<b>RouteStyle</b>	<i>Long</i>	Read/Write The route style.	
<b>SequenceNo</b>	<i>Long</i>	Read/Write The <i>SequenceNo</i> of the connector.	
<b>StartPointX</b>	<i>Long</i>	Read/Write The x-coordinate of the connector's start point. Connector end points are specified in Cartesian coordinates with the origin to the top left of the screen.	
<b>StartPointY</b>	<i>Long</i>	Read/Write The y-coordinate of the connector's start point. Connector end points are specified in Cartesian coordinates with the origin to the top left of the screen.	
<b>StateFlags</b>	<i>String</i>	Read/Write A structure to hold a variety of flags concerned with State signaling on messages; the list is delimited by semi-colons.	
<b>Stereotype</b>	<i>String</i>	Read/Write Sets or gets the stereotype for this connector end.	

Attribute	Type	Notes	See also
<b>StereotypeEx</b>	<i>String</i>	Read/Write All the applied stereotypes of the connector in a comma-separated list.	
<b>StyleEx</b>	<i>String</i>	Read/Write Advanced style settings; reserved for the use of Sparx Systems.	
<b>Subtype</b>	<i>String</i>	Read/Write A possible subtype to refine the meaning of the connector.	
<b>SupplierEnd</b>	<i>ConnectorEnd</i>	Read only A pointer to the <i>ConnectorEnd</i> object representing the target end of the relationship.	<a href="#">ConnectorEnd Class</a> <small>[2940]</small>
<b>SupplierID</b>	<i>Long</i>	Read/Write The <i>ElementID</i> of the element at the target end of this connector.	
<b>TaggedValues</b>	<i>Collection of type ConnectorTag</i>	Read only The collection of <i>ConnectorTag</i> objects.	<a href="#">Collection Class</a> <small>[2829]</small> <a href="#">ConnectorTag Class</a> <small>[2943]</small>
<b>TemplateBindings</b>	<i>Collection of type TemplateBinding</i>	Read only A collection of <i>TemplateBinding</i> objects.	<a href="#">Collection Class</a> <small>[2829]</small> <a href="#">TemplateBinding Class</a> <small>[2946]</small>
<b>TransitionAction</b>	<i>String</i>	Read/Write See the <i>Transition</i> topic for appropriate values.	<a href="#">Transition</a> <small>[1446]</small>
<b>TransitionEvent</b>	<i>String</i>	Read/Write See the <i>Transition</i> topic for appropriate values.	
<b>TransitionGuard</b>	<i>String</i>	Read/Write See the <i>Transition</i> topic for appropriate values.	

Attribute	Type	Notes	See also
<b>Type</b>	<i>String</i>	Read/Write The connector type; valid types are held in the <i>t_connectortypes</i> table in the .EAP file.	
<b>VirtualInheritance</b>	<i>String</i>	Read/Write For <i>Generalization</i> , indicates if the inheritance is virtual.	
<b>Width</b>	<i>Long</i>	Read/Write Specifies the width of the connector.	

#### Connector Methods

Method	Type	Notes
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.
<b>Update ()</b>	<i>Boolean</i>	Update the current <i>ConnectorObject</i> after modification or appending a new item.  If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.

#### **20.2.2.7.2 ConnectorConstraint Class**

A **ConnectorConstraint** holds information about special conditions that apply to a connector. It is accessed through the Connector **Constraints** collection.

Associated table in .EAP file *t\_connectorconstraints*

#### ConnectorConstraint Attributes

Attribute	Type	Notes	See also
<b>ConnectorID</b>	<i>Long</i>	Read/Write A local ID value (long) - system generated.	
<b>Name</b>	<i>String</i>	Read/Write The constraint name.	

Attribute	Type	Notes	See also
<b>Notes</b>	<i>String</i>	Read/Write Notes about this constraint.	
<b>ObjectType</b>	<i>ObjectType</i>	Read only Distinguishes objects referenced through a Dispatch interface.	<a href="#">ObjectType</a> [2822]
<b>Type</b>	<i>String</i>	Read/Write The constraint type.	

#### ConnectorConstraint Methods

Method	Type	Notes
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.
<b>Update ()</b>	<i>Boolean</i>	Update the current <i>ConnectorConstraint</i> object after modification or appending a new item.  If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.

#### 20.2.2.7.3 *ConnectorEnd Class*

A **ConnectorEnd** contains information about a single end of a connector. A **ConnectorEnd** is accessed from the connector as either the **ClientEnd** or **SupplierEnd**.

Associated table in .EAP file    *derived from t\_connector*

#### ConnectorEnd Attributes

Attribute	Type	Notes	See also
<b>Aggregation</b>	<i>Long</i>	Read/Write The type of Aggregation as it applies to this end; valid values are:  <b>0</b> = None <b>1</b> = Shared <b>2</b> = Composite	
<b>Alias</b>	<i>String</i>	Read/Write An optional alias for this connector end.	

Attribute	Type	Notes	See also
<b>AllowDuplicates</b>	<i>Boolean</i>	Read/Write For multiplicities greater than <b>1</b> , indicates that duplicate entries are possible.	
<b>Cardinality</b>	<i>String</i>	Read/Write The cardinality associated with this end.	
<b>Constraint</b>	<i>String</i>	Read/Write A constraint that can be applied to this connector end.	
<b>Containment</b>	<i>String</i>	Read/Write The containment type applied to this connector end.	
<b>Derived</b>	<i>Boolean</i>	Read/Write Indicates that the value of this end is derived.	
<b>DerivedUnion</b>	<i>Boolean</i>	Read/Write Indicates the value of this role derived from the union of all roles that subset this.	
<b>End</b>	<i>String</i>	Read only The end this <i>ConnectorEnd</i> object applies to - <b>Client</b> or <b>Supplier</b> .	
<b>IsChangeable</b>	<i>String</i>	Read/Write Flag indicating whether this end is changeable or not - <b>frozen</b> , <b>addOnly</b> or <b>none</b> .	
<b>IsNavigable</b>	<i>Boolean</i>	Read/Write A flag indicating this end is navigable from the other end.	
<b>Navigable</b>	<i>String</i>	Read/Write Indicates whether this role of an association is navigable from the opposite classifier - <b>Navigable</b> , <b>Non-Navigable</b> or <b>Unspecified</b> .	
<b>ObjectType</b>	<i>ObjectTy</i>	Read only	<a href="#">ObjectType</a>

Attribute	Type	Notes	See also
	<i>pe</i>	Distinguishes objects referenced through a Dispatch interface.	<a href="#">[2822]</a>
<b>Ordering</b>	<i>Long</i>	Read/Write Ordering for this connector end.	
<b>OwnedByClassifier</b>	<i>Boolean</i>	Read/Write Indicates that this Association end corresponds to an attribute on the opposite end of the Association.	
<b>Qualifier</b>	<i>String</i>	Read/Write A qualifier that can apply to the connector end.	
<b>Role</b>	<i>String</i>	Read/Write The connector end role.	
<b>RoleNote</b>	<i>String</i>	Read/Write Notes associated with the role of this connector end.	
<b>RoleType</b>	<i>String</i>	Read/Write The role type applied to this end of the connector.	
<b>Stereotype</b>	<i>String</i>	Read/Write Sets or gets the stereotype for this connector end.	
<b>StereotypeEx</b>	<i>String</i>	Read/Write All the applied stereotypes of the connector end in a comma-separated list.	
<b>TaggedValues</b>	<i>Private</i>	Read only A collection of <i>RoleTag</i> objects.	<a href="#">RoleTag Class</a> <a href="#">[2944]</a>
<b>Visibility</b>	<i>String</i>	Read/Write The Scope associated with this connector end - <b>Public</b> , <b>Private</b> , <b>Protected</b> or <b>Package</b> .	

#### ConnectorEnd Methods

Method	Type	Notes
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.
<b>Update ()</b>	<i>Boolean</i>	Update the current <i>ConnectorEnd</i> object after modification or appending a new item.  If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.

#### 20.2.2.7.4 ConnectorTag Class

A **ConnectorTag** is a Tagged Value for a connector and is accessed through the Connector **TaggedValues** collection.

Associated table in .EAP file *t\_connectortag*

#### ConnectorTag Attributes

Attribute	Type	Notes	See also
<b>ConnectorID</b>	<i>Long</i>	Read/Write The local ID of the associated connector.	
<b>FQName</b>	<i>String</i>	Read only The fully qualified name of the tag.	
<b>Name</b>	<i>String</i>	Read/Write The tag or name.	
<b>Notes</b>	<i>String</i>	Read/Write Further descriptive notes on this tag.  If <b>Value</b> (below) is set to "<memo>", then <b>Notes</b> should contain the actual Tagged Value content.	
<b>ObjectType</b>	<i>ObjectType</i>	Read only Distinguishes objects referenced through a Dispatch interface.	<a href="#">ObjectType</a> e <sub>[2822]</sub>
<b>TagGUID</b>	<i>String</i>	Read/Write A globally unique ID for this Tagged Value.	
<b>TagID</b>	<i>Long</i>	Read only	

Attribute	Type	Notes	See also
		A local ID to identify the Tagged Value.	
<b>Value</b>	<i>String</i>	<p>Read/Write</p> <p>The value assigned to this tag.</p> <p>This field has a 255 character limit. If the value is greater than 255 characters long, set the value to "&lt;memo&gt;" and insert the body of text in the <b>Notes</b> attribute (above).</p> <p>When reading existing Tagged Values, if <b>Value</b> = "&lt;memo&gt;" then the developer should read the actual body of text from the <b>Notes</b> attribute.</p>	

#### ConnectorTag Methods

Method	Type	Notes
<b>GetAttribute ( string propName)</b>	<i>String</i>	Returns the text of a single named property within a Structured Tagged Value.
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.
<b>HasAttributes ()</b>	<i>Boolean</i>	Returns true if the Tagged Value is a Structured Tagged Value with one or more properties.
<b>SetAttribute ( string propName, string propValue)</b>	<i>Boolean</i>	Sets the text of a single named property within a Structured Tagged Value.
<b>Update ()</b>	<i>Boolean</i>	<p>Update the current <i>ConnectorTag</i> object after modification or appending a new item.</p> <p>If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.</p>

#### 20.2.2.7.5 RoleTag Class

The **RoleTag** interface provides access to an Association's Role Tagged Values. Each connector end has a RoleTag collection that can be accessed to add, delete and access the RoleTags.

You might use this in creating code that resembles this fragment for accessing a RoleTag in VB.NET (where *con* is a Connector Object):

```
client = con.ClientEnd
```



```

client.Role = "m_client"
client.Update()
tag = client.TaggedValues.AddNew("tag", "value")
tag.Update()
tag = client.TaggedValues.AddNew("tag2", "value2")
tag.Update()
client.TaggedValues.Refresh()
For idx = 0 To client.TaggedValues.Count - 1
    tag = client.TaggedValues.GetAt(idx)
    Console.WriteLine(tag.Tag)
    client.TaggedValues.DeleteAt(idx, False)
Next
tag = Nothing

```

Associated table in .EAP file    *t\_taggedvalue*

### RoleTag Attributes

Attribute	Type	Notes	See also
<b>BaseClass</b>	<i>String</i>	Read/Write Indicates the role end; set to <b>ASSOCIATION_SOURCE</b> or <b>ASSOCIATION_TARGET</b> .	
<b>ElementGUID</b>	<i>String</i>	Read/Write The GUID of the connector with which this role tag is associated.	
<b>FQName</b>	<i>String</i>	Read only The fully qualified name of the tag.	
<b>ObjectType</b>	<i>ObjectType</i>	Read only Distinguishes objects referenced through a Dispatch interface.	<a href="#">ObjectType</a> e <sub>2822</sub>
<b>PropertyGUID</b>	<i>String</i>	Read/Write A system generated GUID to identify the Tagged Value.	
<b>Tag</b>	<i>String</i>	Read/Write The actual tag name.	
<b>Value</b>	<i>String</i>	Read/Write The value associated with this tag.	

### RoleTag Methods

Method	Type	Notes
<b>GetAttribute (string propName)</b>	<i>String</i>	Returns the text of a single named property within a Structured Tagged Value.
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.
<b>HasAttributes ()</b>	<i>Boolean</i>	Returns true if the Tagged Value is a Structured Tagged Value with one or more properties.
<b>SetAttribute (string propName, string propValue)</b>	<i>Boolean</i>	Sets the text of a single named property within a Structured Tagged Value.
<b>Update ()</b>	<i>Boolean</i>	Update the RoleTag after changes or on initial creation.  If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.

#### 20.2.2.7.6 TemplateBinding Class

A **TemplateBinding** defines the connector between a binding Class and a parameterized Class, and the binding expression on that connector.

##### TemplateBinding Attributes

Attribute	Type	Notes	See also
<b>ActualGUID</b>	<i>String</i>	Read/Write  The GUID of the element classifier set as the <i>Actual</i> Template Binding parameter.  If the <i>Actual</i> Template Binding parameter is set as a string expression only, this will be an empty string.  Assigning a GUID value will automatically change the <b>ActualName</b> attribute after <b>Update()</b> has been called.	
<b>ActualName</b>	<i>String</i>	Read/Write  The name of the <i>Actual</i> Template Binding parameter.  Assigning a new value will clear any current <b>ActualGUID</b> value.	
<b>BindingExpression</b>	<i>String</i>	Read only  The Binding Expression as shown in Enterprise Architect.	

Attribute	Type	Notes	See also
<b>ConnectorGUID</b>	<i>String</i>	Read only The Globally Unique ID of the associated connector.	
<b>ConnectorType</b>	<i>String</i>	Read only The type of the associated connector.	
<b>FormalName</b>	<i>String</i>	Read/Write The name of the <i>Formal</i> Template Binding parameter.	
<b>ObjectType</b>	<i>ObjectType</i>	Read only Distinguishes objects referenced through a Dispatch Interface.	<a href="#">ObjectType</a> [2822]
<b>Pos</b>	<i>String</i>	Read only The position of the Template Binding in the list (as on the Bindings page of the connector Properties dialog).	
<b>TemplateBindingID</b>	<i>String</i>	Read only The Globally Unique ID of the current Template Binding.	

#### TemplateBinding Methods

Method	Type	Notes
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.
<b>Update ()</b>	<i>Boolean</i>	Update the current <b>TemplateBinding</b> object after modification or appending a new item.  If <b>false</b> is returned, check the <b>GetLastError</b> function for more information.

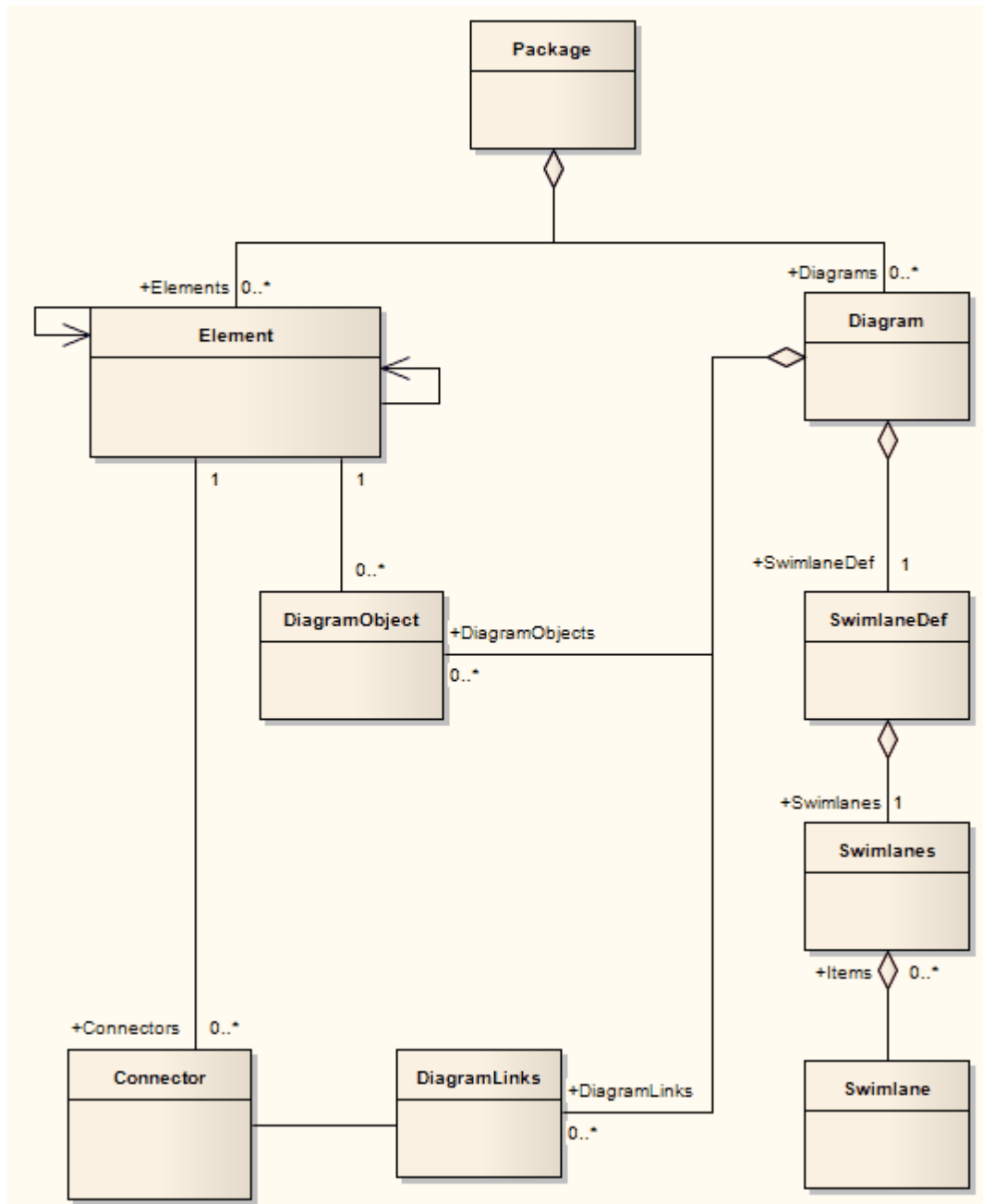
#### Learn more

- [Template Binding](#) [1444]

### 20.2.2.8 Diagram Package

The *Diagram* package has information on a diagram and on *DiagramObject* and *DiagramLink*, which are the instances of elements within a diagram.

#### Example



#### 20.2.2.8.1 Diagram Class

A **Diagram** corresponds to a single UML diagram. It is accessed through the **Package Diagrams** collection and in turn contains a collection of diagram objects and diagram connectors. Adding to the *DiagramObject* Class adds an existing element to the diagram. When adding a new diagram, you must set the diagram type to one of the valid types:

- Activity
- Analysis
- Component
- Custom
- Deployment
- Logical
- Sequence
- Statechart
- Use Case

For a Collaboration (Communication) diagram, use the Analysis type.

**Associated table in .EAP file** `t_diagram`

#### Diagram Attributes

Attribute	Type	Notes	See also
<b>Author</b>	<i>String</i>	Read/Write The name of the author.	
<b>CreatedDate</b>	<i>Date</i>	Read/Write The date the diagram was created.	
<b>cx</b>	<i>Long</i>	Read/Write The X dimension of the diagram (the default is <b>800</b> ).	
<b>cy</b>	<i>Long</i>	Read/Write The Y dimension of the diagram (the default is <b>1100</b> ).	
<b>DiagramGUID</b>	<i>Variant</i>	Read/Write A globally unique ID for this diagram.	
<b>DiagramID</b>	<i>Long</i>	Read only A local ID for the diagram.	
<b>DiagramLinks</b>	<a href="#">Collection</a> <small>[2829]</small>	Read only A list of <b>DiagramLink</b> objects, each containing information about the display characteristics of a connector in a diagram.	<a href="#">DiagramLink Class</a> <small>[2955]</small>

Attribute	Type	Notes	See also
<b>DiagramObjects</b>	<a href="#">Collection</a> <small>[2828]</small>	Read only  A collection of references to <b>DiagramObjects</b> . A <i>DiagramObject</i> is an instance of an element in a diagram, and includes size and display characteristics.	<a href="#">DiagramObject Class</a> <small>[2956]</small>
<b>ExtendedStyle</b>	<i>String</i>	Read/Write  An extended style attribute.	
<b>HighlightImports</b>	<i>Boolean</i>	Read/Write  A flag to indicate that elements from other packages should be highlighted. Corresponds with the 'Show Namespace' option in the diagram properties dialog.	<a href="#">Configure Diagram Display</a> <small>[825]</small>
<b>IsLocked</b>	<i>Boolean</i>	Read/Write  A flag indicating whether this diagram is locked or not.	
<b>MetaType</b>	<i>String</i>	Read only  The diagram's domain-specific meta type, as defined by an MDG Technology.	
<b>ModifiedDate</b>	<i>Variant</i>	Read/Write  The date the diagram was last modified.	
<b>Name</b>	<i>String</i>	Read/Write  The diagram name.	
<b>Notes</b>	<i>String</i>	Read/Write  Set or retrieve notes for this diagram.	
<b>ObjectType</b>	<a href="#">ObjectType</a> <small>[2822]</small>	Read only  Distinguishes objects referenced through a Dispatch interface.	
<b>Orientation</b>	<i>String</i>	Read/Write  The page orientation: <b>P</b> for Portrait or <b>L</b> for Landscape.	
<b>PackageID</b>	<i>Long</i>	Read/Write	

Attribute	Type	Notes	See also
		The ID of the Package that this diagram belongs to.	
<b>PageHeight</b>	<i>Long</i>	Read The number of pages high the diagram is.	
<b>PageWidth</b>	<i>Long</i>	Read The number of pages wide the diagram is.	
<b>ParentID</b>	<i>Long</i>	Read/Write The optional ID of an element that 'owns' this diagram; for example, a Sequence diagram owned by a Use Case.	
<b>Scale</b>	<i>Long</i>	Read/Write The zoom scale (the default is <b>100</b> ).	
<b>SelectedConnector</b>	<a href="#">Connector</a> [2933]	Read/Write The currently selected connector on this diagram. Null if there is no currently selected diagram.	
<b>SelectedObjects</b>	<a href="#">Collection</a> [2829]	Read only Gets a collection representing the currently selected elements on the diagram.  You can remove objects from this collection to deselect them, and add elements to the collection by passing the Object ID as a name to select them.	
<b>ShowDetails</b>	<i>Long</i>	Read/Write A flag to indicate that the Diagram Details text should be shown: <b>1</b> = Show, <b>0</b> = Hide.	
<b>ShowPackageContents</b>	<i>Boolean</i>	Read/Write A flag to indicate that the package contents should be shown in the current diagram.	
<b>ShowPrivate</b>	<i>Boolean</i>	Read/Write A flag to show or hide Private features.	
<b>ShowProtected</b>	<i>Boolean</i>	Read/Write	

Attribute	Type	Notes	See also
		A flag to show or hide Protected features.	
<b>ShowPublic</b>	<i>Boolean</i>	Read/Write A flag to show or hide Public features.	
<b>Stereotype</b>	<i>String</i>	Read/Write Sets or gets the stereotype for this diagram.	
<b>StyleEx</b>	<i>String</i>	Read/Write Advanced style settings, reserved for the use of Sparx Systems.	
<b>Swimlanes</b>	<i>String</i>	Read/Write Information on swimlanes contained in the diagram. Please note that this property is superseded by <b>SwimlaneDef</b> .	<a href="#">SwimlaneDef Class</a> <small>[2958]</small>
<b>SwimlaneDef</b>	<a href="#">SwimlaneDef</a> <small>[2958]</small>	Read/Write Information on swimlanes contained in the diagram.	
<b>Type</b>	<i>String</i>	Read only The diagram type; see the <i>t_diagramtypes</i> table in the .EAP file for more information.	
<b>Version</b>	<i>String</i>	Read/Write The version of the diagram.	

#### Diagram Methods

Method	Type	Notes	See also
<b>ApplyGroupLock (string aGroupName)</b>	<i>Boolean</i>	Applies a group lock to this diagram object, for the specified group, on behalf of the current user.  Returns <b>true</b> if the operation is successful; returns <b>false</b> if the operation is unsuccessful. Use <b>GetLastError()</b> to retrieve error information.  Parameter:	



Method	Type	Notes	See also
		<ul style="list-style-type: none"> <li>aGroupName: String - the name of the user group for which to set the group lock</li> </ul>	
<b>ApplyUserLock ()</b>	<i>Boolean</i>	<p>Applies a user lock to this diagram object, for the current user.</p> <p>Returns <b>true</b> if the operation is successful; returns <b>false</b> if the operation is unsuccessful. Use <b>GetLastError()</b> to retrieve error information.</p>	
<b>GetDiagramObject ByID ( long ID, string DUID)</b>	<i>DiagramObject</i>	<p>Returns the DiagramObject object, if it exists on the diagram.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>ID: Long - the ElementID of the diagram object</li> <li>DUID: String - the optional Diagram Unique ID of the diagram object.</li> </ul>	<a href="#">DiagramObject Class</a> [2956]
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.	
<b>ReadStyle ( string StyleName)</b>	<i>String</i>	<p>Returns the current value of the named diagram style.</p> <p>Use <b>GetLastError()</b> to retrieve error information.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>StyleName: String - the name of the diagram style whose value is to be retrieved; valid StyleNames are: <ul style="list-style-type: none"> <li>Show Element Property String</li> <li>Show Connector Property String</li> <li>Show Feature Property String</li> </ul> </li> </ul>	
<b>ReleaseUserLock ()</b>	<i>Boolean</i>	<p>Releases a group lock or user lock on this diagram object.</p> <p>Returns <b>true</b> if the operation is successful; returns <b>false</b> if the operation is unsuccessful. Use <b>GetLastError()</b> to retrieve error information.</p>	
<b>ReorderMessages ()</b>	<i>Void</i>	<p>Resets the display order of Sequence and Collaboration messages.</p> <p>This is typically used after inserting or deleting messages in the diagram.</p>	
<b>SaveAsPDF ( string FileName)</b>	<i>Boolean</i>	<p>Export the diagram to a PDF document. Returns <b>true</b> on success.</p> <p>Parameters:</p>	

Method	Type	Notes	See also
		<ul style="list-style-type: none"> <li>FileName: String - full path to file location</li> </ul>	
<b>SaveImagePage (longx, longy, longSizeX, long sizeY, string filename, long flags)</b>	<i>Boolean</i>	<p>Saves a page of the diagram to disk.</p> <p>Returns <b>true</b> if the operation is successful; returns <b>false</b> if the operation is unsuccessful.</p> <p>Use <b>GetLastError()</b> to retrieve error information.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>x: Long - the horizontal page</li> <li>y: Long - the vertical page</li> <li>sizeX: Long - the page width to use</li> <li>sizeY: Long - the page height to use</li> <li>filename: String - the filename to save the image to.</li> <li>flags: Long - options</li> </ul> <p>The image type is determined by the extension of the filename. Currently only .EMF, .BMP and .PNG formats are supported.</p>	
<b>ShowAsElementList (bool ShowAsList, bool Persist)</b>	<i>Boolean</i>	<p>Toggles the diagram display between diagram format and Diagram List depending on the value of <i>ShowAsList</i>.</p> <p>If <i>Persist</i> is set, the display format is written to the database so the diagram always opens in that format (diagram or list). Otherwise, the display format falls back to the default (diagram) once the display is closed.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>ShowAsList: Boolean - indicates diagram or Diagram List</li> <li>Persist: Boolean - indicates set (maintain <i>ShowAsList</i> value) or not (revert to default)</li> </ul>	<a href="#">Diagram List</a> <sup>[684]</sup>
<b>Update ()</b>	<i>Boolean</i>	<p>Updates this diagram object after modification or appending a new item.</p> <p>If <b>false</b> is returned, use <b>GetLastError()</b> to retrieve error information.</p>	
<b>WriteStyle (string StyleName, string StyleValue)</b>	<i>Void</i>	<p>Sets the value of the named diagram style.</p> <p>Use <b>GetLastError()</b> to retrieve error information.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>StyleName: String - the name of the diagram style whose value is to be retrieved; valid StyleNames are:</li> </ul>	

Method	Type	Notes	See also
		<ul style="list-style-type: none"> <li>• Show Element Property String</li> <li>• Show Connector Property String</li> <li>• Show Feature Property String</li> <li>• StyleValue: String - the value to be set in the named diagram style; valid values for the StyleNames listed above are <b>0</b> and <b>1</b>.</li> </ul>	

#### 20.2.2.8.2 DiagramLinks Class

A *DiagramLink* is an object that holds display information about a connector between two elements in a specific diagram. It includes, for example, the custom points and display appearance. It can be accessed from the Diagram **DiagramLinks** collection.

**Associated table in .EAP file:** *t\_diagramlinks*

#### DiagramLinks Attributes:

Attribute	Type	Notes
<b>ConnectorID</b>	<i>Long</i>	Read/Write The ID of the associated connector.
<b>DiagramID</b>	<i>Long</i>	Read/Write The local ID for the associated diagram.
<b>Geometry</b>	<i>String</i>	Read/Write The geometry associated with the current connector in this diagram.
<b>InstanceID</b>	<i>Long</i>	Read only The connector identifier for the current model.
<b>IsHidden</b>	<i>Boolean</i>	Read/Write A flag to indicate if this item is hidden or not.
<b>ObjectType</b>	<a href="#">ObjectType</a> [2822]	Read only Distinguishes objects referenced through a Dispatch interface.
<b>Path</b>	<i>String</i>	Read/Write

Attribute	Type	Notes
		The path of the connector in this diagram.
<b>Style</b>	<i>String</i>	Read/Write Additional style information; for example, color or thickness.

**DiagramLinks Methods:**

Method	Type	Notes
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.  This function is rarely used as an exception is thrown when an error occurs.
<b>Update ()</b>	<i>Boolean</i>	Update the current <i>DiagramLink</i> object after modification or appending a new item.  If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.

**Learn more**

- [DiagramLinks](#)  <sup>2949</sup>

**20.2.2.8.3 DiagramObject Class**

The *DiagramObject* Class stores presentation information that indicates what is displayed in a diagram and how it is shown.

**Associated table in .EAP file:** *t\_diagramobjects*

**DiagramObject Attributes:**

Attribute	Type	Notes	See also
<b>Bottom</b>	<i>Long</i>	Read/Write The bottom edge position of the element.	
<b>DiagramID</b>	<i>Long</i>	Read/Write The ID of the associated diagram.	
<b>ElementID</b>	<i>Long</i>	Read/Write	

Attribute	Type	Notes	See also
		The <i>ElementID</i> of the object instance in this diagram.	
<b>InstanceID</b>	<i>Long</i>	Read The attribute that holds the connector identifier for the current model.	
<b>Left</b>	<i>Long</i>	Read/Write The left edge position of the element.	
<b>ObjectType</b>	<a href="#">ObjectTyp</a> <a href="#">e</a> <sup>[2822]</sup>	Read only Distinguishes objects referenced through a Dispatch interface.	
<b>Right</b>	<i>Long</i>	Read/Write The right edge position of the element.	
<b>Sequence</b>	<i>Long</i>	Read/Write The sequence position when loading the object into the diagram (this affects its Z order). The Z-order is one-based and the lowest value is in the foreground.	
<b>Style</b>	<i>Variant</i>	Write only (reading this value gives undefined results) The style information for this object.	<a href="#">Setting the Style</a> <sup>[2958]</sup>
<b>Top</b>	<i>Long</i>	Read/Write The top edge position of the element.	

**DiagramObject Methods**

Method	Type	Notes
<b>GetLastError ()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.  This function is rarely used as an exception is thrown when an error occurs.
<b>Update ()</b>	<i>Boolean</i>	Update the current <i>Diagram Object</i> after modification or appending a new item

Method	Type	Notes
		If <b>false</b> is returned, check the <i>GetLastError</i> function for more information.

### Topics

Topic	Detail	See also
<b>Setting The Style</b>	<p>The <i>Style</i> attribute is used for setting the appearance of a DiagramObject; it is set with a string value in the format:</p> <p><code>BCol =n; BFol =n; LCol =n; LWth =n;</code></p> <p>where:</p> <ul style="list-style-type: none"> <li>• <i>BCol</i> = Background Color</li> <li>• <i>BFol</i> = Font Color</li> <li>• <i>LCol</i> = Line Color</li> <li>• <i>LWth</i> = Line Width</li> </ul> <p>The color value is a decimal representation of the hex RGB value, where Red=FF, Green=FF00 and Blue=FF0000</p> <p><code>DiagramObj . Style = " BCol =35723; BFol =9342520; LCol =9342520; LWth =1; "</code></p> <p>The following code snippet shows how you might change the style settings for all of the objects in the current diagram, in this case changing everything to red:</p> <pre> For Each aDiagramObj In aDiagram.DiagramObjects     aDiagramObj . Style = " BCol =255;     BFol =9342520; LCol =9342520; LWth =1; "     aDiagramObj . Update aRepos . ReloadDiagram aDiagramObj . DiagramID Next </pre>	

#### 20.2.2.8.4 SwimlaneDef Class

A **SwimlaneDef** object makes available attributes relating to a single row or column in a list of swimlanes.

Attribute	Type	Notes
<b>Bold</b>	<i>Boolean</i>	Read/Write Show the title text in bold.

Attribute	Type	Notes
<b>FontColor</b>	<i>Long</i>	Read/Write The RGB color used to draw the titles.
<b>HideClassifier</b>	<i>Boolean</i>	Read/Write Removes any classifier from the title display.
<b>HideNames</b>	<i>Boolean</i>	Read/Write Set to <b>true</b> to hide the swimlane titles.
<b>LineColor</b>	<i>Long</i>	Read/Write The RGB color used to draw swimlane borders.
<b>LineWidth</b>	<i>Long</i>	Read/Write The width, in pixels, of the line used to draw swimlanes. Valid values are <b>1</b> , <b>2</b> or <b>3</b> .
<b>Locked</b>	<i>Boolean</i>	Read/Write If set to <b>true</b> , disables user modification of the swimlanes via the diagram.
<b>ObjectType</b>	<a href="#">ObjectType</a> [2822]	Read only Distinguishes objects referenced through a Dispatch interface.
<b>Orientation</b>	<i>String</i>	Read/Write Indicates whether the swimlanes are vertical or horizontal.
<b>ShowInTitleBar</b>	<i>Boolean</i>	Read/Write Enables vertical swimlane titles to be shown in the title bar.
<b>Swimlanes</b>	<a href="#">Swimlanes</a> [2960]	Read/Write A list of individual swimlanes.

### 20.2.2.8.5 Swimlanes Class

A *Swimlanes* object is attached to a diagram's *SwimlaneDef* object and provides a mechanism to access individual swimlanes.

#### Swimlanes Attributes

Attribute	Type	Notes
<b>Count</b>	<i>Long</i>	Read/Write Gives the number of swimlanes.
<b>ObjectType</b>	<a href="#">ObjectType</a> <small>[2822]</small>	Read only Distinguishes objects referenced through a Dispatch interface.

#### Swimlanes Methods

Method	Type	Notes
<b>Add (string Title, long Width)</b>	<a href="#">Swimlane</a> <small>[2961]</small>	Adds a new swimlane to the end of the list, and returns a swimlane object representing the newly added entry.  Parameters: <ul style="list-style-type: none"> <li>Title: String - The title text that appears at the top of the swimlane; this can be the same as an existing swimlane title</li> <li>Width: Long - The width of the swimlane in pixels</li> </ul>
<b>Delete (object Index)</b>	<i>Void</i>	Deletes a selected swimlane.  If the string matches more than one entry, only the first entry is deleted.  Parameter: <ul style="list-style-type: none"> <li>Index: Object - Either a string representing the title text or an integer representing the zero-based index of the swimlane to delete</li> </ul>
<b>DeleteAll ()</b>	<i>Void</i>	Removes all swimlanes.
<b>Insert (long Index, string Title, long Width)</b>	<a href="#">Swimlane</a> <small>[2961]</small>	Inserts a swimlane at a specific position, and returns a swimlane object representing the newly added entry.  Parameters: <ul style="list-style-type: none"> <li>Index: Long - The zero-based index of the existing Swimlane before which this new entry is inserted</li> <li>Title: String - The title text which appears at the top of the swimlane; this can be the same as an existing swimlane title</li> <li>Width: Long - The width of the swimlane in pixels</li> </ul>



Method	Type	Notes
<b>Items (object Index)</b>	<a href="#">Swimlane</a> collection	<p>Accesses an individual swimlane.</p> <p>If the string matches more than one swimlane title, the first matching swimlane is returned.</p> <p>Parameter:</p> <ul style="list-style-type: none"> <li>Index: Object - Either a string representing the title text or an integer representing the zero-based index of the swimlane to get</li> </ul>

Learn more

- [SwimlaneDef Class](#)

**20.2.2.8.6 Swimlane Class**

A *Swimlane* object makes available attributes relating to a single row or column in a list of swimlanes.

Attribute	Type	Notes
<b>BackColor</b>	<i>Long</i>	<p>Read/Write</p> <p>The RGB color that the swimlane is filled with.</p>
<b>ClassifiedGuid</b>	<i>String</i>	<p>Read/Write</p> <p>The GUID of the classifier Class. This can be obtained from the corresponding Element object via the <i>ElementGUID</i> property.</p>
<b>ObjectType</b>	<a href="#">ObjectType</a>	<p>Read only</p> <p>Distinguishes objects referenced through a Dispatch interface.</p>
<b>Title</b>	<i>String</i>	<p>Read/Write</p> <p>The text at the head of the swimlane.</p>
<b>Width</b>	<i>Long</i>	<p>Read/Write</p> <p>The width of the swimlane, in pixels.</p>

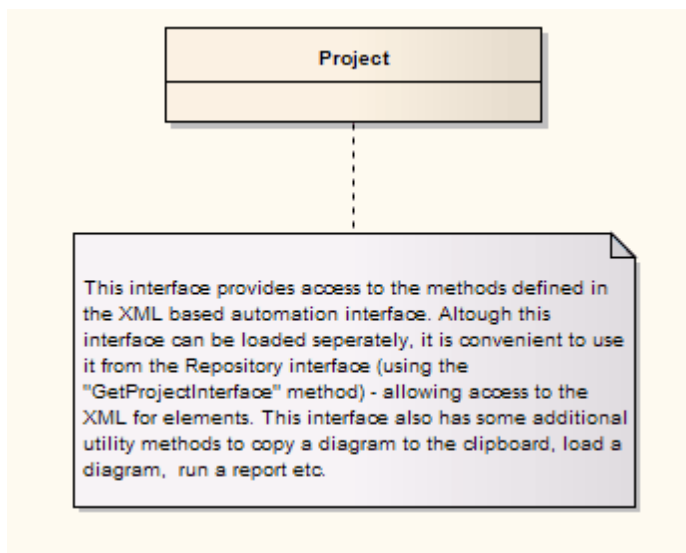
Learn more

- [Swimlane Class](#)

### 20.2.2.9 Project Interface Package

The *Enterprise Architect.Project* interface. This is the interface to Enterprise Architect elements; it also includes some utility functions. You can get a pointer to this interface using the *Repository*. *GetProjectInterface* method.

#### Example



#### 20.2.2.9.1 Project Class

The Project interface can be accessed from the Repository using *GetProjectInterface()*. The returned interface provides access to the XML-based Enterprise Architect Automation Interface. Use this interface to get XML for the various internal elements and to run some utility functions to perform tasks such as load diagrams or run reports.

#### Project Attributes

Attribute	Type	Notes
<b>ObjectType</b>	<a href="#">Object Type</a> <sup>2822</sup>	Read only Distinguishes objects referenced through a Dispatch interface.


#### Project Methods

Method	Type	Notes	See also
<b>CancelValidation ()</b>	<i>Void</i>	Cancels a validation process.	<a href="#">Model Validation</a> <sup>2594</sup>
<b>CanValidate ()</b>	<i>Boolean</i>	Returns a value to indicate that the Model Validation component is loaded.	

Method	Type	Notes	See also
<b>CreateBaseline (string PackageGUID, string Version, string Notes)</b>	<i>Boolean</i>	Creates a Baseline of a specified package. Parameters: <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package to Baseline</li> <li>Version: String - the version of the Baseline</li> <li>Notes: String - any notes concerning the Baseline</li> </ul>	
<b>CreateBaselineEx (string PackageGUID, string Version, string Notes, EA.CreateBaselineFlag Flags)</b>	<i>Boolean</i>	Creates a Baseline of a specified package, with a flag to <b>exclude package contents</b> below the first level. Parameters: <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package to be Baselined</li> <li>Version: String - the version of the Baseline</li> <li>Notes: String - any notes concerning the Baseline</li> <li>Flags: <i>EA.CreateBaselineFlag</i> - whether or not to exclude the package contents below the first level</li> </ul>	<a href="#">Create Baselines</a> <sup>[462]</sup> <a href="#">EA.CreateBaselineFlag</a> <sup>[2817]</sup>
<b>DefineRule (string CategoryId, EA.EnumMVErrorType ErrorType, string ErrorMessage)</b>	<i>String</i>	Defines the individual rules that can be performed during model validation. It must be called once for each rule from the <i>EA_OnInitializeUserRules</i> broadcast handler. The return value is a <i>RuleId</i> , which can be used for reference purposes when an individual rule is executed by Enterprise Architect during model validation. See the <i>Model Validation Example</i> for a detailed example of the use of this method. Parameters: <ul style="list-style-type: none"> <li>CategoryId: String - should be passed the return value from the <i>DefineRuleCategory</i> method</li> <li>ErrorType: <i>EA.EnumMVErrorType</i> - depending on the severity of the error being validated, can be: <ul style="list-style-type: none"> <li><b>mvErrorCritical</b></li> <li><b>mvError</b></li> <li><b>mvWarning</b>, or</li> </ul> </li> </ul>	<a href="#">EA_OnInitializeUserRules</a> <sup>[3048]</sup> <a href="#">Model Validation Example</a> <sup>[3056]</sup> <a href="#">DefineRuleCategory</a> <sup>[2964]</sup> <a href="#">PublishResult</a> <sup>[2979]</sup>

Method	Type	Notes	See also
		<ul style="list-style-type: none"> <li>• <b>mvlInformation</b></li> <li>• ErrorMessage: String - can contain a default error string, although this is probably overridden by the <i>PublishResult</i> call</li> </ul>	
<b>DefineRuleCategory (string CategoryName)</b>	<i>String</i>	<p>Defines a category of rules that can be performed during model validation (there is typically one category per Add-In). It must be called once from the <i>EA_OnInitializeUserRules</i> broadcast handler.</p> <p>The return value is a CategoryId that must to be passed to the <i>DefineRule</i> method.</p> <p>See the <i>Model Validation Example</i> for a detailed example of the use of this method.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• CategoryName: String - a text string that is visible in the Model Validation Configuration dialog</li> </ul>	<a href="#">EA_OnInitializeUserRules</a> <sup>[3048]</sup> <a href="#">DefineRule</a> <sup>[2963]</sup> <a href="#">Model Validation Example</a> <sup>[3056]</sup>
<b>DeleteBaseline (string BaselineGUID)</b>	<i>Boolean</i>	<p>Deletes a Baseline, identified by the BaselineGUID, from the repository.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• BaselineGUID: String - the GUID (in XML format) of the Baseline to delete</li> </ul>	
<b>DoBaselineCompare (string PackageGUID, string Baseline, string ConnectString)</b>	<i>String</i>	<p>Performs a Baseline comparison using the supplied package GUID and Baseline GUID (obtained in the result list from <i>GetBaselines</i>).</p> <p>Optionally you can include the connection string required to find the Baseline if it exists in a different model file.</p> <p>This method returns a log file of the status of all elements found and compared in the difference procedure. You can use this log information as input to <i>DoBaselineMerge</i> - automatically merging information from the Baseline.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• PackageGUID: String - the GUID (in XML format) of the package to run the comparison on</li> <li>• Baseline: String - the GUID (in XML format) of the Baseline to run the comparison on</li> <li>• ConnectString: String - the location of the external .EAP file or DBMS to extract</li> </ul>	<a href="#">GetBaselines</a> <sup>[2972]</sup> <a href="#">DoBaselineMerge</a> <sup>[2965]</sup>

Method	Type	Notes	See also
		the Baseline from	
<b>DoBaselineMerge (</b> <b>string PackageGUID,</b> <b>string Baseline,</b> <b>string</b> <b>MergeInstructions,</b> <b>string ConnectString)</b>	<i>String</i>	<p>Performs a batch merge based on instructions contained in an XML file (<i>MergeInstructions</i>). You can supply an optional connection string if the Baseline is located in another model.</p> <p>In the <i>MergeInstructions</i> file, each <i>MergeItem</i> node supplies the GUID of a differenced item from the XML difference log. As the merge is uni-directional and actioned in only one possible way, no additional arguments are required. Enterprise Architect chooses the correct procedure based on the Difference results.</p> <pre>&lt;Merge&gt;   &lt;MergeItem guid="{XXXXXX}" /&gt;   &lt;MergeItem guid="{XXXXXX}" /&gt; &lt;/Merge&gt;</pre> <p>Alternatively, you can supply a single <i>MergeItem</i> with a GUID of <i>RestoreAll</i>. In this case, Enterprise Architect batch-processes ALL differences.</p> <pre>&lt;Merge&gt;   &lt;MergeItem     guid="RestoreAll"     changed="true"     baselineOnly="true"     modelOnly="true"     moved="true"     fullRestore="false" /&gt; &lt;/Merge&gt;</pre> <p>Parameters:</p> <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package to merge the Baseline into</li> <li>Baseline: String - the GUID of the Baseline (in XML format) to merge into the package</li> <li>MergeInstructions: String - the file containing the GUID of each differenced item from the XML difference log returned by <i>DoBaselineCompare()</i></li> <li>ConnectString: String - the location of the EAP file or DBMS to get the Baseline from, if not in the same</li> </ul>	<a href="#">DoBaselineCompare()</a> <sup>[2964]</sup>

Method	Type	Notes	See also
		model as the package	
<b>EnumDiagramElements (string DiagramGUID)</b>	protected abstract: <i>String</i>	Gets an XML list of all elements in a diagram. Parameters: <ul style="list-style-type: none"> <li>DiagramGUID: String - the GUID (in XML format) of the diagram to get elements for</li> </ul>	
<b>EnumDiagrams (string PackageGUID)</b>	protected abstract: <i>String</i>	Gets an XML list of all diagrams in a specified package. Parameters: <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package to list diagrams for</li> </ul>	
<b>EnumElements (string PackageGUID)</b>	protected abstract: <i>String</i>	Gets an XML list of elements in a specified package. Parameters: <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package to get a list of elements for</li> </ul>	
<b>EnumLinks (string ElementGUID)</b>	protected abstract: <i>String</i>	Gets an XML list of connectors for a specified element. Parameters: <ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element to get all associated connectors for</li> </ul>	
<b>EnumPackages (string PackageGUID)</b>	protected abstract: <i>String</i>	Gets an XML list of child packages inside a parent package. Parameters: <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the parent package</li> </ul>	
<b>EnumProjects ()</b>	protected abstract: <i>String</i>	Gets a list of projects in the current file; corresponds to <i>Models</i> in <i>Repository</i> .	<a href="#">Models</a> 
<b>EnumViews ()</b>	protected abstract: <i>String</i>	Enumerates the Views for a project. Returned as an XML document.	
<b>EnumViewEx (string ProjectGUID)</b>	protected abstract	Gets a list of Views in the current project. Parameters:	

Method	Type	Notes	See also
	t: <i>String</i>	<ul style="list-style-type: none"> <li>ProjectGUID: String - the GUID (in XML format) of the project to get views for</li> </ul>	
<b>Exit ()</b>	protected abstract t: <i>String</i>	<p>Exits the current instance of Enterprise Architect; this function is maintained for backward compatibility and should never be called.</p> <p>Enterprise Architect automatically exits when you are no longer using any of the provided objects.</p>	
<b>ExportPackageXML (</b> string PackageGUID, enumXMIMType XMIMType, long DiagramXML, long DiagramImage, long FormatXML, long UseDTD, string FileName)	protected abstract t: <i>String</i>	<p>Exports XMI for a specified package.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package to be exported</li> <li>XMIMType: EnumXMIMType - specifies the XMI type and version information; see <i>XMIMType Enum</i> for accepted values</li> <li>DiagramXML: Long - <b>true</b> if XML for diagrams is required; accepted values: <ul style="list-style-type: none"> <li><b>0</b> = Do not export diagrams</li> <li><b>1</b> = Export diagrams</li> <li><b>2</b> = Export diagrams along with alternate images</li> </ul> </li> <li>DiagramImage: Long - the format for diagram images to be created at the same time; accepted values: <ul style="list-style-type: none"> <li><b>-1</b>=NONE</li> <li><b>0</b>=EMF</li> <li><b>1</b>=BMP</li> <li><b>2</b>=GIF</li> <li><b>3</b>=PNG</li> <li><b>4</b>=JPG</li> </ul> </li> <li>FormatXML: Long - <b>true</b> if XML output should be formatted prior to saving</li> <li>UseDTD: Long - <b>true</b> if a DTD should be used</li> <li>FileName: String - the filename to output to</li> </ul>	<a href="#">XMIMType Enum</a> <sup>[2825]</sup>
<b>ExportPackageXMIMEx (</b> string PackageGUID, enumXMIMType XMIMType, long DiagramXML,	protected abstract t: <i>String</i>	<p>Exports XMI for a specified package, with a flag to determine whether the export <b>includes package content</b> below the first level.</p> <p>Parameters:</p>	<a href="#">Package Content</a> <sup>[488]</sup> <a href="#">XMIMType Enum</a> <sup>[2825]</sup> <a href="#">ExportPackageXMIMFI</a>

Method	Type	Notes	See also
<b>long DiagramImage,</b> <b>long FormatXML,</b> <b>long UseDTD,</b> <b>string FileName,</b> <b>ea.</b> <b>ExportPackageXMIFlag</b> <b>Flags)</b>		<ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package to be exported</li> <li>XMLType: EnumXMLType - specifies the XML type and version information; see <i>XMLType Enum</i> for accepted values</li> <li>DiagramXML: Long - <b>true</b> if XML for diagrams is required; accepted values: <ul style="list-style-type: none"> <li><b>0</b> = Do not export diagrams</li> <li><b>1</b> = Export diagrams</li> <li><b>2</b> = Export diagrams along with alternate images</li> </ul> </li> <li>DiagramImage: Long - the format for diagram images to be created at the same time; accepted values: <ul style="list-style-type: none"> <li><b>-1</b>=NONE</li> <li><b>0</b>=EMF</li> <li><b>1</b>=BMP</li> <li><b>2</b>=GIF</li> <li><b>3</b>=PNG</li> <li><b>4</b>=JPG</li> </ul> </li> <li>FormatXML: Long - <b>true</b> if XML output should be formatted prior to saving</li> <li>UseDTD: Long - <b>true</b> if a DTD should be used.</li> <li>FileName: String - the filename to output to</li> <li>Flags: ea.ExportPackageXMIFlag - whether or not to include package content below the first level (currently only supported for <i>xmiEADefault</i>)</li> </ul>	<a href="#">ag</a> <sup>[2820]</sup>
<b>GenerateClass (</b> <b>string ElementGUID,</b> <b>string ExtraOptions)</b>	<i>Boolean</i>	Generates the code for a single Class. Parameters: <ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element to generate</li> <li>ExtraOptions: String - enables extra options to be given to the command; currently unused</li> </ul>	
<b>GenerateDiagramFromScenario (</b> <b>string ElementGUID,</b> <b>EnumScenarioDiagramType DiagramType,</b>	<i>Boolean</i>	Generates various diagrams from the Structured Specification of an element. Parameters: <ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element containing</li> </ul>	<a href="#">ScenarioDiagramType Enum</a> <sup>[2824]</sup>



Method	Type	Notes	See also
<b>long OverwriteExistingDiagram()</b>		<p>the Structured Specification</p> <ul style="list-style-type: none"> <li>Diagram Type: EnumScenarioDiagramType - the type of diagram to generate; see <i>ScenarioDiagramType Enum</i> for accepted values</li> <li>OverwriteExistingDiagram: Long - determines whether to overwrite the existing diagram or synchronize the existing elements with the scenario steps</li> </ul> <p><b>0</b> = Delete the existing diagram and elements, and create a new diagram and elements  <b>1</b> = Synchronize existing elements with the scenario steps and preserve the diagram layout  <b>2</b> = Synchronize existing elements with the scenario steps and re-cast the diagram layout  <b>3</b> = Do not generate a diagram if one already exists</p>	
<b>GenerateElementDDL (string ElementGUID, string FileName, string ExtraOptions)</b>	<i>Boolean</i>	<p>Generates DDL for an element.</p> <p>For example:</p> <pre>owner=true; pkfkconstraints=true; commentlevel=3;</pre> <p>Parameters:</p> <ul style="list-style-type: none"> <li>ElementGUID: String – the GUID (in XML format) of the element to generate DDL for</li> <li>FileName: String – the target file path to which to write the generated DDL</li> <li>ExtraOptions: String – enables extra options to be given to the command; currently enables: <ul style="list-style-type: none"> <li>Generate Owner (<i>owner</i>)</li> <li>Generate PK/FK Key Constraints (<i>pkfkconstraints</i>)</li> <li>Generate Comments: <p><b>0</b>=None  <b>1</b>=Table  <b>2</b>=Column  <b>3</b>=All</p> </li> </ul> </li> </ul>	

Method	Type	Notes	See also
<b>GeneratePackage (string PackageGUID, string ExtraOptions)</b>	<i>Boolean</i>	<p>Generates the code for all Classes within a package.</p> <p>For example:</p> <pre>recurse=1; overwrite=1; dir=C:\</pre> <p>Parameters:</p> <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package to generate code for</li> <li>ExtraOptions: String - enables extra options to be given to the command; currently enables: <ul style="list-style-type: none"> <li>Generation of all subpackages (<i>recurse</i>)</li> <li>Force overwrite of all files (<i>overwrite</i>) and</li> <li>Specification to auto generate all paths (<i>dir</i>)</li> </ul> </li> </ul>	
<b>GeneratePackageDDL (string PackageGUID, string FileName, string ExtraOptions)</b>	<i>Boolean</i>	<p>Generates DDL for a package.</p> <p>For example:</p> <pre>owner=true; pkfkconstraints=true; commentlevel=3; singlefile=true; includechild=true;</pre> <p>Parameters:</p> <ul style="list-style-type: none"> <li>PackageGUID: String – the GUID (in XML format) of the package to generate DDL for</li> <li>FileName: String – the target file path to which to write the generated DDL</li> <li>ExtraOptions: String – enables extra options to be given to the command; currently enables: <ul style="list-style-type: none"> <li>Generate DDL for child packages (<i>includechild</i>)</li> <li>Generate Owner (<i>owner</i>)</li> <li>Generate PK/FK Key Constraints (<i>pkfkconstraints</i>)</li> <li>Generate Comments: <p>0 - None 1 - Table 2 - Column</p> </li> </ul> </li> </ul>	

Method	Type	Notes	See also
		<p>3 - All</p> <ul style="list-style-type: none"> <li>Generate in SingleFile (<i>singlefile</i>)</li> </ul>	
<b>GenerateTestFromScenario</b> ( string ElementGUID, EnumScenarioTestType TestType)	Boolean	<p>Generates either an <b>Internal test</b> or an <b>External test</b> from the Structured Specification of an element.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element containing the Structured Specification</li> <li>TestType: EnumScenarioTestType - the type of test to generate; see <i>ScenarioTestType Enum</i> for accepted values</li> </ul>	<a href="#">Generate Test Cases</a> <sup>[986]</sup> <a href="#">ScenarioTestType Enum</a> <sup>[2825]</sup>
<b>GenerateWSDL</b> ( string WSDLComponentGUID, string Filename, string Encoding, string ExtraOptions)	Boolean	<p>Generates WSDL for the specified WSDL stereotyped Component.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>WSDLComponentGUID: String - the GUID (in XML format) of the WSDL stereotyped Component</li> <li>Filename: String - the target file path</li> <li>Encoding: String - the XML encoding for the code page instruction</li> <li>ExtraOptions: String - enables extra options to be given to the command; currently unused</li> </ul>	<a href="#">Web Services - WSDL</a> <sup>[2423]</sup>
<b>GenerateXSD</b> ( string PackageGUID, string FileName, string Encoding, string Options)	Boolean	<p>Creates an XML schema for a package, specified by its GUID. Returns <b>true</b> on success.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package</li> <li>FileName: String - the target filepath</li> <li>Encoding: String - the XML encoding for the code page instruction</li> <li>Options: String - enables extra options to be given to the command; currently enables: <ul style="list-style-type: none"> <li><i>GenGlobalElement</i> - turn the generation of global elements for all global <i>ComplexTypes</i> <b>On</b> or <b>Off</b>; for example: - GenGlobalElement = 1</li> </ul> </li> </ul>	

Method	Type	Notes	See also
<b>GetBaselines (string PackageGUID, string ConnectString)</b>	<i>String</i>	<p>Returns a list (in XML format) of Baselines associated with the supplied package GUID. Optionally, you can provide a connection string to get Baselines from the same package, but located in a different model file (or DBMS).</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package to get Baselines for</li> <li>ConnectString: String - the location of the EAP file or DBMS to get the Baselines from, if not in the same model as the package</li> </ul>	
<b>GetDiagram (string DiagramGUID)</b>	protected abstract: <i>String</i>	<p>Gets the diagram details, in XML format.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>DiagramGUID: String - the GUID (in XML format) of the diagram to get details for</li> </ul>	
<b>GetElement (string ElementGUID)</b>	protected abstract: <i>String</i>	<p>Gets XML for the specified element.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element to retrieve XML for</li> </ul>	
<b>GetElementConstraints (string ElementGUID)</b>	protected abstract: <i>String</i>	<p>Gets constraints for an element, in XML format.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element</li> </ul>	
<b>GetElementEffort (string ElementGUID)</b>	protected abstract: <i>String</i>	<p>Gets efforts for an element, in XML format.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element</li> </ul>	
<b>GetElementFiles (string ElementGUID)</b>	protected abstract: <i>String</i>	<p>Gets metrics for an element, in XML format.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element</li> </ul>	
<b>GetElementMetrics (string ElementGUID)</b>	protected	Gets files for an element, in XML format.	

Method	Type	Notes	See also
	abstract: <i>String</i>	Parameters: <ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element</li> </ul>	
<b>GetElementProblems (string ElementGUID)</b>	protected abstract: <i>String</i>	Gets a list of issues (problems) associated with an element, in XML format. Parameters: <ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element</li> </ul>	
<b>GetElementProperties (string ElementGUID)</b>	protected abstract: <i>String</i>	Gets Tagged values for an element, in XML format. Parameters: <ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element</li> </ul>	
<b>GetElementRequirements (string ElementGUID)</b>	protected abstract: <i>String</i>	Gets a list of requirements for an element, in XML format. Parameters: <ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element</li> </ul>	
<b>GetElementResources (string ElementGUID)</b>	protected abstract: <i>String</i>	Gets a list of resources for an element, in XML format. Parameters: <ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element</li> </ul>	
<b>GetElementRisks (string ElementGUID)</b>	protected abstract: <i>String</i>	Gets a list of risks associated with an element, in XML format. Parameters: <ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element</li> </ul>	
<b>GetElementScenarios (string ElementGUID)</b>	protected abstract: <i>String</i>	Gets a list of scenarios for an element, in XML format. Parameters: <ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element</li> </ul>	
<b>GetElementTests (string ElementGUID)</b>	protected abstract: <i>String</i>	Gets a list of tests for an element, in XML format. Parameters: <ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element</li> </ul>	


Method	Type	Notes	See also
		XML format) of the element	
<b>GetFileNameDialog (</b> <b>string Filename,</b> <b>string FilterString,</b> <b>long FilterIndex,</b> <b>long Flags,</b> <b>string InitialDirectory,</b> <b>long OpenOrSave)</b>	<i>String</i>	<p>Opens a standard File Open or Save As dialog and returns a string containing the full path to the selected file on success. Returns an empty string if the dialog was canceled.</p> <p>For example:</p> <pre> Filename = " " FilterString = " CSV Files (*.csv) *.csv All Files (*.*) *.*  " FilterIndex = 1 Flags = &amp;H2 ' OFN_OVERWRITEPROMPT InitialDirectory = " " OpenOrSave = 1 filepath = Project. GetFileNameDialog (Filename, FilterString, FilterIndex, Flags, InitialDirectory, OpenOrSave) </pre> <p>In this example, the Save As dialog will prompt for a CSV file.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• Filename: String - default filename specified in the dialog</li> <li>• FilterString: String - delimited list of available file type filters</li> <li>• FilterIndex: Long - One-based index of the filter to be used by default</li> <li>• Flags: Long - additional bit flags used to initialize the file dialog. See OPENFILENAME structure in MSDN documentation for accepted values.</li> <li>• InitialDirectory: String - directory path to open this dialog</li> <li>• OpenOrSave: Long - Show dialog as an "Open" or "Save As" style dialog. Accepted values: 0 = Open, 1 = Save As</li> </ul>	
<b>GetLastError ()</b>	protected abstract: <i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.	
<b>GetLink (</b> <b>string LinkGUID)</b>	protected abstract: <i>String</i>	<p>Gets connector details, in XML format.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• LinkGUID: String - the GUID (in XML format) of the connector to get details</li> </ul>	

Method	Type	Notes	See also
		of	
<b>GUIDtoXML (string GUID)</b>	<i>String</i>	<p>Changes an internal GUID to the form used in XML.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>GUID: String - the Enterprise Architect style GUID to convert to XML format</li> </ul>	
<b>ImportDirectory (string PackageGUID, string Language, string DirectoryPath, string ExtraOptions)</b>	<i>Boolean</i>	<p>Imports a source code directory into the model.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package to reverse engineer code into</li> <li>Language: String - specifies the language of the code to be imported</li> <li>DirectoryPath: String - specifies the path where the code is found on the computer</li> <li>ExtraOptions: String - enables extra options to be given to the command; currently enables import of source from all child directories (<i>recurse</i>) - for example: <i>recurse=1</i></li> </ul>	
<b>ImportFile (string PackageGUID, string Language, string FileName, string ExtraOptions)</b>	<i>Boolean</i>	<p>Imports an individual file or binary module into the model, in a package per namespace style import.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package to reverse engineer code into; this is expected to be a namespace root package.</li> <li>Language: String - specifies the language of the code to be imported</li> </ul> <p>Use the value <b>"DNPE"</b> to import a binary module; this imports a .Net assembly or Java .class file, but <b>not</b> a .jar file.</p> <ul style="list-style-type: none"> <li>Filename: String - specifies the path where the code or module is found on the computer</li> <li>ExtraOptions: String - enables extra options to be given to the command; currently unused</li> </ul>	

Method	Type	Notes	See also
<b>ImportPackageXML</b> ( string PackageGUID, string Filename, long ImportDiagrams, long StripGUID)	<i>String</i>	<p>Imports an XML file at a point in the tree.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the target package to import the XML file into (or overwrite with the XML file)</li> <li>Filename or XMLText: String - the name of the XML file</li> </ul> <p>If the String is of type <i>filename</i> it is interpreted as a source file, otherwise the String is imported as XML text.</p> <ul style="list-style-type: none"> <li>ImportDiagrams: Long</li> <li>StripGUID: Long - boolean value to indicate whether to replace the element UniqueIDs on import; if stripped, then a package could be imported twice into Enterprise Architect, as two different versions</li> </ul> <p>Returns an empty string if successful, or returns an error message on failure.</p>	
<b>LayoutDiagram</b> ( string DiagramGUID, long LayoutStyle)	<i>Boolean</i>	<p><b>Deprecated.</b> it is recommended that <i>LayoutDiagramEx</i> is used instead.</p> <p>Calls the function to automatically layout a diagram in hierarchical fashion. It is only recommended for Class and Object diagrams.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>DiagramGUID: String - the GUID (in XML format) of the diagram to lay out</li> <li>LayoutStyle: Long - always ignored</li> </ul>	
<b>LayoutDiagramEx</b> ( string DiagramGUID, long LayoutStyle, long Iterations, long LayerSpacing, long ColumnSpacing, boolean SaveToDiagram )	<i>Boolean</i>	<p>Calls the function to automatically layout a diagram in hierarchical fashion. It is only recommended for Class and Object diagrams.</p> <p><i>LayoutStyle</i> accepts the following options</p> <ul style="list-style-type: none"> <li>Default Options: <ul style="list-style-type: none"> <li>IsDiagramDefault</li> <li>IsProgramDefault</li> </ul> </li> <li>Cycle Removal Options: <ul style="list-style-type: none"> <li>IsCycleRemoveGreedy</li> </ul> </li> </ul>	<a href="#">ConstLayoutStyles Enum</a> <sup>[2816]</sup>



Method	Type	Notes	See also
		<p>IsCycleRemoveDFS</p> <ul style="list-style-type: none"> <li>Layering Options: <ul style="list-style-type: none"> <li>IsLayeringLongestPathSink</li> <li>IsLayeringLongestPathSource</li> <li>IsLayeringOptimalLinkLength</li> </ul> </li> <li>Initialize Options: <ul style="list-style-type: none"> <li>IsInitializeNaive</li> <li>IsInitializeDFSOut</li> <li>IsInitializeDFSIn</li> </ul> </li> <li>Crossing Reduction Option: <ul style="list-style-type: none"> <li>IsCrossReduceAggressive</li> </ul> </li> <li>Layout Options - Direction <ul style="list-style-type: none"> <li>IsLayoutDirectionUp</li> <li>IsLayoutDirectionDown</li> <li>IsLayoutDirectionLeft</li> <li>IsLayoutDirectionRight</li> </ul> </li> </ul> <p>Parameters:</p> <ul style="list-style-type: none"> <li>DiagramGUID: String - the GUID (in XML format) of the diagram to lay out</li> <li>LayoutStyle: Long - the layout style</li> <li>Iterations: Long - the number of layout iterations the Layout process should take to perform cross reduction (Default value = <b>4</b>)</li> <li>LayerSpacing: Long - the per-element layer spacing the Layout process shall use (Default value = <b>20</b>)</li> <li>ColumnSpacing: Long - the per-element column spacing the Layout process shall use (Default value = <b>20</b>)</li> <li>SaveToDiagram: Boolean - specifies whether or not Enterprise Architect should save the supplied layout options as default to the diagram in question</li> </ul>	
<b>LoadControlledPackage</b> ( <b>string PackageGUID</b> )	<i>String</i>	<p>Loads a package that has been marked and configured as controlled. The filename details are stored in the package control data.</p> <p>Parameters:</p>	

Method	Type	Notes	See also
		<ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package to load</li> </ul>	
<b>LoadDiagram (string DiagramGUID)</b>	protected abstract: Boolean	<p>Loads a diagram by its GUID.</p> <p>Parameter:</p> <ul style="list-style-type: none"> <li>DiagramGUID: String - the GUID (in XML format) of the diagram to load; if you retrieve the GUID using the Diagram interface, use the <i>GUIDtoXML</i> function to convert it to XML format</li> </ul>	<a href="#">GUIDtoXML</a>  <sup>[2975]</sup>
<b>LoadProject (string FileName)</b>	protected abstract: Boolean	<p>Loads an Enterprise Architect project file.</p> <p>Do not use this method if you have accessed the Project interface from the Repository, which has already loaded a file.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>FileName: String - the name of the project file to load</li> </ul>	
<b>Migrate (string GUID, string SourceType, string DestinationType)</b>	Void	<p>Migrates a model (or part of a model) from one BPMN or SysML format to an upgraded format.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>GUID: String - the GUID of the package or element for which the contents are to be migrated</li> <li>SourceType: String - the type of model to be upgraded; accepted values: <ul style="list-style-type: none"> <li>BPMN</li> <li>BPMN1.1</li> <li>SysML1.1</li> <li>SysML1.2</li> </ul> </li> <li>DestinationType: String - the type of model to upgrade to; accepted values: <ul style="list-style-type: none"> <li>BPMN1.1</li> <li>BPMN1.1::BPEL</li> <li>BPMN2.0</li> <li>SysML1.2</li> <li>SysML1.3</li> </ul> </li> </ul>	
<b>MigrateToBPMN11 (string GUID,</b>	Void	Migrates every BPMN 1.0 construct in a package or an element (including elements,	


Method	Type	Notes	See also
<b>string Type)</b>		<p>attributes, diagrams and connectors) to BPMN 1.1.</p> <p>Parameters</p> <ul style="list-style-type: none"> <li>• GUID: String - the GUID of the package or element for which the contents are to be migrated to BPMN 1.1</li> <li>• Type: String - the type of upgrade, either just to BPMN 1.1 or to BPMN 1.1 and BPEL. Accepted values are:</li> </ul> <p><b>BPMN</b> = migrate to BPMN 1.1  <b>BPEL</b> = migrate to BPMN 1.1 and update:</p> <ul style="list-style-type: none"> <li>• any diagram with stereotype <i>BPMN</i> to <i>BPEL</i></li> <li>• any element with stereotype <i>BusinessProcess</i> to <i>BPELProcess</i></li> </ul> <p>Migrating to BPEL is possible only in the Ultimate or Business and Software Engineering editions of Enterprise Architect.</p>	
<b>ProjectTransfer (</b> <b>string SourceFilePath,</b> <b>string TargetFilePath,</b> <b>string LogFilePath)</b>	<i>Boolean</i>	<p>Transfers the project from a .EAP file or DBMS to a .EAP file.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• SourceFilePath: String - the path of the source file to transfer</li> <li>• TargetFilePath: String - the path of the target file; Enterprise Architect creates a new Base project in this location</li> <li>• LogFilePath: String - the path of the log file where the status of the transfer process is updated</li> </ul> <p>In automation, the target file does not have to exist; the file path is enough. Enterprise Architect creates a new, empty Base.EAP file and transfers the source project into it.</p>	
<b>PublishResult (</b> <b>string CategoryID,</b> <b>EA.EnumMVErrorType</b> <b>ErrorType,</b> <b>string ErrorMessage)</b>	<i>String</i>	<p>Returns the results of each rule that can be performed during model validation. It must be called once for each rule from the <i>EA_OnInitializeUserRules</i> broadcast handler.</p> <p>The return value is a <b>RuleId</b>, which can be used for reference purposes when an individual rule is executed by Enterprise Architect during model validation.</p> <p>See the <i>Model Validation Example</i> for a</p>	<p><a href="#">EA_OnInitializeUserRules</a> <sup>[3048]</sup></p> <p><a href="#">Model Validation Example</a> <sup>[3058]</sup></p> <p><a href="#">DefineRuleCategory</a> <sup>[2964]</sup></p>

Method	Type	Notes	See also
		<p>detailed example of the use of this method.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>CategoryId: String - should be passed the return value from the <i>DefineRuleCategory</i> method</li> <li>ErrorType: EA.EnumMVErrType - depending on the severity of the error being validated, can be: <ul style="list-style-type: none"> <li><b>mvErrorCritical</b></li> <li><b>mvError</b></li> <li><b>mvWarning</b>, or</li> <li><b>mvInformation</b></li> </ul> </li> <li>ErrorMessage: String - contains an error string</li> </ul>	
<b>PutDiagramImageOnClipboard (string DiagramGUID, long Type)</b>	protected abstract: Boolean	<p>Copies an image of the specified diagram to the clipboard.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>DiagramGUID: String - the GUID (in XML format) of the diagram to copy</li> <li>Type: Long - the file type <ul style="list-style-type: none"> <li>If Type = 0 then it is a metafile</li> <li>If Type = 1 then it is a Device Independent Bitmap</li> </ul> </li> </ul>	
<b>PutDiagramImageToFile (string Diagram GUID, string FileName, long Type)</b>	protected abstract: Boolean	<p>Saves an image of the specified diagram to file.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>DiagramGUID: String - the GUID (in XML format) of the diagram to save</li> <li>FileName: String - the name of the file to save the diagram into</li> <li>Type: Long - the file type <ul style="list-style-type: none"> <li>If type = 0 then it is a metafile</li> <li>If type = 1 then it uses the file type from the name extension (that is, .bmp, .jpg, .gif, .png, .tga)</li> </ul> </li> </ul>	
<b>ReloadProject ()</b>	protected abstract:	<p>Reloads the current project.</p> <p>This is a convenient method to refresh the current loaded project (in case of outside changes to the .EAP file).</p>	

Method	Type	Notes	See also
	<i>Boolean</i>		
<b>RunModelSearch (string Search, string SearchTerm, bool ShowInEA)</b>	<i>Void</i>	<p>Invokes the Model Search component.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>Search: String - the name of an Enterprise Architect defined search</li> <li>SearchTerm: String - the term to search for in the project</li> <li>ShowInEA: Boolean - execute the search and output in the Model Search window</li> </ul>	<a href="#">Model Search</a> <sup>[700]</sup> <a href="#">Repository Class</a> <sup>[2869]</sup> (a different call to Run Model Search)
<b>RunReport (string PackageGUID, string TemplateName, string Filename)</b>	protected abstract: <i>Void</i>	<p>Runs a named document report.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID of the package or master document to run the report on</li> <li>TemplateName: String - the document report template to use; if the PackageGUID has a stereotype of <i>MasterDocument</i>, the template is not required</li> <li>FileName: String - the file name and path to store the generated report. The file extension specified will determine the format of the generated document. E.g. RTF, PDF.</li> </ul>	<a href="#">Generate Documentation</a> <sup>[2644]</sup> <a href="#">Virtual Documents</a> <sup>[2669]</sup>
<b>RunHTMLReport (string PackageGUID, string ExportPath, string ImageFormat, string Style, string Extension)</b>	<i>String</i>	<p>Runs an HTML report (as for <b>Documentation   HTML Documentation</b> when you right-click on a package in the Project Browser).</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package or master document to run the report on</li> <li>ExportPath: String - the directory path to store the generated report files</li> <li>ImageFormat: String - file format in which to store images - .PNG or .GIF</li> <li>Style: String - name of the web style template to apply. Use &lt;default&gt; for the standard, system-provided template</li> <li>Extension: String - file extension for generated HTML files (Example: .htm)</li> </ul>	<a href="#">Create an HTML Report</a> <sup>[2744]</sup> <a href="#">Master Documents and Model Documents</a> <sup>[2759]</sup>
<b>SaveControlledPackage</b>	<i>String</i>	Saves a package that has been configured as	

Method	Type	Notes	See also
<b>( string PackageGUID)</b>		a controlled package, to XML. Only the package GUID is required, Enterprise Architect picks the rest up from the package control information.  Parameter: <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package to save</li> </ul>	
<b>SaveDiagramImageToFile ( string Filename)</b>	protected abstract: <i>String</i>	Saves a diagram image of the current diagram to file.  Parameters: <ul style="list-style-type: none"> <li>FileName: String - the filename of the image to save</li> </ul>	
<b>ShowWindow ( long Show)</b>	protected abstract: <i>Void</i>	Shows or hides the Enterprise Architect User Interface.  Parameters: <ul style="list-style-type: none"> <li>Show: Long</li> </ul>	
<b>SynchronizeClass ( string ElementGUID, string ExtraOptions)</b>	<i>Boolean</i>	Synchronizes a Class with the latest source code.  Parameters: <ul style="list-style-type: none"> <li>ElementGUID: String - the GUID (in XML format) of the element to update from code</li> <li>ExtraOptions: String - enables extra options to be given to the command; currently unused</li> </ul>	
<b>SynchronizePackage ( string PackageGUID, string ExtraOptions)</b>	<i>Boolean</i>	Synchronizes each Class in a package with the latest source code.  Parameters: <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID (in XML format) of the package containing the elements to update from code</li> <li>ExtraOptions: String - enables extra options to be given to the command; currently enables synchronization of all child packages (children) - for example : <i>children=1</i></li> </ul>	
<b>TransformElement ( string TransformName, string ElementGUID, string TargetPackage,</b>	<i>Boolean</i>	Transforms an element into a package.  Parameters: <ul style="list-style-type: none"> <li>TransformName: String - specifies the</li> </ul>	

Method	Type	Notes	See also
<b>string ExtraOptions)</b>		<p>transformation that should be executed</p> <ul style="list-style-type: none"> <li>• ElementGUID: String - the GUID (in XML format) of the element to transform</li> <li>• TargetPackageGUID: String - the GUID (in XML format) of the package to transform into</li> <li>• ExtraOptions: String - enables extra options to be given to the command: <ul style="list-style-type: none"> <li>• GenCode=<b>True/False</b> - articulate code generation from the transformed elements; this option supercedes the current model setting</li> </ul> </li> </ul>	
<b>TransformPackage (string TransformName, string SourcePackage, string TargetPackage, string ExtraOptions)</b>	<i>Boolean</i>	<p>Runs a transformation on the contents of a package.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• TransformName: String - specifies the transformation that should be executed</li> <li>• SourcePackageGUID: String - the GUID (in XML format) of the package to transform</li> <li>• TargetPackageGUID: String - the GUID (in XML format) of the package to transform into</li> <li>• ExtraOptions: String - enables extra options to be given to the command: <ul style="list-style-type: none"> <li>• GenCode=true/false - articulate code generation from the transformed elements; this option supercedes the current model setting</li> <li>• SubPackages=true/false - specify if the child packages are to be included whilst transforming a package</li> </ul> </li> </ul>	
<b>ValidateDiagram (string DiagramGUID)</b>	<i>Boolean</i>	<p>Invokes the Enterprise Architect Model Validation component, then validates the diagram (for correctness) and the elements and connectors within the diagram.</p> <p>Output can be viewed through <b>View   System Output &gt; Model Validation</b>.</p> <p>Returns a boolean value to indicate the success or failure of the process, regardless of the results of the validation.</p> <p>Parameters:</p>	

Method	Type	Notes	See also
		<ul style="list-style-type: none"> <li>DiagramGUID: String - the GUID of the Diagram Class object</li> </ul>	<a href="#">Model Validation</a>  <sup>2594</sup>
<b>ValidateElement (string ElementGUID)</b>	<i>Boolean</i>	<p>Invokes the Enterprise Architect Model Validation component, then validates the element and all child elements, diagrams, connectors, attributes and operations.</p> <p>Output can be viewed through <b>View   System Output &gt; Model Validation</b>.</p> <p>Returns a boolean value to indicate the success or failure of the process, regardless of the results of the validation.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>ElementGUID: String - the GUID of the Element Class object</li> </ul>	
<b>ValidatePackage (string PackageGUID)</b>	<i>Boolean</i>	<p>Invokes the Enterprise Architect Model Validation component, then validates the package and all sub-packages, elements, connectors and diagrams within it.</p> <p>Output can be viewed through <b>View   System Output &gt; Model Validation</b>.</p> <p>Returns a boolean value to indicate the success or failure of the process, regardless of the results of the validation.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>PackageGUID: String - the GUID of the Package Class object</li> </ul>	
<b>XMLtoGUID (string GUID)</b>	<i>String</i>	<p>Changes a GUID in XML format to the form used inside Enterprise Architect.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>GUID: String - the XML style GUID to convert to Enterprise Architect internal format</li> </ul>	

**Notes**

- These methods all require input GUIDs in XML format; use *GUIDtoXML* to change the Enterprise Architect GUID to an XML GUID

**Learn more**

- [GUIDtoXML](#)  <sup>2975</sup>



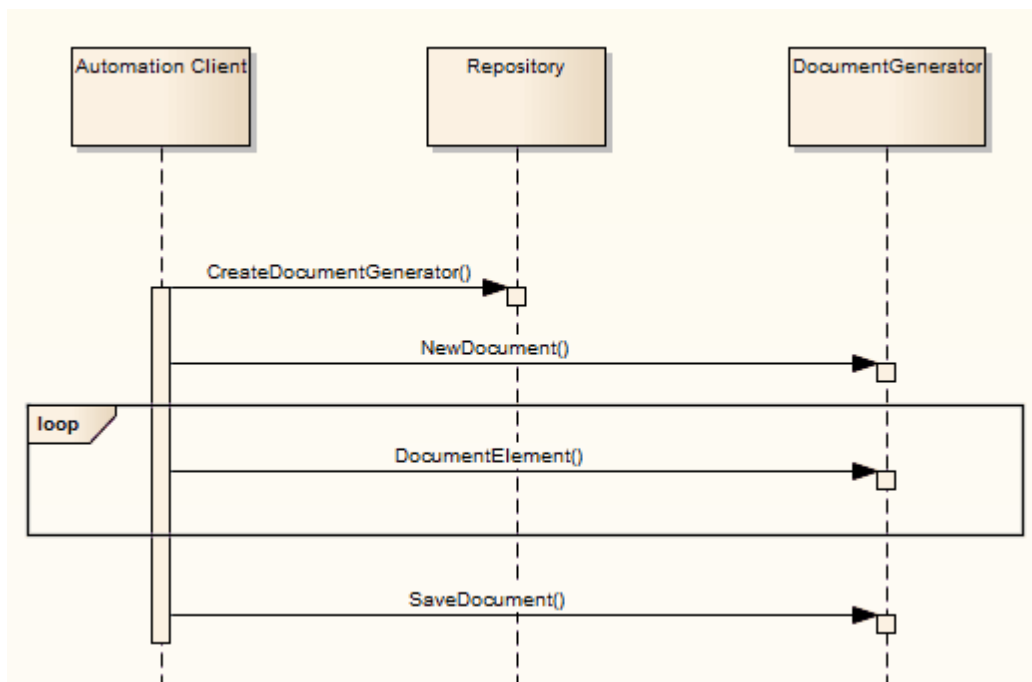
### 20.2.2.10 Document Generator Interface Package

The *DocumentGenerator* Class provides an interface to the document and web reporting facilities, which you can use to generate reports on specific packages, diagrams and elements in your model.

**Access** You can create a pointer to this interface using the *Repository.CreateDocumentGenerator* method

#### Example

This diagram illustrates how you might use the Document Generator interface in generating a report through the Automation Interface.



#### Learn more

- [Repository Class](#) <sup>[2850]</sup>
- [DocumentGenerator Class](#) <sup>[2985]</sup>

#### 20.2.2.10.1 DocumentGenerator Class

The **DocumentGenerator** Class provides an interface to the document and web reporting facilities, which you can use to generate reports on specific packages, diagrams and elements in your model. This Class is accessed from the **Repository** Class using the **CreateDocumentGenerator()** method.

#### DocumentGenerator Attributes

Attribute	Type	Notes	See Also
<b>ObjectType</b>	<a href="#">ObjectType</a> <sup>[2822]</sup>	Read only Distinguishes objects referenced through a	


		Dispatch interface.	
--	--	---------------------	--

**DocumentGenerator Methods**

Method	Type	Notes	See Also
<b>DocumentDiagram (</b> <b>long diagramID,</b> <b>long nDepth,</b> <b>string templateName)</b>	<i>Boolean</i>	Documents a diagram.  Parameters: <ul style="list-style-type: none"> <li>• diagramId: Long - the ID of the diagram</li> <li>• nDepth: Long - the depth by which to adjust the heading level</li> <li>• templateName: String - the name of a template to use when documenting diagrams; this can be blank</li> </ul>	
<b>DocumentElement (</b> <b>long elementID,</b> <b>long nDepth,</b> <b>string templateName)</b>	<i>Boolean</i>	Documents an element.  Parameters: <ul style="list-style-type: none"> <li>• elementId: Long - the ID of the element</li> <li>• nDepth: Long - the depth by which to adjust the heading level</li> <li>• templateName: String - the name of a template to use when documenting elements; this can be blank</li> </ul>	
<b>DocumentModelAuthor (</b> <b>string name,</b> <b>long nDepth,</b> <b>string templateName)</b>	<i>Boolean</i>	Documents a model author.  Parameters: <ul style="list-style-type: none"> <li>• name: String - the name of the author</li> <li>• nDepth: Long - the depth by which to adjust the heading level</li> <li>• templateName: String - a template to use when documenting model authors; this can be blank</li> </ul>	
<b>DocumentModelClient (</b> <b>string name,</b> <b>long nDepth,</b> <b>string templateName)</b>	<i>Boolean</i>	Documents a single model client.  Parameters: <ul style="list-style-type: none"> <li>• name: String - the name of the client</li> <li>• nDepth: Long - the depth by which to adjust the heading level</li> <li>• templateName: String - a template to use when documenting model clients; this can be blank</li> </ul>	
<b>DocumentModelGlossary (</b> <b>long id,</b>	<i>Boolean</i>	Documents a single model glossary term.  Parameters:	

<b>long nDepth, string templateName)</b>		<ul style="list-style-type: none"> <li>• id: Long - the ID of the term</li> <li>• nDepth: Long - the depth by which to adjust the heading level</li> <li>• templateName: String - a template to use when documenting model glossary terms; this can be blank</li> </ul>	
<b>DocumentModelIssue (</b> <b>long id,</b> <b>long nDepth,</b> <b>string templateName)</b>	<i>Boolean</i>	Documents a single model issue. Parameters: <ul style="list-style-type: none"> <li>• id: Long - the ID of the issue</li> <li>• nDepth: Long - the depth by which to adjust the heading level</li> <li>• templateName: String - a template to use when documenting model issues; this can be blank</li> </ul>	
<b>DocumentModelResource (</b> <b>string name,</b> <b>long nDepth,</b> <b>string templateName)</b>	<i>Boolean</i>	Documents a single model resource. Parameters: <ul style="list-style-type: none"> <li>• name: String - the name of the resource</li> <li>• nDepth: Long - the depth by which to adjust the heading level</li> <li>• templateName: String - a template to use when documenting model resources; this can be blank</li> </ul>	
<b>DocumentModelRole (</b> <b>string name,</b> <b>long nDepth,</b> <b>string templateName)</b>	<i>Boolean</i>	Documents a single model role. Parameters: <ul style="list-style-type: none"> <li>• name: String - the name of the role</li> <li>• nDepth: Long - the depth by which to adjust the heading level</li> <li>• templateName: String - a template to use when documenting model roles; this can be blank</li> </ul>	
<b>DocumentModelTask (</b> <b>long id,</b> <b>long nDepth,</b> <b>string templateName)</b>	<i>Boolean</i>	Documents a single model task. Parameters: <ul style="list-style-type: none"> <li>• id: Long - the ID of the the task</li> <li>• nDepth: Long - the depth by which to adjust the heading level</li> <li>• templateName: String - a template to use when documenting model tasks; this can be blank</li> </ul>	
<b>DocumentPackage (</b> <b>long packageID,</b> <b>long nDepth,</b> <b>string templateName)</b>	<i>Boolean</i>	Documents a package. Parameters: <ul style="list-style-type: none"> <li>• packageID: Long - the ID of the package</li> </ul>	

		<ul style="list-style-type: none"> <li>nDepth: Long - the depth by which to adjust the heading level</li> <li>templateName: String - a template to use when documenting packages; this can be blank</li> </ul>	
<b>GetLastError()</b>	<i>String</i>	Returns a string value describing the most recent error that occurred in relation to this object.	
<b>InsertBreak (long breakType)</b>	<i>Boolean</i>	<p>Inserts a break into the report at the current location.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>breakType: Long - <b>0</b> = page break, <b>1</b> = section break</li> </ul>	<a href="#">DocumentBreak</a> <small>[2818]</small>
<b>InsertCoverPageDocument (string name)</b>	<i>Boolean</i>	<p>Inserts the Coverpage into the document at the current location.</p> <p>Note: The style sheet is applied to the document before it is insert into the generated document.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>name: String - the name of the Cover page document found in the Resource tree</li> </ul>	
<b>InsertLinkedDocument (string guid)</b>	<i>Boolean</i>	<p>Inserts a linked document into the report at the current location.</p> <p>A linked document can used to set the header and footer of the report. These are taken from the first linked document added to the report.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>guid: String - the GUID of the element that has a linked document</li> </ul>	
<b>InsertTableOfContents</b>	<i>Boolean</i>	Inserts a Table of Contents at the current position.	
<b>InsertTeamReviewPost (string path)</b>	<i>Boolean</i>	<p>Inserts a Team Review posting into the report at the current location.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>path: String - the path of the Team Review post</li> </ul>	
<b>InsertText (string text,</b>	<i>Boolean</i>	Inserts static text into the report at the current location.	

<b>string style)</b>		<p>A carriage return is not included; if you need to use one, you can add it manually.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• text: String - the static text to be inserted</li> <li>• style: String - the name of the style in the template; defaults to <i>Normal</i> style</li> </ul>	
<b>InsertTOCDocument (string name)</b>	<i>Boolean</i>	<p>Inserts the Table of Contents into the document at the current location.</p> <p>Note: The stylesheet is applied to the document before it is insert into the generated document.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• name: String - the name of the Table of Contents document found in the Resource tree</li> </ul>	
<b>NewDocument (string templateName)</b>	<i>Boolean</i>	<p>Starts a new document; you call this before attempting to document anything else.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• templateName: String - the name of a template to use when documenting elements; this can be blank</li> </ul>	
<b>ReplaceField (string fieldname, string fieldvalue)</b>	<i>Boolean</i>	<p>Replaces the <b>Section</b> field identified by the <i>fieldname</i> parameter with the value provided in <i>fieldvalue</i>. For example:</p> <pre>ReplaceField ( " Element . Alias " , " MyAl i as " )</pre> <p>If you call this function more than once with the same <i>fieldname</i>, the field only has the most recent value set.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• fieldname: String - the field name to find (this does not include the {} braces)</li> <li>• fieldvalue: String - the value to insert into the field; this can be a constant or a derived value</li> </ul>	
<b>SaveDocument (string filename, long nDocType)</b>	<i>Boolean</i>	<p>Saves the document to disk.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li>• filename: String - the filename to save the file to</li> <li>• nDocType: Long - <b>0</b> = RTF, <b>1</b> = HTML, <b>2</b> = PDF, <b>3</b> = DOCX</li> </ul>	<a href="#">DocumentType</a>  2819

<b>SetPageOrientation (long pageOrientation)</b>	<i>Boolean</i>	Sets the current page orientation.  Parameters: <ul style="list-style-type: none"> <li>pageOrientation: Long - <b>0</b> = Portrait, <b>1</b> = Landscape</li> </ul>	<a href="#">DocumentPageOrientation</a> <sup>[2816]</sup>
<b>SetStyleSheetDocument (string name)</b>	<i>Boolean</i>	Sets the Stylesheet to be used for TOC, Coverpage and templates used. This can be called before <b>NewDocument</b> (above).  Parameters: <ul style="list-style-type: none"> <li>name: String - the name of the stylesheet found in the Resource tree</li> </ul>	

Learn more

- [Reporting](#) <sup>[2638]</sup>
- [Repository Class](#) <sup>[2850]</sup>

**20.2.2.11 Mail Interface Package**

The **MailInterface** Package contains:

- A function to retrieve a pointer to the interface
- Functions to create and send a mail message within the current mode
- Utility functions for creating hyperlinks to selected model elements

You can get a pointer to this interface using the method *Repository.GetMailInterface*.

Learn more

- [Repository Package](#) <sup>[2826]</sup>
- [MailInterface Class](#) <sup>[2990]</sup>

**20.2.2.11.1 MailInterface Class**

The **MailInterface** interface can be accessed from the Repository using *GetMailInterface()*. The returned interface provides access to the Enterprise Architect Model Mail Interface. Use this interface to automate the process of creating and sending messages using Enterprise Architect's Model Mail system.

MailInterface Attributes

Attribute	Type	Notes	See also
<b>MessagingEnabled</b>	<i>Boolean</i>	Read Only  Advises whether messaging is enabled on the current model.	
<b>ObjectType</b>	<i>ObjectTy</i>	Read Only	<a href="#">ObjectType</a> <sup>[2822]</sup>

Attribute	Type	Notes	See also
	<i>pe</i>	Distinguishes objects referenced through a dispatch interface.	

**MailInterface Methods**

Method	Type	Notes
<b>ComposeMailMessage</b> ( string InitialRecipientGUID, string InitialSubject, messageflag InitialFlag, string InitialMessageText)	<i>Boolean</i>	Creates a new mail message using the values specified in the input parameters; the message is displayed in the composition window, ready for sending.  This method does NOT send the message.  Parameters: <ul style="list-style-type: none"> <li>InitialRecipientGUID: String - Initial value for the GUID of the addressee user (an Enterprise Architect user defined in the current model)</li> <li>InitialSubject: String - Initial value for the Subject text to display for this message</li> <li>InitialFlag: MessageFlag - Initial value for the flag type/color to attach to this message</li> <li>InitialMessageText: String - Initial value for the text that is the body of the message</li> </ul>
<b>GetAttributeHyperlink</b> ( string AttributeGUID, string LinkText)	<i>String</i>	Returns a string containing a hyperlink to the attribute specified by the input parameter AttributeGUID.  Parameters: <ul style="list-style-type: none"> <li>AttributeGUID: String - The GUID of the attribute for which a hyperlink is required</li> <li>LinkText: String - The text to display for the hyperlink (e.g. the attribute name)</li> </ul>
<b>GetDiagramHyperlink</b> ( string DiagramGUID, string LinkText)	<i>String</i>	Returns a string containing a hyperlink to the diagram specified by the input parameter DiagramGUID.  Parameters: <ul style="list-style-type: none"> <li>DiagramGUID: String - The GUID of the diagram for which a hyperlink is required</li> <li>LinkText: String - The text to display for the hyperlink (e.g. the diagram name)</li> </ul>
<b>GetElementHyperlink</b> ( string ElementGUID, string LinkText)	<i>String</i>	Returns a string containing a hyperlink to the element specified by the input parameter ElementGUID.  Parameters: <ul style="list-style-type: none"> <li>ElementGUID: String - The GUID of the element for which a hyperlink is required</li> <li>LinkText: String - The text to display for the hyperlink (e.g. the</li> </ul>

Method	Type	Notes
		element name)
<b>GetFileHyperlink( string FilePath, string LinkText)</b>	<i>String</i>	Returns a string containing a hyperlink to the file specified by the input parameter FilePath.  Parameters: <ul style="list-style-type: none"> <li>• FilePath: String - The path name of the file for which a hyperlink is required</li> <li>• LinkText: String - The text to display for the hyperlink (e.g. the file name)</li> </ul>
<b>GetLastError( )</b>	<i>String</i>	Returns the last error message set for the MailInterface.
<b>GetMethodHyperlink( string MethodGUID, string LinkText)</b>	<i>String</i>	Returns a string containing a hyperlink to the method specified by the input parameter MethodGUID.  Parameters: <ul style="list-style-type: none"> <li>• MethodGUID: String - The GUID of the method for which a hyperlink is required</li> <li>• LinkText: String - The text to display for the hyperlink (e.g. the method name)</li> </ul>
<b>GetPackageHyperlink( string PackageGUID, string LinkText)</b>	<i>String</i>	Returns a string containing a hyperlink to the package specified by the input parameter PackageGUID.  Parameters: <ul style="list-style-type: none"> <li>• PackageGUID: String - The GUID of the package for which a hyperlink is required</li> <li>• LinkText: String - The text to display for the hyperlink (e.g. the package name)</li> </ul>
<b>GetRecipientGUID( string UserName)</b>	<i>String</i>	Returns the GUID of the specified Enterprise Architect user.  Parameters: <ul style="list-style-type: none"> <li>• UserName: String - The name of a user defined in the current model</li> </ul>
<b>GetWebHyperlink( string URL, string LinkText)</b>	<i>String</i>	Returns a string containing a hyperlink to the URL specified by the input parameter URL.  Parameters: <ul style="list-style-type: none"> <li>• URL: String - The URL of the item for which a hyperlink is required</li> <li>• LinkText: String - The text to display for the hyperlink</li> </ul>
<b>SendMailMessage( string RecipientGUID, string Subject,</b>	<i>Boolean</i>	Creates and sends a new mail message using the values specified in the input parameters.  Parameters:



Method	Type	Notes
<b>messageflag Flag, string MessageText)</b>		<ul style="list-style-type: none"> <li>RecipientGUID: String - The GUID of the addressee user (an Enterprise Architect user defined in the current model)</li> <li>Subject: String - The Subject text to display for this message</li> <li>Flag: MessageFlag - The flag type/color to attach to this message</li> <li>MessageText: String - The text that is the body of the message</li> </ul>

### 20.2.2.12 Simulation Package

The **Simulation** Package contains:

- An attribute to set, increase and decrease the speed of the simulation
- A function to check if a simulation is currently running
- Functions to Start, Stop, Step Into, Step Out of and Step Over and Pause a simulation
- A function to send a broadcast signal to the simulation that is currently running

#### Learn more

- [Simulation Class](#) <sup>[2993]</sup>
- [Model Simulation](#) <sup>[2463]</sup>

#### 20.2.2.12.1 Simulation Class


The **Simulation** Class provides an interface to the Enterprise Architect Model Simulation facilities.

#### Simulation Attributes

Attribute	Type	Notes	See Also
<b>ObjectType</b>	<i>ObjectType</i>	Read only Distinguishes objects referenced through a Dispatch interface.	<a href="#">ObjectType</a> <sup>[2822]</sup>
<b>Speed</b>	<i>Long</i>	Read/Write Retrieve or set the current simulation running speed.	<a href="#">Run Model Simulation</a> <sup>[2475]</sup>

#### Simulation Methods

Method	Type	Notes	See Also
<b>BroadcastSignal ( string sSignalName,</b>	<i>Boolean</i>	Send a signal into the running simulation. If the simulation is stopped, do nothing.	<a href="#">Dynamic Simulation with Javascript</a> <sup>[2486]</sup>



<b>string sParameters)</b>		Parameters: <ul style="list-style-type: none"> <li>sSignalName: String - the name of the signal OR the GUID of the Signal element</li> <li>sParameters: String - a string of one or more signal parameters, in the following format  <pre>{parameter1 : 5, parameter2 : "test", parameter3 : 3.2}</pre> </li> </ul>	
<b>IsSimulatorRunning ()</b>	<i>Boolean</i>	Check the state of the simulation.  Returns <b>true</b> if the simulation is running; returns <b>false</b> if the simulation is stopped.	
<b>Pause ()</b>	<i>Boolean</i>	Pause the simulation if it is running.	<a href="#">Run Model Simulation</a>  <sup>[2475]</sup>
<b>Start ()</b>	<i>Boolean</i>	Start the simulation based on the current selection. If the current simulation is in a paused state, then the simulation is resumed.	
<b>StepIn ()</b>	<i>Boolean</i>	Step In to the routine in the current simulation.	
<b>StepOut ()</b>	<i>Boolean</i>	Step Out of the routine in the current simulation.	
<b>StepOver ()</b>	<i>Boolean</i>	Step Over the routine in the current simulation.	
<b>Stop ()</b>	<i>Boolean</i>	Stop the simulation.	

#### Learn more

- [Model Simulation](#)  <sup>[2463]</sup>

### 20.2.2.13 Code Samples

As you write or edit code for using the Automation Interface, you might want to review these public Object examples, written in VB.Net.

Example	See also
Open the Repository	<a href="#">Open the Repository</a>  <sup>[2995]</sup>
Iterate Through a .EAP File	<a href="#">Iterate Through a .EAP File</a>  <sup>[2996]</sup>

Example	See also
Add and Manage Packages	<a href="#">Add and Manage Packages</a> <sup>[2996]</sup>
Add and Manage Elements	<a href="#">Add and Manage Elements</a> <sup>[2997]</sup>
Add a Connector	<a href="#">Add a Connector</a> <sup>[2998]</sup>
Add and Manage Diagrams	<a href="#">Add and Manage Diagrams</a> <sup>[2999]</sup>
Add and Delete Features	<a href="#">Add and Delete Features</a> <sup>[3000]</sup>
Element Extras	<a href="#">Element Extras</a> <sup>[3001]</sup>
Repository Extras	<a href="#">Repository Extras</a> <sup>[3004]</sup>
Stereotypes	<a href="#">Stereotypes</a> <sup>[3006]</sup>
Work with Attributes	<a href="#">Work with Attributes</a> <sup>[3006]</sup>
Work with Methods	<a href="#">Work with Methods</a> <sup>[3007]</sup>

### 20.2.2.13.1 Open the Repository

An example of the VB.Net code to open an Enterprise Architect repository.

```
Public Class AutomationExample
    'class level variable for Repository
    Public m_Repository As Object

    Public Sub Run()
        try
            'create the repository object
            m_Repository = CreateObject("EA.Repository")

            'open an EAP file
            m_Repository.OpenFile("F:\Test\EAAuto.EAP")
            'use the Repository in any way required
            DumpModel

            'close the repository and tidy up
            m_Repository.Exit()
            m_Repository = Nothing

            ....catch e as exception
        end try
    end Sub
end Class
```

```

        Console.WriteLine(e)
    End try
End Sub
end Class

```

### 20.2.2.13.2 Iterate Through a .EAP File

An example of the VB.Net code to iterate through a .EAP file starting at the Model level, after the repository has been opened.

```

Sub DumpModel()
    Dim idx as Integer
    For idx=0 to m_Repository.Models.Count-1
        DumpPackage("", m_Repository.Models.GetAt(idx))
    Next
End Sub

' output package name, then element contents, then process child packages
Sub DumpPackage(Indent as String, Package as Object)
    Dim idx as Integer
    Console.WriteLine(Indent + Package.Name)
    DumpElements(Indent + "    ", Package)

    For idx = 0 to Package.Packages.Count-1
        DumpPackage(Indent + "    ", Package.Packages.GetAt(idx))
    Next
End Sub

'' dump element name
Sub DumpElements(Indent as String, Package as Object)
    Dim idx as Integer
    For idx = 0 to Package.Elements.Count-1
        Console.WriteLine(Indent + "::" + Package.Elements.GetAt(idx).
Name)
    Next
End Sub

```

### 20.2.2.13.3 Add and Manage Packages

#### Topics

Topic	Detail
<b>public Object</b>	<p>Example illustrating how to add a Model or a Package.</p> <pre> Sub TestPackageLifecycle      Dim idx as integer     Dim idx2 as integer     Dim package as object     Dim model as object     Dim o as object      ''first add a new Model      model = m_Repository.Models.AddNew ("AdvancedModel", "")     If not model.Update() Then         Console.WriteLine(model.GetLastError())     End If      ''refresh the models collection     m_Repository.Models.Refresh </pre>

Topic	Detail
	<pre> '' now work through models collection and add a package  For idx = 0 to m_Repository.Models.Count - 1     o = m_Repository.Models.GetAt(idx)     Console.WriteLine(o.Name)     If o.Name = "AdvancedModel" Then         package = o.Packages.Addnew ("Subpackage", "Nothing")         If not package.Update() Then             Console.WriteLine(package. GetLastError())         End If          package.Element.Stereotype = "system"         package.Update  '' for testing purposes just delete the '' newly created Model and its contents m_Repository.Models.Delete(idx)      End If Next  End Sub </pre>

#### 20.2.2.13.4 Add and Manage Elements

##### Topics

Topic	Detail
<b>public Object</b>	<pre> '' Add and delete elements in a package.  Sub ElementLifecycle      Dim package as Object     Dim element as Object      package = m_Repository.GetPackageByID(2)     element = package.elements.AddNew("Login to Website", "UseCase")     element.Stereotype = "testcase"     element.Update     package.elements.Refresh()      Dim idx as integer      '' note the repeated calls to "package.elements. GetAt"     '' in general you should make this call once and assign to a local     '' variable - in the example below, Enterprise Architect loads the     '' element required every time a call is made - rather than loading once     '' and keeping a local reference. </pre>

Topic	Detail
	<pre>         For idx = 0 to package.elements.count - 1             Console.WriteLine(package.elements.GetAt(idx). Name)             If (package.elements.GetAt(idx).Name = "Login to Website" and _                 package.elements.GetAt(idx).Type = "UseCase") Then                 package.elements.deleteat(idx, false)             End If         Next     End Sub </pre>

### 20.2.2.13.5 Add a Connector

#### Topics

Topic	Detail
<b>public Object</b>	<pre> "Add a connector and set values.  Sub ConnectorTest      Dim source as object     Dim target as object     Dim con as object     Dim o as object      Dim client as object     Dim supplier as object      ''use ElementIDs to quickly load an element in this example     ''... you must find suitable IDs in your model      source = m_Repository.GetElementByID(129)     target = m_Repository.GetElementByID(169)      con = source.Connectors.AddNew("test link 2", "Association")      ''again- replace ID with a suitable one from your model     con.SupplierID = 169      If not con.Update Then         Console.WriteLine(con.GetLastError)     End If     source.Connectors.Refresh      Console.WriteLine("Connector Created")      o = con.Constraints.AddNew("constraint 2", "type")     If not o.Update Then         Console.WriteLine(o.GetLastError)     End If      o = con.TaggedValues.AddNew("Tag", "Value")     If not o.Update Then         Console.WriteLine(o.GetLastError)     End If </pre>

Topic	Detail
	<pre> '' use the client and supplier ends to set '' additional information  client = con.ClientEnd client.Visibility = "Private" client.Role = "m_client" client.Update supplier = con.SupplierEnd supplier.Visibility = "Protected" supplier.Role = "m_supplier" supplier.Update  Console.WriteLine("Client and Supplier set")  Console.WriteLine(client.Role) Console.WriteLine(supplier.Role)  End Sub </pre>

### 20.2.2.13.6 Add and Manage Diagrams

#### Topics

Topic	Detail
<b>public Object</b>	<pre> '' An example of how to create a diagram and add an element to it. '' Note the optional use of element rectangle setting using '' left, right, top and bottom dimensions in AddNew call.  Sub DiagramLifeCycle      Dim diagram as object     Dim v as object     Dim o as object     Dim package as object      Dim idx as Integer     Dim idx2 as integer      package = m_Repository.GetPackageById(5)      diagram = package.Diagrams.AddNew("Logical Diagram", "Logical")     If not diagram.Update Then         Console.WriteLine(diagram.GetLastError)     End if      diagram.Notes = "Hello there this is a test"     diagram.update()      o = package.Elements.AddNew ("ReferenceType", "Class")     o.Update      '' add element to diagram - supply optional rectangle co-ordinates </pre>

Topic	Detail
	<pre> v = diagram.DiagramObjects.AddNew("l=200;r=400; t=200;b=600;", "") v.ElementID = o.ElementID v.Update  Console.WriteLine(diagram.DiagramID)  End Sub </pre>

### 20.2.2.13.7 Add and Delete Features

#### Topics

Topic	Detail
<b>public Object</b>	<pre> Dim element as object Dim idx as integer Dim attribute as object Dim method as object  'just load an element by ID - you must 'substitute a valid ID from your model element = m_Repository.GetElementByID(246)  'create a new method method = element.Methods.AddNew("newMethod", "int") method.Update element.Methods.Refresh  'now loop through methods for Element - and delete our addition For idx = 0 to element.Methods.Count - 1     method = element.Methods.GetAt(idx)     Console.WriteLine(method.Name)     If (method.Name = "newMethod") Then         element.Methods.Delete(idx)     End If Next  'create an attribute attribute = element.attributes.AddNew("NewAttribute", "int") attribute.Update element.attributes.Refresh  'loop through and delete our new attribute For idx = 0 to element.attributes.Count - 1     attribute = element.attributes.GetAt(idx)     Console.WriteLine(attribute.Name)     If (attribute.Name = "NewAttribute") Then         element.attributes.Delete(idx)     End If Next </pre>



20.2.2.13.8 *Element Extras*Topics

Topic	Detail
<b>public Object</b>	<pre> '' Examples of how to access and use element extras, such as '' scenarios, constraints and requirements.  Sub ElementExtras      Dim element as object     Dim o as object     Dim idx as Integer     Dim bDel as boolean     bDel = true      try         element = m_Repository.GetElementById(129)          ' manage constraints for an element         ' demonstrate addnew and delete         o = element.Constraints.AddNew ("Appended", "Type")         If not o.Update Then             Console.WriteLine("Constraint error:" + o. GetLastError())         End if         element.Constraints.Refresh         For idx = 0 to element.Constraints.Count - 1             o = element.Constraints.GetAt(idx)             Console.WriteLine(o.Name)             If (o.Name="Appended") Then                 If bDel Then element.Constraints.Delete (idx)             End if         Next          ' efforts         o = element.Efforts.AddNew("Appended", "Type")         If not o.Update Then             Console.WriteLine("Efforts error:" + o. GetLastError())         End if         element.Efforts.Refresh         For idx = 0 to element.Efforts.Count - 1             o = element.Efforts.GetAt(idx)             Console.WriteLine(o.Name)             If (o.Name="Appended") Then                 If bDel Then element.Efforts.Delete (idx)             End if         Next          ' Risks         o = element.Risks.AddNew("Appended", "Type")         If not o.Update Then             Console.WriteLine("Risks error:" + o. GetLastError())         End if         element.Risks.Refresh         For idx = 0 to element.Risks.Count - 1             o = element.Risks.GetAt(idx) </pre>

Topic	Detail
	<pre> Console.WriteLine(o.Name) If(o.Name="Appended") Then     If bDel Then element.Risks.Delete(idx) End if Next  ' Metrics o = element.Metrics.AddNew("Appended","Change") If not o.Update Then     Console.WriteLine("Metrics error:" + o. GetLastError()) End if element.Metrics.Refresh For idx = 0 to element.Metrics.Count - 1     o = element.Metrics.GetAt(idx)     Console.WriteLine(o.Name)     If(o.Name="Appended") Then         If bDel Then element.Metrics.Delete (idx)     End if Next  ' TaggedValues o = element.TaggedValues.AddNew ("Appended","Change") If not o.Update Then     Console.WriteLine("TaggedValues error:" + o.GetLastError()) End if element.TaggedValues.Refresh For idx = 0 to element.TaggedValues.Count - 1     o = element.TaggedValues.GetAt(idx)     Console.WriteLine(o.Name)     If(o.Name="Appended") Then         If bDel Then element.TaggedValues. Delete(idx)     End if Next  ' Scenarios o = element.Scenarios.AddNew ("Appended","Change") If not o.Update Then     Console.WriteLine("Scenarios error:" + o. GetLastError()) End if element.Scenarios.Refresh For idx = 0 to element.Scenarios.Count - 1     o = element.Scenarios.GetAt(idx)     Console.WriteLine(o.Name)     If(o.Name="Appended") Then         If bDel Then element.Scenarios.Delete (idx)     End if Next  ' Files o = element.Files.AddNew("MyFile","doc") If not o.Update Then     Console.WriteLine("Files error:" + o. GetLastError()) End if element.Files.Refresh For idx = 0 to element.Files.Count - 1     o = element.Files.GetAt(idx) </pre>

Topic	Detail
	<pre>         Console.WriteLine(o.Name)         If (o.Name="MyFile") Then             If bDel Then element.Files.Delete (idx)         End if     Next      ' Tests     o = element.Tests.AddNew("Test Plan", "Load")     If not o.Update Then         Console.WriteLine("Tests error:" + o. GetLastError())     End if     element.Tests.Refresh     For idx = 0 to element.Tests.Count - 1         o = element.Tests.GetAt(idx)         Console.WriteLine(o.Name)         If (o.Name="Test Plan") Then             If bDel Then element.Tests.Delete (idx)         End if     Next      ' Defect     o = element.Issues.AddNew("Broken", "Defect")     If not o.Update Then         Console.WriteLine("Issues error:" + o. GetLastError())     End if     element.Issues.Refresh     For idx = 0 to element.Issues.Count - 1         o = element.Issues.GetAt(idx)         Console.WriteLine(o.Name)         If (o.Name="Broken") Then             If bDel Then element.Issues.Delete (i dx)         End if     Next      ' Change     o = element.Issues.AddNew("Change", "Change")     If not o.Update Then         Console.WriteLine("Issues error:" + o. GetLastError())     End if     element.Issues.Refresh     For idx = 0 to element.Issues.Count - 1         o = element.Issues.GetAt(idx)         Console.WriteLine(o.Name)         If (o.Name="Change") Then             If bDel Then element.Issues.Delete (i dx)         End if     Next      catch e as exception         Console.WriteLine(element.Methods.GetLastError ())         Console.WriteLine(e)     End try  End Sub </pre>

## 20.2.2.13.9 Repository Extras

Topics

Topic	Detail
<b>public Object</b>	<pre> '' Examples of how to access repository '' collections for system level information.  Sub RepositoryExtras      Dim o as object     Dim idx as integer      ' issues     o = m_Repository.Issues.AddNew("Problem", "Type")     If (o.Update=false) Then         Console.WriteLine(o.GetLastError())     End if     o = nothing     m_Repository.Issues.Refresh     For idx = 0 to m_Repository.Issues.Count - 1         Console.WriteLine(m_Repository.Issues.GetAt (idx).Name)         If (m_Repository.Issues.GetAt(idx).Name = "Problem") then             m_Repository.Issues.DeleteAt(idx,false)             Console.WriteLine("Delete Issues")         End if     Next      ' tasks     o = m_Repository.Tasks.AddNew("Task 1", "Task type")     If (o.Update=false) Then         Console.WriteLine("error - " + o.GetLastError ())     End if     o = nothing     m_Repository.Tasks.Refresh     For idx = 0 to m_Repository.Tasks.Count - 1         Console.WriteLine(m_Repository.Tasks.GetAt (idx).Name)         If (m_Repository.Tasks.GetAt(idx).Name = "Task 1") then             m_Repository.Tasks.DeleteAt(idx,false)             Console.WriteLine("Delete Tasks")         End if     Next      ' glossary     o = m_Repository.Terms.AddNew("Term 1", "business")     If (o.Update=false) Then         Console.WriteLine("error - " + o.GetLastError ())     End if     o = nothing     m_Repository.Terms.Refresh     For idx = 0 to m_Repository.Terms.Count - 1         Console.WriteLine(m_Repository.Terms.GetAt (idx).Term)         If (m_Repository.Terms.GetAt(idx).Term = "Term 1") then             m_Repository.Terms.DeleteAt(idx,false)             Console.WriteLine("Delete Terms")         End if     Next </pre>

Topic	Detail
	<pre> ' authors o = m_Repository.Authors.AddNew("Joe B", "Writer") If (o.Update=false) Then     Console.WriteLine(o.GetLastError()) End if o = nothing m_Repository.Authors.Refresh For idx = 0 to m_Repository.authors.Count - 1     Console.WriteLine(m_Repository.Authors.GetAt (idx).Name)     If (m_Repository.authors.GetAt(idx).Name = "Joe B") then         m_Repository.authors.DeleteAt(idx, false)         Console.WriteLine("Delete Authors")     End if Next  o = m_Repository.Clients.AddNew("Joe Sphere", "Client") If (o.Update=false) Then     Console.WriteLine(o.GetLastError()) End if o = nothing m_Repository.Clients.Refresh For idx = 0 to m_Repository.Clients.Count - 1     Console.WriteLine(m_Repository.Clients.GetAt (idx).Name)     If (m_Repository.Clients.GetAt(idx).Name = "Joe Sphere") then         m_Repository.Clients.DeleteAt(idx, false)         Console.WriteLine("Delete Clients")     End if Next  o = m_Repository.Resources.AddNew("Joe Worker", "Resource") If (o.Update=false) Then     Console.WriteLine(o.GetLastError()) End if o = nothing m_Repository.Resources.Refresh For idx = 0 to m_Repository.Resources.Count - 1     Console.WriteLine(m_Repository.Resources.GetAt (idx).Name)     If (m_Repository.Resources.GetAt(idx).Name = "Joe Worker") then         m_Repository.Resources.DeleteAt(idx, false)         Console.WriteLine("Delete Resources")     End if Next  End Sub </pre>

### 20.2.2.13.10 Stereotypes

#### Topics

Topic	Detail
<b>public Object</b>	<pre> Sub Test Stereotypes      Dim o as object     Dim idx as integer      '' add a new stereotype to the Stereotypes collection     o = m_Repository.Stereotypes.AddNew ("funky", "class")     If (o.Update=false) Then         Console.WriteLine (o.GetLastError())     End if     o = nothing      '' make sure you refresh     m_Repository.Stereotypes.Refresh      '' then iterate through - deleting our new entry in the process     For idx = 0 to m_Repository.Stereotypes.Count - 1         Console.WriteLine(m_Repository.Stereotypes. GetAt(idx).Name)         If (m_Repository.Stereotypes.GetAt(idx).Name = "funky") then             m_Repository.Stereotypes.DeleteAt(idx, false)             Console.WriteLine("Delete element")         End if     Next  End Sub </pre>

### 20.2.2.13.11 Work With Attributes

#### Topics

Topic	Detail
<b>public Object</b>	<pre> '' An example of working with attributes.  Sub AttributeLifecycle      Dim element as object     Dim o as object     Dim t as object     Dim idx as Integer     Dim idx2 as integer     try         element = m_Repository.GetElementByID(129)          For idx = 0 to element.Attributes.Count - 1              Console.WriteLine("attribute=" + element. Attributes.GetAt(idx).Name) </pre>

Topic	Detail
	<pre> o = element.Attributes.GetAt(idx) t = o.Constraints.AddNew("&gt; 123", "Precision") t.Update() o.Constraints.Refresh For idx2 = 0 to o.Constraints.Count - 1     t = o.Constraints.GetAt(idx2)     Console.WriteLine("Constraint: " + t. Name)     If (t.Name = "&gt; 123") Then         o.Constraints.DeleteAt(idx2, false)     End If Next  For idx2 = 0 to o.TaggedValues.Count - 1     t = o.TaggedValues.GetAt(idx2)     If (t.Name = "Type2") Then         Console.WriteLine("deleting")         o.TaggedValues.DeleteAt(idx2, true)     End If Next  t = o.TaggedValues.AddNew("Type2", "Number") t.Update o.TaggedValues.Refresh For idx2 = 0 to o.TaggedValues.Count - 1     t = o.TaggedValues.GetAt(idx2)     Console.WriteLine("Tagged Value: " + t. Name) Next  If (element.Attributes.GetAt(idx).Name = "m_Toolle") Then     Console.WriteLine("delete attribute")     element.Attributes.DeleteAt(idx, false) End If  Next  catch exception     Console.WriteLine(element.Attributes. GetLastError())     Console.WriteLine(e) End try End Sub </pre>

### 20.2.2.13.12 Work With Methods

#### Topics

Topic	Detail
<b>public Object</b>	<pre> '' An example of working with the Methods collection '' of an element - and with Method collections.  Sub MethodLifeCycle      Dim element as object </pre>

Topic	Detail
	<pre> Dim method as object Dim t as object Dim idx as Integer Dim idx2 as integer  try     element = m_Repository.GetElementById(129)      For idx = 0 to element.Methods.Count - 1         method = element.Methods.GetAt(idx)         Console.WriteLine(method.Name)          t = method.PreConditions.AddNew ("Test Constraint", "something")         If t.Update = false Then             Console.WriteLine("PreConditions: " + t.GetLastError)         End if          method.PreConditions.Refresh         For idx2 = 0 to method.PreConditions.Count - 1             t = method.PreConditions.GetAt(idx2)             Console.WriteLine("PreConditions: " + t.Name)              If t.Name = "Test Constraint" Then                 method.PreConditions.DeleteAt(idx2, false)             End If         Next          t = method.PostConditions.AddNew ("Test Constraint", "something")         If t.Update = false Then             Console.WriteLine("PostConditions: " + t.GetLastError)         End if          method.PostConditions.Refresh         For idx2 = 0 to method.PostConditions. Count - 1             t = method.PostConditions.GetAt(idx2)             Console.WriteLine("PostConditions: " + t.Name)              If t.Name = "Test Constraint" Then                 method.PostConditions.DeleteAt (idx2, false)             End If         Next          t = method.TaggedValues.AddNew ("Test TaggedValue", "something")         If t.Update = false Then             Console.WriteLine("Tagged Values: " + t.GetLastError)         End if          For idx2 = 0 to method.TaggedValues.Count - 1             t = method.TaggedValues.GetAt(idx2)             Console.WriteLine("Tagged Value: " + t. Name)              If (t.Name= "Test TaggedValue") Then                 method.TaggedValues.DeleteAt(idx2, false)             End If         Next </pre>



Topic	Detail
	<pre>        t = method.Parameters.AddNew ("Test Param", "string")         If t.Update = false Then             Console.WriteLine("Parameters: " + t. Get Last Error)         End If          method.Parameters.Refresh         For idx2 = 0 to method.Parameters.Count - 1             t = method.Parameters.Get At (idx2)             Console.WriteLine("Parameter: " + t. Name)             If (t.Name="Test Param") Then                 method.Parameters.DeleteAt (idx2, false)             End If         Next          method = nothing     Next catch e as exception     Console.WriteLine(element.Methods.Get Last Error ())     Console.WriteLine(e) End try  End Sub</pre>

## 20.3 Enterprise Architect Add-In Model



### Topics

Topic	Detail	See also
<b>Introduction</b>	<p>Add-Ins enable you to add functionality to Enterprise Architect. The Enterprise Architect Add-In model builds on the features provided by the <b>Automation Interface</b> to enable you to extend the Enterprise Architect user interface.</p> <p>Add-Ins are ActiveX COM objects that expose public Dispatch methods. They have several advantages over stand-alone automation clients:</p> <ul style="list-style-type: none"> <li>• Add-Ins can define Enterprise Architect menus and sub-menus</li> <li>• Add-Ins receive notifications about various Enterprise Architect user-interface events including menu clicks and file changes</li> <li>• Add-Ins can (and should) be written as in-process (DLL) components. This provides lower call overhead and better integration into the Enterprise Architect environment</li> <li>• Because a current version of Enterprise Architect is already running there is no requirement to start a second copy of Enterprise Architect via the automation interface</li> <li>• Because the Add-In receives object handles associated with the currently running copy of Enterprise Architect, more information is available about the current user's activity; for example, which diagram objects are selected</li> <li>• You are not required to do anything other than to install the Add-In to make it usable; that is, you do not have to configure Add-Ins to run on your systems</li> <li>• Because Enterprise Architect is constantly evolving in response to customer requests, the Add-In interface is flexible</li> <li>• The Add-In interface does not have its own version, rather it is identified by the version of Enterprise Architect it first appeared in; for example, the current version of the Enterprise Architect Add-In interface is version 2.1</li> <li>• When creating your Add-In, you do not have to subscribe to a type-library</li> </ul> <p>From Enterprise Architect release 7.0, Add-Ins created before 2004 are no longer supported. If an Add-In subscribes to the Addn_Tmpl.tlb interface (2003 style), it fails on load. In this event, contact the vendor or author of the Add-In and request an upgrade.</p> <ul style="list-style-type: none"> <li>• Add-Ins do not have to implement methods that they never use</li> <li>• Add-Ins prompt users via context menus in the tree view and the</li> </ul>	<p><a href="#">Automation Interface</a> <sup>[2804]</sup></p> <p><a href="#">Scripts</a> <sup>[2791]</sup></p> <p><a href="#">UML Profiles</a> <sup>[1485]</sup></p> <p><a href="#">Register Add-in</a> <sup>[3171]</sup></p> <p><a href="#">Add-In Manager</a> <sup>[3018]</sup></p>

Topic	Detail	See also
	<p>diagram</p> <ul style="list-style-type: none"> <li>Menu check and disable states can be controlled by the Add-In</li> </ul> <p>Add-Ins enhance the existing functionality of Enterprise Architect through a variety of mechanisms such as <b>Scripts</b>, <b>UML Profiles</b> and the <b>Automation Interface</b>. Once an <b>Add-In</b> is registered, it can be managed using the <b>Add-In Manager</b>.</p>	








### Create and Use Add-Ins:

This topic covers the following information on Add-Ins:

Topic	Link
Add-In Tasks	<a href="#">Add-In Tasks</a> 
Add-In Events	<a href="#">Add-In Events</a> 
Broadcast Events	<a href="#">Broadcast Events</a> 
Custom Views	<a href="#">Custom Views</a> 
MDG Add-Ins	<a href="#">MDG Add-Ins</a> 

## 20.3.1 Add-In Tasks

This topic provides instructions on how to create, test, deploy and manage Add-Ins.

Topic	Detail	Link
<b>Create an Add-In</b>		<a href="#">Create an Add-In</a> 
	Define Menu Items.	<a href="#">Define Menu Items</a> 
	Respond to Menu Events.	<a href="#">Respond to Menu Events</a> 
	Handle Add-In Events.	<a href="#">Handle Add-In Events</a> 
<b>Deploy your Add-In</b>		<a href="#">Deploy your Add-In</a> 
	Potential Pitfalls.	<a href="#">Potential Pitfalls</a> 
<b>Manage Add-Ins</b>		
	Register an Add-In(developed in-house or brought-in).	<a href="#">Register an Add-In</a> 

Topic	Detail	Link
	The Add-In Manager.	<a href="#">The Add-In Manager</a> 

### 20.3.1.1 Create Add-Ins




Before you start you must have an application development tool that is capable of creating ActiveX COM objects supporting the IDispatch interface, such as:

- Borland Delphi
- Microsoft Visual Basic
- Microsoft Visual Studio .Net

You should consider how to **define menu items**. To help with this, you could review some **examples of Automation Interfaces** - examples of code used to create Add-Ins for Enterprise Architect - on the Sparx Systems web page.

#### How to

An Enterprise Architect Add-In can be created in four steps:

Step	Action	See also
1	Use a development tool to create an ActiveX COM DLL project. Visual Basic users, for example, choose <i>File&gt;Create New Project-ActiveX DLL</i> .	
2	Connect to the interface using the syntax appropriate to the language.	<a href="#">Connect to the Interface</a> 
3	Create a COM Class and implement each of the <b>general Add-In Events</b> applicable to your Add-In. You only have to define methods for events to respond to.	<a href="#">Add-In Events</a> 
4	Add a registry key that identifies your Add-In to Enterprise Architect, as described in the <i>Deploy Add-Ins</i> topic.	<a href="#">Deploy Add-Ins</a> 

#### Learn more

- [Define Menu Items](#) 
- [Examples of Automation Interfaces](#) (Online Resource)

### 20.3.1.1.1 Define Menu Items

#### Topics

Topic	Detail
<b>Defining Menu Items</b>	<p>Menu items are defined by responding to the <i>GetMenuItems</i> event.</p> <p>The first time this event is called, <i>MenuName</i> is an empty string, representing the top-level menu. For a simple Add-In with just a single menu option you can return a string</p> <pre>Function EA_GetMenuItems(Repository as EA.Repository, MenuLocation As String, MenuName As String) As Variant     EA_GetMenuItems = "&amp;Joe's Add-In" End Function</pre>
<b>Defining Sub-Menus</b>	<p>To define sub-menus, prefix a parent menu with a dash. Parent and sub-items are defined as follows:</p> <pre>Function EA_GetMenuItems(Repository as EA.Repository, MenuLocation As String, MenuName As String) As Variant     Select Case MenuName         Case ""             ' Parent Menu Item             EA_GetMenuItems = "- &amp;Joe's Add-In"         Case "- &amp;Joe's Add-In"             ' Define Sub-Menu Items using the Array notation.             ' In this example, "Diagram" and "Treeview"             compose the "Joe's Add-In" sub-menu.             EA_GetMenuItems = Array("&amp;Diagram", "&amp;Treeview")         Case Else             MsgBox "Invalid Menu", vbCritical         End Select     End Function</pre>
<b>Defining Further Sub-Menus</b>	<p>Similarly, you can define further sub-items:</p> <pre>Function EA_GetMenuItems(Repository as EA.Repository, MenuLocation As String, MenuName As String) As Variant     Select Case MenuName         Case ""             EA_GetMenuItems = "- Joe's Add-In"         Case "- Joe's Add-In"             EA_GetMenuItems = Array("- &amp;Diagram", "&amp;TreeView")         Case "- &amp;Diagram"             EA_GetMenuItems = "&amp;Properties"         Case Else             MsgBox "Invalid Menu", vbCritical         End Select     End Function</pre>
<b>Enabling/Disabling menu options</b>	<p>To enable or disable menu options by default, you can use this method to show particular items to the user:</p> <pre>Sub EA_GetMenuState(Repository As EA.Repository, Location As String, MenuName As String, ItemName As String, IsEnabled As Boolean, IsChecked As Boolean)     Select Case Location</pre>

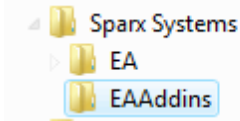
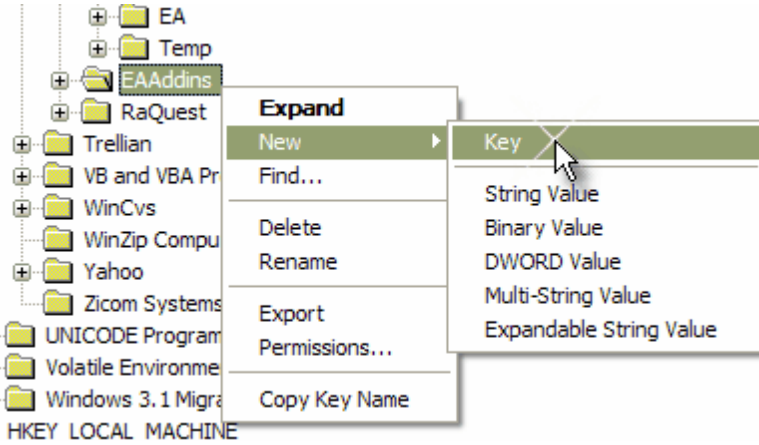
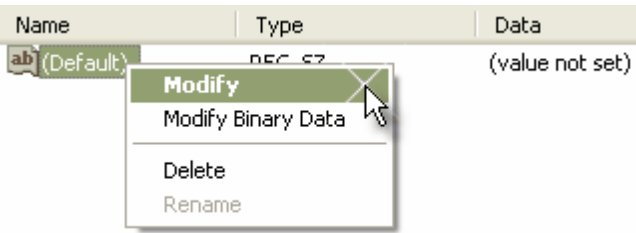
Topic	Detail
	<pre> Case "TreeVie w" ' Al ways enable Case "Di agram" ' Al ways enable Case "Mai nMenu" Select Case ItemName Case "&amp;Tran slate", " Save &amp;Pr oject " If GetIsPr oject Sel ect ed() Then IsEnabled = Fal se End If End Select End Select IsChecked = GetIsCur rent Sel ect ion() End Sub </pre>

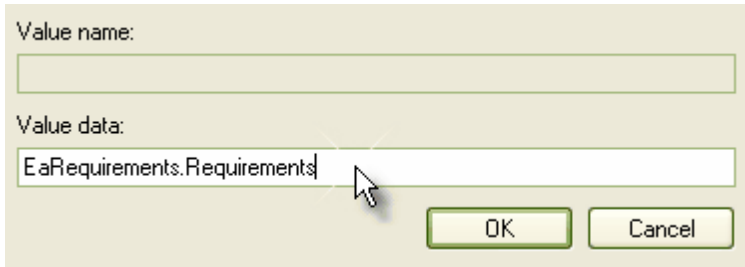
### 20.3.1.1.2 Deploy Add-Ins

#### How to

To deploy Add-Ins to users' sites

Step	Action	See also
1	Add the Add-In DLL file to an appropriate directory on the user's computer; that is: C:\Program Files\ ( new dir )	
2	<p>Register the DLL as appropriate to your platform:</p> <ul style="list-style-type: none"> <li>If compiled as a native Win32 DDL, such as VB6 or C++, register the DDL using the <b>regsvr32</b> command from the command prompt   <pre>regsvr32 "C:\Program Files\MyCompany\EAAddin\EAAddin.dll"</pre> </li> <li>If compiled as a .NET DLL, such as C# or VB.NET, register the DLL using the <b>RegAsm</b> command from the command prompt   <pre>C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727 \RegAsm.exe "C:\Program Files\MyCompany\EAAddin\EAAddin.dll" /codebase</pre> </li> </ul>	
3	Place a new entry into the registry using the registry editor (run <b>regedit</b> ) so that Enterprise Architect recognizes the presence of your Add-In.	
4	<p>Add a new key value <b>EAAddIns</b> under the location:</p> <ul style="list-style-type: none"> <li><i>HKEY_CURRENT_USER\Software\Sparx Systems</i> for single users</li> <li><i>HKEY_LOCAL_MACHINE\Software\Sparx Systems</i> for multiple users on a machine</li> </ul>	

Step	Action	See also
		
5	<p>Add a new key under this key with the project name.</p>  <p>( ProjectName ) is not necessarily the name of your DLL, but the name of the Project; in Visual Basic, this is the value for the property <b>Name</b> corresponding to the project file.</p>	
6	<p>Specify the default value by modifying the default value of the key.</p> 	
7	<p>Enter the value of the key by typing in the ( project name ) . ( class name )</p> <p>EaRequirements. Requirements</p> <p>where <i>EaRequirements</i> is the project name, as shown in the example below:</p>	

Step	Action	See also
		

### 20.3.1.1.3 Tricks and Traps

#### Topics

Topic	Detail	See also
<b>Visual Basic 5/6 Users Note</b>	<p>Visual Basic 5/6 users should note that the version number of the Enterprise Architect interface is stored in the VBP project file in a form similar to the following:</p> <pre>Reference=*\G{64FB2BF4-9EFA-11D2-8307-C45586000000}#2.2#0#... \Program Files\Sparx Systems\EA\EA.TLB#Enterprise Architect Object Model 2.02</pre> <p>If you experience problems moving from one version of Enterprise Architect to another, open the VBP file in a text editor and remove this line. Then open the project in Visual Basic and use <i>Project-References</i> to create a new reference to the Enterprise Architect Object model.</p>	
<b>Add-In Fails to Load</b>	<p>From Enterprise Architect release 7.0, Add-Ins created before 2004 are no longer supported. If an Add-In subscribes to the <code>Addn_Tmpl.tlb</code> interface (2003 style), it fails on load. In this event, contact the vendor or author of the Add-In and request an upgrade.</p>	
<b>Holding State Information</b>	<p>It is possible for an Add-In to hold state information, meaning that data can be stored in member variables in response to one event and retrieved in another. There are some dangers in doing this:</p> <ul style="list-style-type: none"> <li>Enterprise Architect Automation Objects do not update themselves in response to user activity, to activity on other workstations, or even to the actions of other objects in the same automation client; retaining handles to such objects between calls can result in the second event querying objects that have no relationship with the current state of Enterprise Architect</li> <li>When you close Enterprise Architect, all Add-Ins are</li> </ul>	



Topic	Detail	See also
	<p>asked to shut down; if there are any external automation clients Enterprise Architect must stay active, in which case all the Add-Ins are reloaded, losing all the data</p> <ul style="list-style-type: none"> <li>Enterprise Architect acting as an automation client does not close if an Add-In still holds a reference to it (releasing all references in the <code>Disconnect()</code> event avoids this problem)</li> </ul> <p>It is recommended that unless there is a specific reason for doing so, the Add-In should use the repository parameter and its method and properties to provide the necessary data.</p>	
<b>Enterprise Architect Not Closing</b>	<p><b>.NET Specific Issues</b></p> <p>Automation checks the use of objects and won't enable any of them to be destroyed until they are no longer being used.</p> <p>As noted in the <i>Automation Interface</i> topic, if your automation controller was written using the .NET framework, Enterprise Architect does not close even after you release all your references to it. To force the release of the COM pointers, call the memory management functions as shown below:</p> <pre>GC. Collect(); GC. WaitForPendingFinalizers();</pre> <p>Additionally, because automation clients hook into Enterprise Architect, which creates Add-Ins which in turn hook back into Enterprise Architect, it is possible to get into a deadlock situation where Enterprise Architect and the Add-Ins won't let go of one another and keep each other active. An Add-In might retain hooks into Enterprise Architect because:</p> <ul style="list-style-type: none"> <li>It keeps a private reference to an Enterprise Architect object (see <b>Holding State Information</b> above), or</li> <li>It has been created by .NET and the GC mechanism hasn't got around to releasing it</li> </ul> <p>There are two actions required to avoid deadlock situations:</p> <ul style="list-style-type: none"> <li>Automation controllers must call <code>Repository.CloseAddIns()</code> at some point (presumably at the end of processing)</li> <li>Add-Ins must release all references to Enterprise Architect in the <code>Disconnect()</code> event; see the <i>Add-In Events</i> topic for details</li> </ul> <p>It is possible that your Automation client controls a running instance of Enterprise Architect where the Add-Ins have not complied with the rule above. In this case you could call <code>Repository.Exit()</code> to terminate Enterprise Architect.</p> <p><b>Miscellaneous</b></p>	<p><a href="#">Automation Interface</a> <sup>[2809]</sup></p> <p><a href="#">Holding State Information</a> <sup>[3018]</sup></p> <p><a href="#">Add-In Events</a> <sup>[3021]</sup></p>

Topic	Detail	See also
	<p>In developing Add-Ins using the .Net framework you must select COM Interoperability in the project's properties in order for it to be recognized as an Add-In.</p> <p>Some development environments do not automatically register COM DLLs on creation. You might have to do that manually before Enterprise Architect recognizes the Add-In.</p> <p>You can use your private Add-In key (as required for Add-In deployment) to store configuration information pertinent to your Add-In.</p>	
<b>Concurrent Calls</b>	<p>In Enterprise Architect releases up to release 7.0, there is a possibility that Enterprise Architect could call two Add-In methods concurrently if the Add-In calls:</p> <ul style="list-style-type: none"> <li>• A message box</li> <li>• A modal dialog</li> <li>• VB DoEvents, .NET Application DoEvents or the equivalent in other languages</li> </ul> <p>In such cases, Enterprise Architect could initiate a second Add-In method before the first returns (re-entrancy). In release 7.0. and subsequent releases, Enterprise Architect cannot make such concurrent calls.</p> <p>If developing Add-Ins, ensure that the Add-In users are running Enterprise Architect release 7.0 or a later release to avoid any risk of concurrent method calls.</p>	

### 20.3.2 The Add-In Manager

If you want to check what Add-Ins are available on your system, and enable or disable them for use, you can review the Add-In Manager dialog. This dialog lists the Add-Ins that have been registered on your system, and their current status (**Enabled** or **Disabled**).

Access **Extensions | Manage Add-Ins**

#### Enable/disable Add-Ins

Action	Detail
<b>Enable an Add-In</b>	<p>To enable an Add-In so that it is available for use, select the <b>Load on Startup</b> check box to the left of the name.</p> <p>Click on the <b>OK</b> button.</p> <ul style="list-style-type: none"> <li>• Any Add-In specific features, facilities and Help are made available through the <b>Extensions   &lt;add-in name&gt;</b> menu option</li> <li>• Any defined Add-In windows are populated with information; select <b>Extensions   Add-In Windows</b></li> </ul>

Action	Detail
<b>Disable an Add-In</b>	<p>To disable an Add-In so that it is not available for use, clear the <b>Load on Startup</b> check box against the name.</p> <p>Click on the <b>OK</b> button.</p> <p>All menu options, features and facilities specific to the Add-In are hidden and made inactive.</p>

**Notes**

- When you enable or disable an Add-In, you must re-start Enterprise Architect to action the change

**Learn more**

- [MDG Technologies](#) <sup>[1475]</sup>
- [Enterprise Architect Add-In Model](#) <sup>[3010]</sup>
- [Register Add-In](#) <sup>[3171]</sup>

**20.3.3 Add-In Search****Topics**

Topic	Detail	See also								
<b>General Usage</b>	Enterprise Architect enables Extensions to integrate with the Model Search. Searches can be defined that execute a method within your Add-In and display your results in an integrated way.  The method that runs the search must be structured in the following way:  <pre>variant &lt;method name&gt; (Rep as Repository, SearchText as String, XMLResults as String)</pre>	<a href="#">Model Search</a> <sup>[700]</sup>  <a href="#">Search Data Format</a> <sup>[3020]</sup>								
	<table><tr><th>Parameter</th><th>Description</th></tr><tr><td><b>Rep</b></td><td>The currently open repository.</td></tr><tr><td><b>SearchText</b></td><td>An optional field that you can fill in through the M</td></tr><tr><td><b>XMLResults</b></td><td>At completion of the method, this should contain the search. The results should be an XML string the <b>Search Data Format</b>.</td></tr></table>	Parameter	Description	<b>Rep</b>	The currently open repository.	<b>SearchText</b>	An optional field that you can fill in through the M	<b>XMLResults</b>	At completion of the method, this should contain the search. The results should be an XML string the <b>Search Data Format</b> .	
Parameter	Description									
<b>Rep</b>	The currently open repository.									
<b>SearchText</b>	An optional field that you can fill in through the M									
<b>XMLResults</b>	At completion of the method, this should contain the search. The results should be an XML string the <b>Search Data Format</b> .									

Topic	Detail	See also
<b>Return</b>	The method must return a value for the results to be displayed.	
<b>Advanced Usage</b>	<p>In addition to the displayed results, two additional hidden fields can be passed into the XML that provide special functionality.</p> <p><b>CLASSTYPE</b></p> <p>Returning a field of CLASSTYPE, containing the <i>Object_Type</i> value from the <i>t_object</i> table, displays the appropriate icon in the column you place the field.</p> <p><b>CLASSGUID</b></p> <p>Returning a field of CLASSGUID, containing an <i>ea_guid</i> value, enables the Model Search to track the object in the Project Browser and open the Properties window for the element by double-clicking in the Model Search.</p>	

### 20.3.3.1 XML Format (Search Data)

#### Topics



Topic	Detail	See also
<b>General Usage</b>	The XML below provides the format for the <i>sSearchData</i> parameter of the <i>RunModelSearch</i> method.	<a href="#">Repository</a> [2869]
<b>Example Code</b>	<pre> &lt;Report ViewData UID=\ " My SearchID\ " &gt;   &lt;!--     // The UID attribute enables XML type     searches to persist column information. That     is, if you run the search, group by column or     adjust column widths, then close the window     and run the search again, the format/     organization changes are retained. To avoid     persisting column arrangements, leave the     attribute value blank or remove it altogether.     // Use this section to declare all possible     fields - columns that appear in Enterprise     Architect's search window - that are used     below in &lt;Rows/&gt;.     // The order of the columns of information     to be appended here must match the order that     the search run in Enterprise Architect would     normally display.     // Furthermore, if you append results onto     a custom SQL Search, then the order used in     your Custom SQL must match the order used     below.     --&gt;      &lt;Fields&gt;       &lt;Field name="" /&gt;       &lt;Field name="" /&gt;       &lt;Field name="" /&gt; </pre>	

Topic	Detail	See also
	<pre>         &lt;Field name="" /&gt;       &lt;/Fields&gt;        &lt;Rows&gt;         &lt;Row&gt;           &lt;Field name="" value="" /&gt;           &lt;Field name="" value="" /&gt;           &lt;Field name="" value="" /&gt;           &lt;Field name="" value="" /&gt;         &lt;/Row&gt;         &lt;Row&gt;           &lt;Field name="" value="" /&gt;           &lt;Field name="" value="" /&gt;           &lt;Field name="" value="" /&gt;           &lt;Field name="" value="" /&gt;         &lt;/Row&gt;         &lt;Row&gt;           &lt;Field name="" value="" /&gt;           &lt;Field name="" value="" /&gt;           &lt;Field name="" value="" /&gt;           &lt;Field name="" value="" /&gt;         &lt;/Row&gt;       &lt;/Rows&gt;     &lt;/ReportViewData&gt; </pre>	

### 20.3.4 Add-In Events



All Enterprise Architect Add-Ins can choose to respond to the following general Add-In events:

Topic	Link
<i>EA_Connect</i> - enables Add-Ins to identify their type and to respond to Enterprise Architect start up.	<a href="#">EA_Connect</a> <sup>[3022]</sup>
<i>EA_Disconnect</i> - enables the Add-In to respond to user requests to disconnect the model branch from an external project.	<a href="#">EA_Disconnect</a> <sup>[3023]</sup>
<i>EA_GetMenuItems</i> - enables the Add-In to provide the Enterprise Architect user interface with additional Add-In menu options in various context and main menus.	<a href="#">EA_GetMenuItems</a> <sup>[3023]</sup>
<i>EA_MenuClick</i> - received by an Add-In in response to user selection of a menu option.	<a href="#">EA_MenuClick</a> <sup>[3025]</sup>
<i>EA_GetMenuState</i> - enables the Add-In to set a particular menu option to either enabled or disabled.	<a href="#">EA_GetMenuState</a> <sup>[3024]</sup>
<i>EA_ShowHelp</i> - enables the Add-In to show a help topic for a particular menu option.	<a href="#">EA_ShowHelp</a> <sup>[3029]</sup>

Topic	Link
<i>EA_OnOutputItemClicked</i> - informs Add-Ins that the user has clicked on a list entry in the system tab or one of the user defined output tabs.	<a href="#">EA_OnOutputItemClicked</a> 
<i>EA_OnOutputItemDoubleClicked</i> - informs Add-Ins that the user has used the mouse to double-click on a list entry in one of the user-defined output tabs.	<a href="#">EA_OnOutputItemDoubleClicked</a> 

### 20.3.4.1 EA\_Connect


#### Topics

Topic	Detail	See also
<b>Details</b>	<p><i>EA_Connect</i> events enable Add-Ins to identify their type and to respond to Enterprise Architect start up.</p> <p>This event occurs when Enterprise Architect first loads your Add-In. Enterprise Architect itself is loading at this time so that while a Repository object is supplied, there is limited information that you can extract from it.</p> <p>The chief uses for <i>EA_Connect</i> are in initializing global Add-In data and for identifying the Add-In as an MDG Add-In.</p>	<a href="#">MDG Add-In</a>  <a href="#">EA_Disconnect</a> 

#### Syntax:


**Function** *EA\_Connect*(*Repository As EA.Repository*) *As String*

The *EA\_Connect* function syntax has the following elements:

Parameter	Type	Direction	Description
<b>Repository</b>	<a href="#">EA.Repository</a> 	IN	An EA.Repository object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.

#### Return Value

A string identifying a specialized type of Add-In:

Type	Details
"MDG"	MDG Add-Ins receive <a href="#">MDG Events</a>  and extra menu options.

Type	Details
""	A non-specialized Add-In.

### 20.3.4.2 EA\_Disconnect

#### Topics

Topic	Detail	See also
<b>Details</b>	<p>The <i>EA_Disconnect</i> event enables the Add-In to respond to user requests to disconnect the model branch from an external project.</p> <p>This function is called when the Enterprise Architect closes. If you have stored references to Enterprise Architect objects (not particularly recommended anyway), you must release them here.</p> <p>In addition, .NET users must call memory management functions as shown below:</p> <pre>GC. Collect ( ) ; GC. Wait For PendingFinalizers ( ) ;</pre>	<a href="#">EA_Connect</a> <sup>[3022]</sup>

#### Syntax:

*Sub* EA\_Disconnect()

#### Return Value:

None.

### 20.3.4.3 EA\_GetMenuItems

#### Topics


Topic	Detail	See also
<b>Details</b>	<p>The <i>EA_GetMenuItems</i> event enables the Add-In to provide the Enterprise Architect user interface with additional Add-In menu options in various context and main menus. When a user selects an Add-In menu option, an event is raised and passed back to the Add-In that originally defined that menu option.</p> <p>This event is raised just before Enterprise Architect has to show particular menu options to the user, and its use is described in the <i>Define Menu Items</i> topic.</p>	<a href="#">Define Menu Items</a> <sup>[3013]</sup> <a href="#">EA_MenuClick</a> <sup>[3025]</sup> <a href="#">EA_GetMenuState</a> <sup>[3024]</sup>

Topic	Detail	See also

**Syntax:**

**Function** *EA\_GetMenuItems*(*Repository As EA.Repository, MenuLocation As String, MenuName As String*) *As Variant*

The *EA\_GetMenuItems* function syntax has the following elements:

Parameter	Type	Direction	Description
<b>MenuLocation</b>	<i>String</i>		A string representing the part of the user interface that brought up the menu. This can be <i>TreeView</i> , <i>MainMenu</i> or <i>Diagram</i> .
<b>MenuName</b>	<i>String</i>		The name of the parent menu for which sub-items are to be defined. In the case of the top-level menu this is an empty string.
<b>Repository</b>	<a href="#">EA.Repository</a> 	IN	An EA.Repository object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.

**Return Value:**



One of the following types:

- A string indicating the label for a single menu option
- An array of strings indicating a multiple menu options
- Empty (Visual Basic/VB.NET) or null (C#) to indicate that no menu should be displayed

In the case of the top-level menu it should be a single string or an array containing only one item, or empty/null.

#### 20.3.4.4 *EA\_GetMenuState*

**Topics**

Topic	Detail	See also
<b>Details</b>	<p>The <i>EA_GetMenuState</i> event enables the Add-In to set a particular menu option to either enabled or disabled. This is useful when dealing with locked packages and other situations where it is convenient to show a menu option, but not enable it for use.</p> <p>This event is raised just before Enterprise Architect has to show particular menu options to the user. Its use is described in the <i>Define Menu Items</i> topic.</p>	<a href="#">Define Menu Items</a>  <a href="#">EA_GetMenuItems</a> 



Topic	Detail	See also

**Syntax:**

**Sub EA\_GetMenuState**(*Repository as EA.Repository, MenuLocation As String, MenuName as String, ItemName as String, IsEnabled as Boolean, IsChecked as Boolean*)

The *EA\_GetMenuState* function syntax has the following elements:


Parameter	Type	Direction	Description
<b>IsChecked</b>	<i>Boolean</i>		Set to <b>True</b> to check this particular menu option.
<b>IsEnabled</b>	<i>Boolean</i>		Set to <b>False</b> to disable this particular menu option.
<b>ItemName</b>	<i>String</i>		The name of the option actually clicked; for example, <i>Create a New Invoice</i> .
<b>MenuLocation</b>	<i>String</i>		A string representing the part of the user interface that brought up the menu. This can be <i>TreeView</i> , <i>MainMenu</i> or <i>Diagram</i> .
<b>MenuName</b>	<i>String</i>		The name of the parent menu for which sub-items must be defined.  In the case of the top-level menu it is an empty string.
<b>Repository</b>	<a href="#">EA.Repository</a> <small>[2850]</small>	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.

**Return Value:**

None.

**20.3.4.5 EA\_MenuClick****Topics**


Topic	Detail	See also
<b>Details</b>	<i>EA_MenuClick</i> events are received by an Add-In in response to user selection of a menu option.	<a href="#">Repository</a> <small>[2850]</small>  <a href="#">EA_GetMenuItems</a>

Topic	Detail	See also
	<p>The event is raised when the user clicks on a particular menu option. When a user clicks on one of your non-parent menu options, your Add-In receives a <i>MenuClick</i> event, defined as follows:</p> <pre>Sub EA_MenuClick( Repository As EA. Repository, ByVal MenuName As String, ByVal ItemName As String)</pre> <p>The code below illustrates an example of use:</p> <pre>    If MenuName = "- &amp;Diagram" And ItemName = "&amp;Properties" then         MsgBox Repository. Get Current Diagram Name, vbInformation     Else         MsgBox "Not Implemented", vbCritical     End If</pre> <p>Notice that your code can directly access Enterprise Architect data and UI elements using <b>Repository</b> methods.</p>	

**Syntax:**

***Sub EA\_MenuClick(Repository As EA.Repository, MenuLocation As String, MenuName As String, ItemName As String)***

The *EA\_GetMenuClick* function syntax has the following elements:

Parameter	Type	Direction	Description
<b>ItemName</b>	<i>String</i>		The name of the option actually clicked, for example, <i>Create a New Invoice</i> .
<b>MenuName</b>	<i>String</i>		The name of the parent menu for which sub-items are to be defined.  In the case of the top-level menu this is an empty string.
<b>Repository</b>	<a href="#">EA.Repository</a> 	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.

**Return Value:**

None.

### 20.3.4.6 EA\_OnOutputItemClicked

#### Topics

Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnOutputItemClicked</i> events inform Add-Ins that the user has clicked on a list entry in the system tab or one of the user defined output tabs.</p> <p>Usually an Add-In responds to this event in order to capture activity on an output tab they had previously created through a call to <i>Repository.AddTab()</i>.</p> <p>Note that every loaded Add-In receives this event for every click on an output tab in Enterprise Architect, irrespective of whether the Add-In created that tab. Add-Ins should therefore check the <i>TabName</i> parameter supplied by this event to ensure that they are not responding to other Add-Ins' events.</p>	<a href="#">EA_OnOutputItemDoubleClicked</a> <sup>[3028]</sup>

#### Syntax:

**EA\_OnOutputItemClicked**(*Repository As EA.Repository, TabName As String, LineText As String, ID As Long*)

The *EA\_OnOutputItemClicked* function syntax has the following elements:

Parameter	Type	Direction	Description
<b>ID</b>	<i>Long</i>	IN	The ID value specified in the original call to <i>Repository.WriteOutput()</i> .
<b>LineText</b>	<i>String</i>	IN	The text that had been supplied as the String parameter in the original call to <i>Repository.WriteOutput()</i> .
<b>Repository</b>	<a href="#">EA.Repository</a> <sup>[2850]</sup>	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.
<b>TabName</b>	<i>String</i>	IN	The name of the tab that the click occurred in. Usually this would have been created through <i>Repository.AddTab()</i> .

#### Return Value:

None.

### 20.3.4.7 EA\_OnOutputItemDoubleClicked

#### Topics

Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnOutputItemDoubleClicked</i> events informs Add-Ins that the user has used the mouse to double-click on a list entry in one of the user-defined output tabs.</p> <p>Usually an Add-In responds to this event in order to capture activity on an output tab they had previously created through a call to <i>Repository.AddTab()</i>.</p> <p>Note that every loaded Add-In receives this event for every double-click on an output tab in Enterprise Architect, irrespective of whether the Add-In created that tab; Add-Ins should therefore check the <i>TabName</i> parameter supplied by this event to ensure that they are not responding to other Add-Ins' events.</p>	<a href="#">EA_OnOutputItemClicked</a> [3027]

#### Syntax:

**EA\_OnOutputItemDoubleClicked**(*Repository As EA.Repository, TabName As String, LineText As String, ID As Long*)

The *EA\_OnOutputItemClicked* function syntax contains the following elements:


Parameter	Type	Direction	Description
<b>ID</b>	<i>Long</i>	IN	The ID value specified in the original call to <i>Repository.WriteOutput()</i> .
<b>LineText</b>	<i>String</i>	IN	The text that had been supplied as the String parameter in the original call to <i>Repository.WriteOutput()</i> .
<b>Repository</b>	<a href="#">EA.Repository</a> [2850]	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model; poll its members to retrieve model data and user interface status information.
<b>TabName</b>	<i>String</i>	IN	The name of the tab that the click occurred in; usually this would have been created through <i>Repository.AddTab()</i> .

#### Return Value:

None.

### 20.3.4.8 EA\_ShowHelp


#### Topics

Topic	Detail	See also
<b>Details</b>	<p>The <i>EA_ShowHelp</i> event enables the Add-In to show a help topic for a particular menu option. When the user has an Add-In menu option selected, pressing ( <b>F1</b> ) can be related to the required Help topic by the Add-In and a suitable help message shown.</p> <p>This event is raised when the user presses ( <b>F1</b> ) on a menu option that is not a parent menu.</p>	<a href="#">EA_GetMenuItems</a> 

#### Syntax:

**Sub** EA\_ShowHelp(*Repository* as EA.Repository, *MenuLocation* As String, *MenuName* as String, *ItemName* as String)

The *EA\_ShowHelp* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>ItemName</b>	<i>String</i>		The name of the option actually clicked, for example, <b>Create a New Invoice</b> .
<b>MenuLocation</b>	<i>String</i>		A string representing the part of the user interface that brought up the menu. This can be <i>Treeview</i> , <i>MainMenu</i> or <i>Diagram</i> .
<b>MenuName</b>	<i>String</i>		The name of the parent menu for which sub-items are to be defined. In the case of the top-level menu this is an empty string.
<b>Repository</b>	<a href="#">EA.Repository</a> 	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.

#### Return Value:

None.



### 20.3.5 Broadcast Events

#### Overview

The following general Broadcast events are sent to all loaded Add-Ins. For an Add-In to receive the event, they must first implement the required automation event interface. If Enterprise Architect detects that the Add-

If it has the required interface, the event is dispatched to the Add-In.

Topic	Link
Add-In Licence Management Events	<a href="#">Add-In Licence Management Events</a> 
Compartment Events	<a href="#">Compartment Events</a> 
Context Item Events	<a href="#">Context Item Events</a> 
File Close Event	<a href="#">File Close Event</a> 
File New Event	<a href="#">File New Event</a> 
File Open Event	<a href="#">File Open Event</a> 
Model Validation Broadcasts	<a href="#">Model Validation Broadcasts</a> 
On Tab Changed Event	<a href="#">EA_OnTabChanged</a> 
Post Close Diagram Event	<a href="#">Post Close Diagram Event</a> 
Post Initialization Event	<a href="#">EA_OnPostInitialized</a> 
Post New Events	<a href="#">Post-New Events</a> 
Post Open Diagram Event	<a href="#">Post Open Diagram Event</a> 
Pre-Deletion Events	<a href="#">Pre-Deletion Events</a> 
Pre-Exit Instance (not currently used)	<a href="#">PreExit Instance</a> 
On the creation of new objects	<a href="#">Pre New-Object Events</a> 
Retrieve Model Template Event	<a href="#">Retrieve Model Template Event</a> 
Tagged Value Broadcasts	<a href="#">Tagged Value Broadcasts</a> 

Topic	Link
Technology Events	<a href="#">Technology Events</a> 
Transformation Event	<a href="#">Transformation Event</a> 




**MDG Events** add quite a number of additional events, but the Add-In must first have registered as an MDG-style Add-In, rather than as a generic Add-In.

#### Learn more

- [MDG Events](#) 

### 20.3.5.1 Add-In License Management Events

Enterprise Architect Add-Ins can respond to the following events associated with Add-In License Management:

Event	Link
EA_AddinLicenseValidate	<a href="#">EA_AddinLicenseValidate</a> 
EA_AddinLicenseGetDescription	<a href="#">EA_AddinLicenseGetDescription</a> 
EA_GetSharedAddinName	<a href="#">EA_GetSharedAddinName</a> 

#### Learn more

- [Register Add-In](#) 

#### 20.3.5.1.1 EA\_AddinLicenseValidate

##### Description

When a user directly enters a license key that doesn't match a Sparx Systems key into the License Management dialog **EA\_AddinLicenseValidate** is broadcast to all Enterprise Architect Add-Ins, providing them with a chance to use the Add-In key to determine the level of functionality to provide. When a key is retrieved from the Sparx Systems Keystore only the target Add-In will be called with the key.

For the Add-In to validate itself against this key, the Add-In's **EA\_AddinLicenseValidate** handler should return **true** to confirm that the license has been validated. As the **EA\_AddinLicenseValidate** event is broadcast to all Add-Ins, one license can validate many Add-Ins.

If an Add-In elects to handle a license key by returning **true** to **EA\_AddinLicenseValidate**, it is called upon to provide a description of the license key through the **EA\_AddinLicenseGetDescription** event. If more than one Add-In elects to handle a license key, the first Add-In that returns **true** to **EA\_AddinLicenseValidate** is

queried for the license key description.

### Syntax

**Function** *EA\_AddInLicenseValidate (Repository As EA.Repository, AddinKey As String) As Boolean*

Parameter	Type	Direction	Description
<b>AddinKey</b>	<i>String</i>	IN	The Add-in license key that has been entered in the License Management dialog.
<b>Repository</b>	<a href="#">EA.Repository</a> <small>[2850]</small>	IN	An <b>EA.Repository</b> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.

### Return Value

For the Add-in to validate against this key it should return **true** to indicate that the key is valid and has been handled.

### Learn more

- [EA\\_AddInLicenseGetDescription](#)  
[3032]
- [EA\\_GetSharedAddinName](#)  
[3033]

#### 20.3.5.1.2 EA\_AddInLicenseGetDescription

### Description

Before the Enterprise Architect License Management dialog is displayed, *EA\_AddInLicenseGetDescription* is sent once for each Add-In key to the first Add-In that elected to handle that key.

The value returned by *EA\_AddInLicenseGetDescription* is used as the key's plain text description.

### Syntax

**Function** *EA\_AddInLicenseGetDescription (Repository as EA.Repository, AddinKey as String) As String*

Parameter	Type	Direction	Description
<b>AddinKey</b>	<i>String</i>	IN	The Add-In license key that Enterprise Architect requires a description for.
<b>Repository</b>	<a href="#">EA.Repository</a> <small>[2850]</small>	IN	An EA.Repository object representing the currently open model. Poll its members to retrieve model data and user interface status information.



Parameter	Type	Direction	Description

### Return Value

A String containing a plain text description of the provided AddinKey.

### Learn more

- [EA\\_AddinLicenseValidate](#)  <sup>[3031]</sup>
- [EA\\_GetSharedAddinName](#)  <sup>[3033]</sup>


#### 20.3.5.1.3 *EA\_GetSharedAddinName*

### Description

As an add-in writer you can distribute keys to your add-in via the Enterprise Architect Keystore providing your keys are generated using a prefix that allows Enterprise Architect to identify the add-in to which they belong. *EA\_GetSharedAddinName* is called by Enterprise Architect to determine what prefix an add-in is using. If a matching key is found in the keystore the License Management dialog will display the name returned by *EA\_AddinLicenseGetDescription* to your users. Finally, when the user selects a key, that key will be passed to your add-in to validate by calling *EA\_AddinLicenseValidate*.

### Syntax

**Function** *EA\_GetSharedAddinName (Repository as EA.Repository) As String*

Parameter	Type	Direction	Description
<b>Repository</b>	<a href="#">EA.Repository</a>  <sup>[2850]</sup>	IN	An EA.Repository object representing the currently open model. Poll its members to retrieve model data and user interface status information.

### Return Value

A String containing a product name code for the provided Add-In. This will be shown in plain text at the start of any keys added to the keystore. We recommend contacting Sparx Systems directly with proposed values to ensure you don't clash with any other add-ins.

eg. The following keys would all be interpreted as belonging to an add-in returning "MYADDIN" from this function:

- MYADDIN-Test
- MYADDIN-{7AC4D426-9083-4fa2-93B7-25E2B7FB8DC5}
- MYADDIN-7AC4D426-9083-4fa2-93B7
- MYADDIN-25E2B7FB8DC5
- MYADDIN-2hDfHKA5jf0GAjn92UvqAnxwC13dxQGJtH7zLHJ9Ym8=

Learn more

- [EA\\_AddinLicenseValidate](#) <sup>[3031]</sup>
- [EA\\_AddinLicenseGetDescription](#) <sup>[3032]</sup>

**20.3.5.2 Compartment Events**

Enterprise Architect Add-Ins can respond to the following events associated with user-generated element compartments:

Topic	Link
EA_QueryAvailableCompartments	<a href="#">EA_QueryAvailableCompartments</a> <sup>[3034]</sup>
EA_GetCompartmentData	<a href="#">EA_GetCompartmentData</a> <sup>[3035]</sup>

**20.3.5.2.1 EA\_QueryAvailableCompartments**Topics

Topic	Detail	See also
<b>Details</b>	<p>This event occurs when Enterprise Architect's diagrams are refreshed. It is a request for the Add-In to provide a list of user-defined compartments.</p> <p>The <i>EA_GetCompartmentData</i> event then queries each object for the data to display in each user-defined compartment.</p>	<a href="#">EA_GetCompartmentData</a> <sup>[3035]</sup>

Syntax:

**Function** *EA\_QueryAvailableCompartments(Repository As EA.Repository) As Variant*

The *EA\_QueryAvailableCompartments* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Repository</b>	<a href="#">EA.Repository</a> <sup>[2850]</sup>	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.

Return Value:

A *String* containing a comma-separated list of user-defined compartments.

**Example**

```

Function EA_QueryAvailableCompartments(Repository As EA.Repository) As Variant
    Dim sReturn As String
    sReturn = ""
    If m_FirstCompartmentVisible = True Then
        sReturn = sReturn + "first,"
    End If
    If m_SecondCompartmentVisible = True Then
        sReturn = sReturn + "second,"
    End If
    If m_ThirdCompartmentVisible = True Then
        sReturn = sReturn + "third,"
    End If

    If Len(sReturn) > 0 Then
        sReturn = Left(sReturn, Len(sReturn) - 1)
    End If

    EA_QueryAvailableCompartments = sReturn
End Function

```

**20.3.5.2.2 EA\_GetCompartmentData****Topics**

Topic	Detail	See also
<b>Details</b>	This event occurs when Enterprise Architect is instructed to redraw an element. It requests that the Add-In provide the data to populate the element's compartment.	

**Syntax:**

**Function** EA\_GetCompartmentData(*Repository As EA.Repository, sCompartment As String, sGUID As String, oType As EA.ObjectType*) **As Variant**

The *EA\_QueryAvailableCompartments* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>oType</b>	<i>ObjectType</i>	IN	The type of the element for which data is being requested.
<b>Repository</b>	<a href="#">EA.Repository</a> <small>[2850]</small>	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.
<b>sCompartment</b>	<i>String</i>	IN	The name of the compartment for which data is being requested.
<b>sGUID</b>	<i>String</i>	IN	The GUID of the element for which data is being requested.

Parameter	Type	Direction	Description

**Return Value:**

A variant containing a formatted string. See the example below to view the format.

**Example**

```

Function EA_GetCompartmentData(Repository As EA.Repository, sCompartment
As String, sGUID As String, oType As EA.ObjectType) As Variant

    If Repository Is Nothing Then
        Exit Function
    End If

    Dim sCompartmentData As String
    Dim oXML As MSXML2.DOMDocument
    Dim Nodes As MSXML2.IXMLDOMNodeList
    Dim Node1 As MSXML2.IXMLDOMNode
    Dim Node As MSXML2.IXMLDOMNode
    Dim sData As String

    sCompartmentData = ""
    Set oXML = New MSXML2.DOMDocument
    sData = ""

    On Error GoTo ERR_GetCompartmentData

    oXML.LoadXML (Repository.GetTreeXMLByGUID(sGUID))
    Set Node1 = oXML.selectSingleNode("//ModelItem")

    If Node1 Is Nothing Then
        Exit Function
    End If

    sCompartmentData = sCompartmentData + "Name=" + sCompartment + ";"
    sCompartmentData = sCompartmentData + "OwnerGUID=" + sGUID + ";"
    sCompartmentData = sCompartmentData +
    "Options=SkiplfOnDiagram&_eq_^1&_sc_^"

    Select Case sCompartment
    Case "parts"
        Set Nodes = Node1.selectNodes("ModelItem( @Metatype=""Part"" ) ")
        For Each Node In Nodes
            sData = sData + "Data&_eq_^" + Node.Attributes.getNamedItem
("Name").nodeValue + "&_sc_^"
            sData = sData + "GUID&_eq_^" + Node.Attributes.getNamedItem
("GUID").nodeValue + "&_sc_^,"
        Next

    Case "ports"
        Set Nodes = Node1.selectNodes("ModelItem( @Metatype=""Port"" ) ")
        For Each Node In Nodes
            sData = sData + "Data&_eq_^" + Node.Attributes.getNamedItem
("Name").nodeValue + "&_sc_^"
            sData = sData + "GUID&_eq_^" + Node.Attributes.getNamedItem
("GUID").nodeValue + "&_sc_^,"
        Next

    End Select

    ' If there's no data to display, then don't return any compartment

```

```

data
  If sData <> "" Then
    sCompartmentData = sCompartmentData + "CompartmentData=" + sData +
    ","
  Else
    sCompartmentData = ""
  End If

  EA_GetCompartmentData = sCompartmentData
  Exit Function




ERR_GetCompartmentData:
  EA_GetCompartmentData = ""

End Function

```




### 20.3.5.3 Context Item Events

Enterprise Architect Add-Ins can respond to the following events associated with changing context:

Topic	Link
EA_OnContextItemChanged	<a href="#">EA_OnContextItemChanged</a> 
EA_OnContextItemDoubleClicked	<a href="#">EA_OnContextItemDoubleClicked</a> 
EA_OnNotifyContextItemModified	<a href="#">EA_OnNotifyContextItemModified</a> 

#### 20.3.5.3.1 EA\_OnContextItemChanged

##### Topics

Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnContextItemChanged</i> notifies Add-Ins that a different item is now in context.</p> <p>This event occurs after a user has selected an item anywhere in the Enterprise Architect GUI. Add-Ins that require knowledge of the current item in context can subscribe to this broadcast function. If <b>ot = otRepository</b>, then this function behaves the same as <i>EA_FileOpen</i>.</p>	<p><a href="#">EA_FileOpen</a> </p> <p><a href="#">EA_OnContextItemDoubleClicked</a> </p> <p><a href="#">EA_OnNotifyContextItemModified</a> </p>

##### Syntax:

**Sub** EA\_OnContextItemChanged(*Repository As EA.Repository, GUID As String, ot as EA.ObjectType*)

The *EA\_OnContextItemChanged* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>GUID</b>	<i>String</i>	IN	<p>Contains the GUID of the new context item. This value corresponds to the following properties, depending on the value of the <b>ot</b> parameter:</p> <p><i>ot (ObjectType)</i> - GUID value</p> <p><i>otElement</i> - <i>Element.ElementGUID</i></p> <p><i>otPackage</i> - <i>Package.PackageGUID</i></p> <p><i>otDiagram</i> - <i>Diagram.DiagramGUID</i></p> <p><i>otAttribute</i> - <i>Attribute.AttributeGUID</i></p> <p><i>otMethod</i> - <i>Method.MethodGUID</i></p> <p><i>otConnector</i> - <i>Connector.ConnectorGUID</i></p> <p><i>otRepository</i> - NOT APPLICABLE, GUID is an empty string</p>
<b>ot</b>	<i>EA.ObjectType</i>	IN	Specifies the type of the new context item.
<b>Repository</b>	<a href="#">EA.Repository</a> [2850]	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.

**Return Value:**

None.

**20.3.5.3.2 EA\_OnContextItemDoubleClicked****Topics**

Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnContextItemDoubleClicked</i> notifies Add-Ins that the user has double-clicked the item currently in context.</p> <p>This event occurs when a user has double-clicked (or pressed ( <b>Enter</b> ) ) on the item in context, either in a diagram or in the Project Browser. Add-Ins to handle events can subscribe to this broadcast function.</p>	<p><a href="#">EA_OnContextItemChanged</a> [3037]</p> <p><a href="#">EA_OnNotifyContextItemModified</a> [3039]</p>

**Syntax:**

**Function** *EA\_OnContextItemDoubleClicked*(*Repository As EA.Repository, GUID As String, ot as EA.ObjectType*)

The *EA\_OnContextItemDoubleClicked* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>GUID</b>	<i>String</i>	IN	Contains the GUID of the new context item. This value corresponds to the following properties, depending on the value of the <b>ot</b> parameter:  <i>ot (ObjectType)</i> - <i>GUID value</i> <i>otElement</i> - <i>Element.ElementGUID</i> <i>otPackage</i> - <i>Package.PackageGUID</i> <i>otDiagram</i> - <i>Diagram.DiagramGUID</i> <i>otAttribute</i> - <i>Attribute.AttributeGUID</i> <i>otMethod</i> - <i>Method.MethodGUID</i> <i>otConnector</i> - <i>Connector.ConnectorGUID</i>
<b>ot</b>	<i>EA.ObjectType</i>	IN	Specifies the type of the new context item.
<b>Repository</b>	<a href="#">EA.Repository</a> [2850]	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.

**Return Value:**

Return **True** to notify Enterprise Architect that the double-click event has been handled by an Add-In.

Return **False** to enable Enterprise Architect to continue processing the event.

**20.3.5.3.3 EA\_OnNotifyContextItemModified****Topics**

Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnNotifyContextItemModified</i> notifies Add-Ins that the current context item has been modified.</p> <p>This event occurs when a user has modified the context item. Add-Ins that require knowledge of when an item has been modified can subscribe to this broadcast function.</p>	<p><a href="#">EA_OnContextItemChanged</a> [3037]</p> <p><a href="#">EA_OnContextItemDoubleClicked</a> [3038]</p>

**Syntax:**

**Sub EA\_OnNotifyContextItemModified(Repository As EA.Repository, GUID As String, ot as EA.ObjectType)**

The *EA\_OnNotifyContextItemModified* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>GUID</b>	<i>String</i>	IN	<p>Contains the GUID of the new context item. This value corresponds to the following properties, depending on the value of the <b>ot</b> parameter:</p> <ul style="list-style-type: none"> <li>• <i>ot(ObjectType)</i> - <i>GUID value</i></li> <li>• <i>otElement</i> - <i>Element.ElementGUID</i></li> <li>• <i>otPackage</i> - <i>Package.PackageGUID</i></li> <li>• <i>otDiagram</i> - <i>Diagram.DiagramGUID</i></li> <li>• <i>otAttribute</i> - <i>Attribute.AttributeGUID</i></li> <li>• <i>otMethod</i> - <i>Method.MethodGUID</i></li> <li>• <i>otConnector</i> - <i>Connector.ConnectorGUID</i></li> </ul>
<b>ot</b>	<i>EA.ObjectType</i>	IN	Specifies the type of the new context item.
<b>Repository</b>	<a href="#">EA.Repository</a> [2850]	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.

**Return Value:**

None.

**20.3.5.4 EA\_FileClose****Topics**

Topic	Detail	See also
<b>Details</b>	<p>The <i>EA_FileClose</i> event enables the Add-In to respond to a File Close event. When Enterprise Architect closes an opened Model file, this event is raised and passed to all Add-Ins implementing this method.</p> <p>This event occurs when the model currently opened within Enterprise Architect is about to be closed (when another model is about to be opened or when Enterprise Architect is about to shutdown)..</p>	<a href="#">EA_FileOpen</a> [3042] <a href="#">EA_FileNew</a> [3041]

**Syntax:*****Sub EA\_FileClose(Repository As EA.Repository)***The *EA\_FileClose* function syntax contains the following elements:



Parameter	Type	Direction	Description
<b>Repository</b>	<a href="#">EA.Repository</a> <small>[2850]</small>	IN	An <i>EA.Repository</i> object representing the Enterprise Architect model about to be closed.  Poll its members to retrieve model data and user interface status information.

**Return Value:**

None.

### 20.3.5.5 *EA\_FileNew*

**Topics**

Topic	Detail	See also
<b>Details</b>	<p>The <i>EA_FileNew</i> event enables the Add-In to respond to a <i>File New</i> event. When Enterprise Architect creates a new model file, this event is raised and passed to all Add-Ins implementing this method.</p> <p>The event occurs when the model being viewed by the Enterprise Architect user changes, for whatever reason (through user interaction or Add-In activity).</p>	<a href="#">EA_FileClose</a> <small>[3040]</small> <a href="#">EA_FileOpen</a> <small>[3042]</small>

**Syntax:*****Sub EA\_FileNew (Repository As EA.Repository)***

The *EA\_FileNew* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Repository</b>	<a href="#">EA.Repository</a> <small>[2850]</small>	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.  Poll its members to retrieve model data and user interface status information.

**Return Value:**

None.

### 20.3.5.6 *EA\_FileOpen*

#### Topics

Topic	Detail	See also
<b>Details</b>	<p>The <i>EA_FileOpen</i> event enables the Add-In to respond to a <i>File Open</i> event. When Enterprise Architect opens a new model file, this event is raised and passed to all Add-Ins implementing this method.</p> <p>The event occurs when the model being viewed by the Enterprise Architect user changes, for whatever reason (through user interaction or Add-In activity).</p>	<a href="#">EA_FileClose</a> <sup>[3040]</sup> <a href="#">EA_FileNew</a> <sup>[3041]</sup>

#### Syntax:

**Sub** *EA\_FileOpen*(*Repository As EA.Repository*)

The *EA\_FileOpen* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Repository</b>	<a href="#">EA.Repository</a> <sup>[2850]</sup>	IN	<p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>

#### Return Value:

None.

### 20.3.5.7 *EA\_OnPostCloseDiagram*

#### Topics

Topic	Detail	See also
<b>Details</b>	<i>EA_OnPostCloseDiagram</i> notifies Add-Ins that a diagram has been closed.	<a href="#">EA_OnPostOpenDiagram</a> <sup>[3044]</sup>

#### Syntax:

**Function** *EA\_OnPostCloseDiagram*(*Repository As EA.Repository, DiagramID As Integer*)

The *EA\_OnPostCloseDiagram* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>DiagramID</b>	<i>Integer</i>	IN	Contains the Diagram ID of the diagram that was closed.
<b>Repository</b>	<a href="#">EA.Repository</a> <small>[2850]</small>	IN	An <i>EA.Repository</i> object representing the Enterprise Architect model about to be closed.  Poll its members to retrieve model data and user interface status information.

**Return Value:**

None.

**20.3.5.8 EA\_OnPostInitialized****Topics**

Topic	Detail	See also
<b>Details</b>	<i>EA_OnPostInitialized</i> notifies Add-Ins that the Repository object has finished loading and any necessary initialization steps can now be performed on the object.  For example, the Add-In can create an Output tab using <i>Repository.CreateOutputTab</i> .	<a href="#">Repository.CreateOutputTab</a> <small>[2857]</small>

**Syntax:*****Sub EA\_OnPostInitialized(Repository As EA.Repository)***The *EA\_OnPostInitialized* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Repository</b>	<a href="#">EA.Repository</a> <small>[2850]</small>	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.  Poll its members to retrieve model data and user interface status information.

**Return Value:**

None.

### 20.3.5.9 EA\_OnPostOpenDiagram

#### Topics

Topic	Detail	See also
<b>Details</b>	<i>EA_OnPostOpenDiagram</i> notifies Add-Ins that a diagram has been opened.	<a href="#">EA_OnPostCloseDiagram</a> <small>[3042]</small>

#### Syntax:

**Function** *EA\_OnPostOpenDiagram*(*Repository As EA.Repository, DiagramID As Integer*)

The *EA\_OnPostOpenDiagram* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>DiagramID</b>	<i>Integer</i>	IN	Contains the Diagram ID of the diagram that was opened.
<b>Repository</b>	<a href="#">EA.Repository</a> <small>[2850]</small>	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.  Poll its members to retrieve model data and user interface status information.

#### Return Value:

None.

### 20.3.5.10 EA\_OnPostTransform

#### Topics

Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnPostTransform</i> notifies Add-Ins that an MDG transformation has taken place with the output in the specified target package.</p> <p>This event occurs when a user runs an MDG transform on one or more target packages; the notification is provided for each transform/target package immediately after all transform processes have completed.</p>	

#### Syntax:

**Function** *EA\_OnPostTransform(Repository As EA.Repository, Info As EA.EventProperties) As Boolean*

The *EA\_OnPostTransform* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Info</b>	<a href="#">EA.EventProperties</a> <small>[2833]</small>	IN	Contains the following <i>EventProperty Objects</i> for the transform performed: <ul style="list-style-type: none"> <li><i>Transform</i>: A string value corresponding to the name of the transform used</li> <li><i>PackageID</i>: A long value corresponding to <i>Package.PackageID</i> of the destination package</li> </ul>
<b>Repository</b>	<a href="#">EA.Repository</a> <small>[2850]</small>	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.  Poll its members to retrieve model data and user interface status information.

**Return Value:**

Reserved for future use.

### 20.3.5.11 *EA\_OnPreExitInstance*

**Details:**

*EA\_OnPreExitInstance* is not currently used.

**Syntax:**

**Sub** *EA\_OnPreExitInstance(Repository As EA.Repository)*

The *EA\_OnPreExitInstance* function syntax contains the following element:

Parameter	Type	Direction	Description
<b>Repository</b>	<a href="#">EA.Repository</a> <small>[2850]</small>	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.  Poll its members to retrieve model data and user interface status information.

**Return Value:**

None.

### 20.3.5.12 EA\_OnRetrieveModelTemplate

#### Description

**EA\_OnRetrieveModelTemplate** requests that an Add-In pass a model template to Enterprise Architect. This event occurs when a user executes the **Add a New Model Using Wizard** command to add a model that has been defined by an MDG Technology.

#### Syntax

**Function** EA\_OnRetrieveModelTemplate(*Repository As EA.Repository, sLocation As String*) *As String*

The **EA\_OnRetrieveModelTemplate** function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Repository</b>	<a href="#">EA.Repository</a> [2850]	IN	An <b>EA.Repository</b> object representing the currently open Enterprise Architect model.  Poll its members to retrieve model data and user interface status information.
<b>sLocation</b>	<i>String</i>	IN	The name of the template requested; this should match the <i>location</i> attribute in the <ModelTemplates> section of an MDG Technology File.

#### Return Value

Return a string containing the XML export of the model that is being used as a template. Return an empty string if access to the template is denied; the Add-In is to handle user notification of the error.

#### Example

```
Public Function EA_OnRetrieveModelTemplate(ByRef Rep As EA.Repository,
ByRef sLocation As String) As String
    Dim sTemplate As String
    Select Case sLocation
        Case "Templates\Template1.xml"
            sTemplate = My.Resources.Template1
        Case "Templates\Template2.xml"
            sTemplate = My.Resources.Template2
        Case "Templates\Template3.xml"
            sTemplate = My.Resources.Template3
        Case Else
            MsgBox("Path for " & sLocation & " not found")
            sTemplate = ""
    End Select
    EA_OnRetrieveModelTemplate = sTemplate
End Function
```

#### Learn more

- [Incorporate Model Templates](#) <sup>[1576]</sup>

### 20.3.5.13 EA\_OnTabChanged

#### Topics

Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnTabChanged</i> notifies Add-Ins that the currently open tab has changed.</p> <p>Diagrams do <b>not</b> generate the message when they are first opened - use the broadcast event <i>EA_OnPostOpenDiagram</i> for this purpose.</p>	<a href="#">EA_OnPostOpenDiagram</a> <sup>[8044]</sup>

#### Syntax

**Function** *EA\_OnTabChanged*(*Repository As EA.Repository, TabName As String, DiagramID As Integer*)

The *EA\_OnTabChanges* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>DiagramID</b>	<i>Long</i>	IN	The diagram ID, or <b>0</b> if switched to an Add-In tab.
<b>Repository</b>	<a href="#">EA.Repository</a> <sup>[2850]</sup>	IN	<p>An EA.Repository object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>
<b>TabName</b>	<i>String</i>	IN	The name of the tab to which focus has been switched.

#### Return Value

None

### 20.3.5.14 Model Validation Broadcasts

#### Perform Model Validation from an Add-In

Using Enterprise Architect broadcasts, it is possible to define a set of rules that are evaluated when the user instructs Enterprise Architect to perform model validation. An Add-In that performs model validation would involve the following broadcast events:

Topic	Detail	Link
<b>EA_OnInitializeUserRu</b>	<i>EA_OnInitializeUserRules</i> is intercepted in order to	<a href="#">EA_OnInitializeUserRule</a>

Topic	Detail	Link
<b>les</b>	define rule categories and rules.	<a href="#">s</a> <sup>[3048]</sup>
<b>EA_OnStartValidation</b>	<i>EA_OnStartValidation</i> can be intercepted to perform any required processing prior to validation.	<a href="#">EA_OnStartValidation</a> <sup>[3049]</sup>
<b>Validate Request</b>	The following functions intercept each request to validate an individual element, package, diagram, connector, attribute and method.	
	<i>EA_OnRunElementRule</i>	<a href="#">EA_OnRunElementRule</a> <sup>[3050]</sup>
	<i>EA_OnRunPackageRule</i>	<a href="#">EA_OnRunPackageRule</a> <sup>[3051]</sup>
	<i>EA_OnRunDiagramRule</i>	<a href="#">EA_OnRunDiagramRule</a> <sup>[3052]</sup>
	<i>EA_OnRunConnectorRule</i>	<a href="#">EA_OnRunConnectorRule</a> <sup>[3053]</sup>
	<i>EA_OnRunAttributeRule</i>	<a href="#">EA_OnRunAttributeRule</a> <sup>[3053]</sup>
	<i>EA_OnRunMethodRule</i>	<a href="#">EA_OnRunMethodRule</a> <sup>[3054]</sup>
<b>EA_OnEndValidation</b>	<i>EA_OnEndValidation</i> can be intercepted to perform any required clean-up after validation has completed.	<a href="#">EA_OnEndValidation</a> <sup>[3050]</sup>

#### Learn more

- [Model Validation Example](#) <sup>[3056]</sup>

#### 20.3.5.14.1 *EA\_OnInitializeUserRules*

#### Topics

Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnInitializeUserRules</i> is called on Enterprise Architect start-up and requests that the Add-In provide Enterprise Architect with a rule category and list of rule IDs for model validation.</p> <p>This function must be implemented by any Add-In that is to perform its own model validation. It must call <i>Project</i>.</p>	<a href="#">Project Interface Class</a> <sup>[2962]</sup>



Topic	Detail	See also
	<i>DefineRuleCategory</i> once and <i>Project.DefineRule</i> for each rule; these functions are described in the <i>Project Interface</i> topic.	

### Syntax

#### Sub EA\_OnInitializeUserRules(*Repository* As *EA.Repository*)

The *EA\_OnInitializeUserRules* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Repository</b>	<a href="#">EA.Repository</a> <small>[2850]</small>	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.  Poll its members to retrieve model data and user interface status information.

#### 20.3.5.14.2 EA\_OnStartValidation

### Topics

Topic	Detail	See also
<b>Details</b>	<i>EA_OnStartValidation</i> notifies Add-Ins that a user has invoked the model validation command from Enterprise Architect.	

### Syntax

#### Sub EA\_OnStartValidation(*Repository* As *EA.Repository*, *ParamArray* *Args*() as *Variant*)

The *EA\_OnStartValidation* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Args</b>	<i>ParamArray of Variant</i>	IN	Contains a list of Rule Categories that are active for the current invocation of model validation.
<b>Repository</b>	<a href="#">EA.Repository</a> <small>[2850]</small>	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.  Poll its members to retrieve model data and user interface status information.

### 20.3.5.14.3 EA\_OnEndValidation


#### Topics

Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnEndValidation</i> notifies Add-Ins that model validation has completed.</p> <p>Use this event to arrange any clean-up operations arising from the validation.</p>	

#### Syntax


**Sub** EA\_OnEndValidation(*Repository As EA.Repository, ParamArray Args() as Variant*)

The *EA\_OnEndValidation* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Args</b>	<i>ParamArray of Variant</i>	IN	Contains a list of Rule Categories that were active for the invocation of model validation that has just completed.
<b>Repository</b>	<a href="#">EA.Repository</a> 	IN	<p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>

### 20.3.5.14.4 EA\_OnRunElementRule

#### Topics

Topic	Detail	See also
<b>Details</b>	<p>This event is triggered once for each rule defined in <i>EA_OnInitializeUserRules</i> to be performed on each element in the selection being validated.</p> <p>If you don't want to perform the rule defined by <b>RuleID</b> on the given element, then simply return without performing any action.</p> <p>On performing any validation, if a validation error is found, use the <i>Repository.ProjectInterface.PublishResult</i> method to notify Enterprise Architect.</p>	<a href="#">EA_OnInitializeUserRule</a> 

#### Syntax

**Sub EA\_OnRunElementRule(*Repository As EA.Repository, RuleID As String, Element As EA.Element*)**

The *EA\_OnRunElementRule* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Element</b>	<i>EA.Element</i>	IN	The element to potentially perform validation on.
<b>Repository</b>	<a href="#">EA.Repository</a> [2850]	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.  Poll its members to retrieve model data and user interface status information.
<b>RuleID</b>	<i>String</i>	IN	The ID that was passed into the <i>Project.DefineRule</i> command.

#### 20.3.5.14.5 EA\_OnRunPackageRule

##### Topics

Topic	Detail	See also
<b>Details</b>	<p>This event is triggered once for each rule defined in <i>EA_OnInitializeUserRules</i> to be performed on each package in the selection being validated.</p> <p>If you don't want to perform the rule defined by <b>RuleID</b> on the given package, then simply return without performing any action.</p> <p>On performing any validation, if a validation error is found, use the <i>Repository.ProjectInterface.PublishResult</i> method to notify Enterprise Architect.</p>	<a href="#">EA_OnInitializeUserRule</a> [3048]

##### Syntax

**Sub EA\_OnRunPackageRule(*Repository As EA.Repository, RuleID As String, PackageID As Long*)**

The *EA\_OnRunElementRule* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>PackageID</b>	<i>Long</i>	IN	The ID of the package to potentially perform validation on.  Use the <i>Repository.GetPackageByID</i> method to retrieve the package object.
<b>Repository</b>	<a href="#">EA.Repository</a> [2850]	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.

Parameter	Type	Direction	Description
			Poll its members to retrieve model data and user interface status information.
<b>RuleID</b>	<i>String</i>	IN	The ID that was passed into the <i>Project.DefineRule</i> method.

#### 20.3.5.14.6 EA\_OnRunDiagramRule

##### Topics

Topic	Detail	See also
<b>Details</b>	<p>This event is triggered once for each rule defined in <i>EA_OnInitializeUserRules</i> to be performed on each diagram in the selection being validated.</p> <p>If you don't want to perform the rule defined by <b>RuleID</b> on the given diagram, then simply return without performing any action.</p> <p>On performing any validation, if a validation error is found, use the <i>Repository.ProjectInterface.PublishResult</i> method to notify Enterprise Architect.</p>	<a href="#">EA_OnInitializeUserRules</a> <small>[3048]</small>

##### Syntax

**Sub EA\_OnRunDiagramRule(*Repository As EA.Repository, RuleID As String, DiagramID As Long*)**

The *EA\_OnRunDiagramRule* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>DiagramID</b>	<i>Long</i>	IN	<p>The ID of the diagram to potentially perform validation on.</p> <p>Use the <i>Repository.GetDiagramByID</i> method to retrieve the diagram object.</p>
<b>Repository</b>	<a href="#">EA.Repository</a> <small>[2850]</small>	IN	<p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>
<b>RuleID</b>	<i>String</i>	IN	The ID that was passed into the <i>Project.DefineRule</i> command.

#### 20.3.5.14.7 EA\_OnRunConnectorRule

##### Topics

Topic	Detail	See also
<b>Details</b>	<p>This event is triggered once for each rule defined in <i>EA_OnInitializeUserRules</i> to be performed on each connector in the selection being validated.</p> <p>If you don't want to perform the rule defined by <b>RuleID</b> on the given connector, then simply return without performing any action.</p> <p>On performing any validation, if a validation error is found, use the <i>Repository.ProjectInterface.PublishResult</i> method to notify Enterprise Architect.</p>	<a href="#">EA_OnInitializeUserRules</a> <small>[3048]</small>

##### Syntax

**Sub EA\_OnRunConnectorRule(*Repository As EA.Repository, RuleID As String, ConnectorID As Long*)**

The *EA\_OnRunConnectorRule* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>ConnectorID</b>	<i>Long</i>	IN	The ID of the connector to potentially perform validation on.  Use the <i>Repository.GetConnectorByID</i> method to retrieve the connector object.
<b>Repository</b>	<a href="#">EA.Repository</a> <small>[2850]</small>	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.
<b>RuleID</b>	<i>String</i>	IN	The ID that was passed into the <i>Project.DefineRule</i> command.

#### 20.3.5.14.8 EA\_OnRunAttributeRule

##### Topics

Topic	Detail	See also
<b>Details</b>	<p>This event is triggered once for each rule defined in <i>EA_OnInitializeUserRules</i> to be performed on each attribute in the selection being validated.</p> <p>If you don't want to perform the rule defined by <b>RuleID</b> on the given attribute, then simply return without performing any action.</p>	<a href="#">EA_OnInitializeUserRules</a> <small>[3048]</small>

Topic	Detail	See also
	On performing any validation, if a validation error is found, use the <i>Repository.ProjectInterface.PublishResult</i> method to notify Enterprise Architect.	

**Syntax:**

**Sub EA\_OnRunAttributeRule(*Repository As EA.Repository, RuleID As String, AttributeGUID As String, ObjectID As Long*)**

The *EA\_OnRunAttributeRule* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>AttributeGUID</b>	<i>String</i>	IN	The GUID of the attribute to potentially perform validation on. Use the <i>Repository.GetAttributeByGuid</i> method to retrieve the attribute object.
<b>ObjectID</b>	<i>Long</i>	IN	The ID of the object that owns the given attribute. Use the <i>Repository.GetElementByID</i> method to retrieve the object.
<b>Repository</b>	<a href="#">EA.Repository</a> <small>[2850]</small>	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.
<b>RuleID</b>	<i>String</i>	IN	The ID that was passed into the <i>Project.DefineRule</i> command.

#### 20.3.5.14.9 EA\_OnRunMethodRule

**Topics**

Topic	Detail	See also
<b>Details</b>	<p>This event is triggered once for each rule defined in <i>EA_OnInitializeUserRules</i> to be performed on each method in the selection being validated.</p> <p>If you don't want to perform the rule defined by <b>RuleID</b> on the given method, then simply return without performing any action.</p> <p>On performing any validation, if a validation error is found, use the <i>Repository.ProjectInterface.PublishResult</i> method to notify Enterprise Architect.</p>	<a href="#">EA_OnInitializeUserRules</a> <small>[3048]</small>

Topic	Detail	See also

**Syntax:**

**Sub EA\_OnRunMethodRule**(*Repository As EA.Repository, RuleID As String, MethodGUID As String, ObjectID As Long*)

The *EA\_OnRunMethodRule* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>MethodGUID</b>	<i>String</i>	IN	The GUID of the method to potentially perform validation on. Use the <i>Repository.GetMethodByGuid</i> method to retrieve the method object.
<b>ObjectID</b>	<i>Long</i>	IN	The ID of the object that owns the given method. Use the <i>Repository.GetElementByID</i> method to retrieve the object.
<b>Repository</b>	<a href="#">EA.Repository</a> <small>[2850]</small>	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.  Poll its members to retrieve model data and user interface status information.
<b>RuleID</b>	<i>String</i>	IN	The ID that was passed into the <i>Project.DefineRule</i> command.

#### 20.3.5.14.10 EA\_OnRunParameterRule

**Topics**

Topic	Detail	See also
<b>Details</b>	<p>This event is triggered once for each rule defined in <i>EA_OnInitializeUserRules</i> to be performed on each parameter in the selection being validated.</p> <p>If you don't want to perform the rule defined by <b>RuleID</b> on the given parameter, then simply return without performing any action.</p> <p>On performing any validation, if a validation error is found, use the <i>Repository.ProjectInterface.PublishResult</i> method to notify Enterprise Architect.</p>	<a href="#">EA_OnInitializeUserRule</a> <small>[3048]</small>

**Syntax**

***Sub EA\_OnRunParameterRule(Repository As EA.Repository, RuleID As String, ParameterGUID As String, MethodGUID As String, ObjectID As Long)***

The *EA\_OnRunMethodRule* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>MethodGUID</b>	<i>String</i>	IN	The GUID of the method that owns the given parameter. Use the <i>Repository.GetMethodByGuid</i> method to retrieve the method object.
<b>ObjectID</b>	<i>Long</i>	IN	The ID of the object that owns the given parameter. Use the <i>Repository.GetElementByID</i> method to retrieve the object.
<b>ParameterGUID</b>	<i>String</i>	IN	The GUID of the parameter to potentially perform validation on. Use this to retrieve the parameter by iterating through the <i>Method.Parameters</i> collection.
<b>Repository</b>	<a href="#">EA.Repository</a> <small>[2850]</small>	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model. Poll its members to retrieve model data and user interface status information.
<b>RuleID</b>	<i>String</i>	IN	The ID that was passed into the <i>Project.DefineRule</i> command.

#### 20.3.5.14.11 Model Validation Example

The following example code is written in C# and provides a skeleton model validation implementation that you might like to use as a starting point in writing your own model validation rules.

##### Main.cs

```
using System;

namespace myAddIn
{
    public class Main
    {
        public Rules theRules;

        public Main()
        {
            theRules = new Rules();
        }

        public string EA_Connect(EA.Repository Repository)
        {
            return "";
        }
    }
}
```



```

public void EA_Disconnect()
{
    GC.Collect();
    GC.WaitForPendingFinalizers();
}

private bool IsProjectOpen(EA.Repository Repository)
{
    try
    {
        EA.Collection c = Repository.Models;
        return true;
    }
    catch
    {
        return false;
    }
}

public object EA_GetMenuItems(EA.Repository Repository, string
MenuLocation, string MenuName)
{
    switch (MenuName)
    {
        case "":
            return "- &myAddin";
        case "- &myAddin":
            string( ) ar = { "&Test" };
            return ar;
    }
    return "";
}

public void EA_GetMenuState(EA.Repository Repository, string
MenuLocation, string MenuName, string ItemName, ref bool IsEnabled, ref
bool IsChecked)
{
    // if no open project, disable all menu options
    if (IsProjectOpen(Repository))
        IsEnabled = true;
    else
        IsEnabled = false;
}

public void EA_MenuClick(EA.Repository Repository, string
MenuLocation, string MenuName, string ItemName)
{
    switch (ItemName)
    {
        case "&Test";
            DoTest(Repository);
            break;
    }
}

public void EA_OnInitializeUserRules(EA.Repository Repository)
{
    if (Repository != null)
    {
        theRules.ConfigureCategories(Repository);
        theRules.ConfigureRules(Repository);
    }
}

public void EA_OnRunElementRule(EA.Repository Repository,
string RuleID, EA.Element element)
{
    theRules.RunElementRule(Repository, RuleID, element);
}

```

```

        public void EA_OnRunDiagramRule(EA.Repository Repository,
string RuleID, long IDiagramID)
        {
            theRules.RunDiagramRule(Repository, RuleID, IDiagramID);
        }

        public void EA_OnRunConnectorRule(EA.Repository Repository,
string RuleID, long IDConnectorID)
        {
            theRules.RunConnectorRule(Repository, RuleID,
IDConnectorID);
        }

        public void EA_OnRunAttributeRule(EA.Repository Repository,
string RuleID, string AttributeID, long IObjectID)
        {
            return;
        }

        public void EA_OnDeleteTechnology(EA.Repository Repository, EA.
Event Properties Info)
        {
            return;
        }

        public void EA_OnImportTechnology(EA.Repository Repository, EA.
Event Properties Info)
        {
            return;
        }

        private void DoTest(EA.Repository Rep)
        {
            // TODO: insert test code here
        }
    }
}

```

**Rules.cs**

```

using System;
using System.Collections;

namespace myAddin
{
    public class Rules
    {
        private string m_sCategoryID;
        private System.Collections.ArrayList m_RuleIDs;
        private System.Collections.ArrayList m_RuleDEx;

        private const string cRule01 = "Rule01";
        private const string cRule02 = "Rule02";
        private const string cRule03 = "Rule03";
        // TODO: expand this list as much as necessary

        public Rules()
        {
            m_RuleIDs = new System.Collections.ArrayList();
            m_RuleDEx = new System.Collections.ArrayList();
        }

        private string LookupMap(string sKey)
        {
            return DoLookupMap(sKey, m_RuleIDs, m_RuleDEx);
        }
    }
}

```

```

private string LookupMapEx(string sRule)
{
    return DoLookupMap(sRule, m_RuleIDEx, m_RuleIDs);
}

private string DoLookupMap(string sKey, ArrayList arrValues,
ArrayList arrKeys)
{
    if (arrKeys.Contains(sKey))
        return arrValues[arrKeys.IndexOf(sKey)].ToString
();
    else
        return "";
}

private void AddToMap(string sRuleID, string sKey)
{
    m_RuleIDs.Add(sRuleID);
    m_RuleIDEx.Add(sKey);
}

private string GetRuleStr(string sRuleID)
{
    switch (sRuleID)
    {
        case cRule01:
            return "Error Message 01";
        case cRule02:
            return "Error Message 02";
        case cRule03:
            return "Error Message 03";
        // TODO: add extra cases as much as necessary
    }
    return "";
}

public void ConfigureCategories(EA.Repository Repository)
{
    EA.Project Project = Repository.GetProjectInterface();
    m_sCategoryID = Project.DefineRuleCategory("Enterprise
Collaboration Architecture (ECA) Rules");
}

public void ConfigureRules(EA.Repository Repository)
{
    EA.Project Project = Repository.GetProjectInterface();
    AddToMap(Project.DefineRule(m_sCategoryID, EA.
EnumMVErrType.mvError, GetRuleStr(cRule01)), cRule01);
    AddToMap(Project.DefineRule(m_sCategoryID, EA.
EnumMVErrType.mvError, GetRuleStr(cRule02)), cRule02);
    AddToMap(Project.DefineRule(m_sCategoryID, EA.
EnumMVErrType.mvError, GetRuleStr(cRule03)), cRule03);
    // TODO: expand this list
}

public void RunConnectorRule(EA.Repository Repository, string
sRuleID, long lConnectorID)
{
    EA.Connector Connector = Repository.GetConnectorByID
((int)lConnectorID);
    if (Connector != null)
    {
        switch (LookupMapEx(sRuleID))
        {
            case cRule02:
                // TODO: perform rule 2 check
                break;
            // TODO: add more cases
        }
    }
}

```

```

        public void RunDiagramRule(EA.Repository Repository, string
sRuleID, long IDiagramID)
        {
            EA.Diagram Diagram = Repository.GetDiagramByID((int)
IDiagramID);
            if (Diagram != null)
            {
                switch (LookupMapEx(sRuleID))
                {
                    case cRule03:
                        // TODO: perform rule 3 check
                        break;
                        // TODO: add more cases
                }
            }
        }

        public void RunElementRule(EA.Repository Repository, string
sRuleID, EA.Element Element)
        {
            if (Element != null)
            {
                switch (LookupMapEx(sRuleID))
                {
                    case cRule01:
                        DoRule01(Repository, Element);
                        break;
                        // TODO: add more cases
                }
            }
        }

        private void DoRule01(EA.Repository Repository, EA.Element
Element)
        {
            if (Element.Stereotype != "myStereotype")
                return;

            // TODO: validation logic here





            // report validation errors
            EA.Project Project = Repository.GetProjectInterface();
            Project.PublishResult(LookupMap(cRule01), EA.
EnumMVErrType.mvError, GetRuleStr(cRule01));
        }
    }
}

```

### 20.3.5.15 Post-New Events


Enterprise Architect Add-Ins can respond to the creation of new elements, connectors, objects, attributes, methods and packages using the following broadcast events:

Topic	Link
EA_OnPostNewElement	<a href="#">EA_OnPostNewElement</a> <sup>[3061]</sup>
EA_OnPostNewConnector	<a href="#">EA_OnPostNewConnector</a> <sup>[3062]</sup>
EA_OnPostNewDiagram	<a href="#">EA_OnPostNewDiagram</a> <sup>[3063]</sup>
EA_OnPostNewDiagramObject	<a href="#">EA_OnPostNewDiagramObject</a> <sup>[3063]</sup>

Topic	Link
EA_OnPostNewAttribute	<a href="#">EA_OnPostNewAttribute</a> 
EA_OnPostNewMethod	<a href="#">EA_OnPostNewMethod</a> 
EA_OnPostNewPackage	<a href="#">EA_OnPostNewPackage</a> 
EA_OnPostNewGlossaryTerm	<a href="#">EA_OnPostNewGlossaryTerm</a> 

### 20.3.5.15.1 EA\_OnPostNewElement



#### Topics

Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnPostNewElement</i> notifies Add-Ins that a new element has been created on a diagram. It enables Add-Ins to modify the element upon creation.</p> <p>This event occurs after a user has dragged a new element from the Toolbox or Resources window onto a diagram. The notification is provided immediately after the element is added to the model.</p> <p>Set <i>Repository.SuppressEADialogs</i> to <b>true</b> to suppress Enterprise Architect from showing its default properties dialog.</p>	<a href="#">EA_OnPreNewElement</a> 

#### Syntax:

**Function** *EA\_OnPostNewElement*(*Repository As EA.Repository, Info As EA.EventProperties*) **As Boolean**

The *EA\_OnPostNewElement* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Info</b>	<a href="#">EA.EventProperties</a> 	IN	Contains the following <i>EventProperty</i> objects for the new element: <ul style="list-style-type: none"> <li><i>ElementID</i>: A long value corresponding to <i>Element.ElementID</i></li> </ul>
<b>Repository</b>	<a href="#">EA.Repository</a> 	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.

Parameter	Type	Direction	Description
			Poll its members to retrieve model data and user interface status information.

**Return Value:**

Return **True** if the element has been updated during this notification. Return **False** otherwise.

**20.3.5.15.2 EA\_OnPostNewConnector****Topics**

Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnPostNewConnector</i> notifies Add-Ins that a new connector has been created on a diagram. It enables Add-Ins to modify the connector upon creation.</p> <p>This event occurs after a user has dragged a new connector from the Toolbox or Resources window onto a diagram. The notification is provided immediately after the connector is added to the model.</p>	<a href="#">EA_OnPreNewConnector</a> [3077]

**Syntax:**

**Function** *EA\_OnPostNewConnector*(*Repository* As *EA.Repository*, *Info* As *EA.EventProperties*) As *Boolean*

The *EA\_OnPostNewConnector* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Info</b>	<a href="#">EA.EventProperties</a> [2833]	IN	<p>Contains the following <i>EventProperty</i> objects for the new connector:</p> <ul style="list-style-type: none"> <li><i>ConnectorID</i>: A long value corresponding to <i>Connector.ConnectorID</i></li> </ul>
<b>Repository</b>	<a href="#">EA.Repository</a> [2850]	IN	<p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>

**Return Value:**

Return **True** if the connector has been updated during this notification. Return **False** otherwise.

### 20.3.5.15.3 EA\_OnPostNewDiagram

#### Topics

Topic	Detail	See also
<b>Details</b>	<i>EA_OnPostNewDiagram</i> notifies Add-Ins that a new diagram has been created. It enables Add-Ins to modify the diagram upon creation.	<a href="#">EA_OnPreNewDiagram</a> [3078]

#### Syntax:

**Function** *EA\_OnPostNewDiagram*(*Repository* As *EA.Repository*, *Info* As *EA.EventProperties*) As *Boolean*

The *EA\_OnPostNewDiagram* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Info</b>	<a href="#">EA.EventProperties</a> [2833]	IN	Contains the following <i>EventProperty</i> objects for the new diagram: <ul style="list-style-type: none"> <li><i>DiagramID</i>: A long value corresponding to <i>Diagram</i>. <i>PackageID</i></li> </ul>
<b>Repository</b>	<a href="#">EA.Repository</a> [2850]	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.  Poll its members to retrieve model data and user interface status information.

#### Return Value:

Return **True** if the diagram has been updated during this notification. Return **False** otherwise.

### 20.3.5.15.4 EA\_OnPostNewDiagramObject

**EA\_OnPostNewDiagramObject** notifies Add-Ins that a new object has been created on a diagram. It enables Add-Ins to modify the object upon creation.

This event occurs after a user has dragged a new object from the Project Browser or Resources window onto a diagram. The notification is provided immediately after the object is added to the diagram.

#### Syntax

**Function** *EA\_OnPostNewDiagramObject*(*Repository* As *EA.Repository*, *Info* As *EA.EventProperties*) As *Boolean*

The *EA\_OnPostNewDiagramObject* function syntax contains the following parameters:

Parameter	Type	Direction	Description
<b>Info</b>	<a href="#">EA.EventProperties</a> <sup>[2833]</sup>	IN	Contains the following <i>EventProperty</i> objects for the new element: <ul style="list-style-type: none"> <li><i>ID</i>: A long value corresponding to the <i>ElementID</i> of the object that has been added to the diagram</li> <li><i>DiagramID</i>: A long value corresponding to the <i>DiagramID</i> of the diagram to which the object has been added</li> <li><i>DUID</i>: A string value for the DUID; can be used with <i>Diagram.GetDiagramObjectByID</i> to retrieve the new <i>DiagramObject</i></li> </ul>
<b>Repository</b>	<a href="#">EA.Repository</a> <sup>[2850]</sup>	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.  Poll its members to retrieve model data and user interface status information.

#### Return Value

Return **True** if the element has been updated during this notification. Return **False** otherwise.

#### Learn more

- [EA\\_OnPreNewDiagramObject](#)<sup>[3078]</sup>

### 20.3.5.15.5 EA\_OnPostNewAttribute

#### Topics

Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnPostNewAttribute</i> notifies Add-Ins that a new attribute has been created on a diagram. It enables Add-Ins to modify the attribute upon creation.</p> <p>This event occurs when a user creates a new attribute on an element by either drag-dropping from the Project Browser, using the Attributes Properties dialog, or using the in-place editor on the diagram. The notification is provided immediately after the attribute is created.</p>	<a href="#">EA_OnPreNewAttribute</a> <sup>[3080]</sup>

#### Syntax:

**Function** *EA\_OnPostNewAttribute*(*Repository As EA.Repository, Info As EA.EventProperties*) **As Boolean**

The *EA\_OnPostNewAttribute* function syntax contains the following elements:



Parameter	Type	Direction	Description
<b>Info</b>	<a href="#">EA.EventProperties</a> [2833]	IN	Contains the following <i>EventProperty</i> objects for the new attribute: <ul style="list-style-type: none"> <li><i>AttributeID</i>: A long value corresponding to <i>Attribute.AttributeID</i></li> </ul>
<b>Repository</b>	<a href="#">EA.Repository</a> [2850]	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.  Poll its members to retrieve model data and user interface status information.

**Return Value:**

Return **True** if the attribute has been updated during this notification. Return **False** otherwise.

**20.3.5.15.6 EA\_OnPostNewMethod****Topics**

Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnPostNewMethod</i> notifies Add-Ins that a new method has been created on a diagram. It enables Add-Ins to modify the method upon creation.</p> <p>This event occurs when a user creates a new method on an element by either drag-dropping from the Project Browser, using the method's Properties dialog, or using the in-place editor on the diagram. The notification is provided immediately after the method is created.</p>	<a href="#">EA_OnPreNewMethod</a> [3081]

**Syntax:**

**Function** *EA\_OnPostNewMethod(Repository As EA.Repository, Info As EA.EventProperties) As Boolean*

The *EA\_OnPostNewMethod* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Info</b>	<a href="#">EA.EventProperties</a> [2833]	IN	Contains the following <i>EventProperty</i> objects for the new method: <ul style="list-style-type: none"> <li><i>MethodID</i>: A long value corresponding to <i>Method.MethodID</i></li> </ul>
<b>Repository</b>	<a href="#">EA.Repository</a> [2850]	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.

Parameter	Type	Direction	Description
			Poll its members to retrieve model data and user interface status information.

**Return Value:**

Return **True** if the method has been updated during this notification. Return **False** otherwise.

**20.3.5.15.7 EA\_OnPostNewPackage****Topics**

Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnPostNewPackage</i> notifies Add-Ins that a new package has been created on a diagram. It enables Add-Ins to modify the package upon creation.</p> <p>This event occurs when a user drags a new package from the Toolbox or Resources window onto a diagram, or by selecting the <b>New Package</b> icon from the Project Browser.</p>	<a href="#">EA_OnPreNewPackage</a> <small>3062</small>

**Syntax:**

**Function** *EA\_OnPostNewPackage*(*Repository As EA.Repository, Info As EA.EventProperties*) **As Boolean**

The *EA\_OnPostNewPackage* function syntax contains the following elements:


Parameter	Type	Direction	Description
<b>Info</b>	<a href="#">EA.EventProperties</a> <small>2833</small>	IN	<p>Contains the following <i>EventProperty</i> objects for the new package:</p> <ul style="list-style-type: none"> <li><i>PackageID</i>: A long value corresponding to <i>Package.PackageID</i></li> </ul>
<b>Repository</b>	<a href="#">EA.Repository</a> <small>2850</small>	IN	<p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>

**Return Value:**

Return **True** if the package has been updated during this notification. Return **False** otherwise.

### 20.3.5.15.8 EA\_OnPostNewGlossaryTerm



#### Topics

Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnPostNewGlossaryTerm</i> notifies Add-Ins that a new glossary term has been created. It enables Add-Ins to modify the glossary term upon creation.</p> <p>The notification is provided immediately after the glossary term is added to the model.</p>	<a href="#">EA_OnPreNewGlossaryTerm</a> 

#### Syntax:

**Function** *EA\_OnPostNewGlossaryTerm*(*Repository* As *EA.Repository*, *Info* As *EA.EventProperties*) As *Boolean*

The *EA\_OnPostNewGlossaryTerm* function syntax contains the following elements:



Parameter	Type	Direction	Description
<b>Info</b>	<a href="#">EA.EventProperties</a> 	IN	<p>Contains the following <i>EventProperty</i> objects for the new glossary term:</p> <ul style="list-style-type: none"> <li><i>ElementID</i>: A long value corresponding to <i>Element.ElementID</i></li> </ul>
<b>Repository</b>	<a href="#">EA.Repository</a> 	IN	<p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>

#### Return Value:

Return **True** if the glossary term has been updated during this notification. Return **False** otherwise.

### 20.3.5.16 Pre-Deletion Events

Enterprise Architect Add-Ins can respond to requests to delete elements, attributes, methods, connectors, diagrams, packages and glossary terms using the following broadcast events:

Topic	Link
EA_OnPreDeleteElement	<a href="#">EA_OnPreDeleteElement</a> 
EA_OnPreDeleteAttribute	<a href="#">EA_OnPreDeleteAttribute</a> 

Topic	Link
EA_OnPreDeleteMethod	<a href="#">EA_OnPreDeleteMethod</a>  <sup>[3070]</sup>
EA_OnPreDeleteConnector	<a href="#">EA_OnPreDeleteConnector</a>  <sup>[3071]</sup>
EA_OnPreDeleteDiagram	<a href="#">EA_OnPreDeleteDiagram</a>  <sup>[3071]</sup>
EA_OnPreDeletePackage	<a href="#">EA_OnPreDeletePackage</a>  <sup>[3073]</sup>
EA_OnPreDeleteGlossaryTerm	<a href="#">EA_OnPreDeleteGlossaryTerm</a>  <sup>[3074]</sup>
EA_OnPreDeleteTechnology (Deprecated)	<a href="#">EA_OnPreDeleteTechnology</a>  <sup>[3091]</sup>

#### 20.3.5.16.1 EA\_OnPreDeleteElement


##### Topics

Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnPreDeleteElement</i> notifies Add-Ins that an element is to be deleted from the model. It enables Add-Ins to permit or deny deletion of the element.</p> <p>This event occurs when a user deletes an element from the Project Browser or on a diagram. The notification is provided immediately before the element is deleted, so that the Add-In can disable deletion of the element.</p>	

##### Syntax:

**Function** *EA\_OnPreDeleteElement*(*Repository As EA.Repository, Info As EA.EventProperties*) *As Boolean*

The *EA\_OnPreDeleteElement* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Info</b>	<a href="#">EA.EventProperties</a>  <sup>[2833]</sup>	IN	<p>Contains the following <i>EventProperty</i> objects for the element to be deleted:</p> <ul style="list-style-type: none"> <li><i>ElementID</i>: A long value corresponding to <i>Element</i>. <i>ElementID</i></li> </ul>

Parameter	Type	Direction	Description
<b>Repository</b>	<a href="#">EA.Repository</a> <small>[2850]</small>	IN	An EA.Repository object representing the currently open Enterprise Architect model.  Poll its members to retrieve model data and user interface status information.

**Return Value:**

Return **True** to enable deletion of the element from the model. Return **False** to disable deletion of the element.

**20.3.5.16.2 EA\_OnPreDeleteAttribute****Topics**

Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnPreDeleteAttribute</i> notifies Add-Ins that an attribute is to be deleted from the model. It enables Add-Ins to permit or deny deletion of the attribute.</p> <p>This event occurs when a user attempts to permanently delete an attribute from the Project Browser. The notification is provided immediately before the attribute is deleted, so that the Add-In can disable deletion of the attribute.</p>	

**Syntax:**

**Function** *EA\_OnPreDeleteAttribute*(*Repository As EA.Repository, Info As EA.EventProperties*) *As Boolean*

The *EA\_OnPreDeleteAttribute* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Info</b>	<a href="#">EA.EventProperties</a> <small>[2833]</small>	IN	Contains the following <i>EventProperty</i> objects for the attribute to be deleted: <ul style="list-style-type: none"> <li><i>AttributeID</i>: A long value corresponding to <i>Attribute.AttributeID</i></li> </ul>
<b>Repository</b>	<a href="#">EA.Repository</a> <small>[2850]</small>	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.  Poll its members to retrieve model data and user interface status information.

**Return Value:**

Return **True** to enable deletion of the attribute from the model. Return **False** to disable deletion of the attribute.

**20.3.5.16.3 EA\_OnPreDeleteMethod****Topics**

Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnPreDeleteMethod</i> notifies Add-Ins that a method (operation) is to be deleted from the model. It enables Add-Ins to permit or deny deletion of the method.</p> <p>This event occurs when a user attempts to permanently delete a method from the Project Browser. The notification is provided immediately before the method is deleted, so that the Add-In can disable deletion of the method.</p>	

**Syntax:**

**Function** *EA\_OnPreDeleteMethod*(*Repository As EA.Repository, Info As EA.EventProperties*) **As Boolean**

The *EA\_OnPreDeleteMethod* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Info</b>	<a href="#">EA.EventProperties</a> [2833]	IN	<p>Contains the following <i>EventProperty</i> objects for the method to be deleted:</p> <ul style="list-style-type: none"> <li><i>MethodID</i>: A long value corresponding to <i>Method</i>. <i>MethodID</i></li> </ul>
<b>Repository</b>	<a href="#">EA.Repository</a> [2850]	IN	<p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>

**Return Value:**

Return **True** to enable deletion of the method from the model. Return **False** to disable deletion of the method.

#### 20.3.5.16.4 EA\_OnPreDeleteConnector

##### Topics

Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnPreDeleteConnector</i> notifies Add-Ins that a connector is to be deleted from the model. It enables Add-Ins to permit or deny deletion of the connector.</p> <p>This event occurs when a user attempts to permanently delete a connector on a diagram. The notification is provided immediately before the connector is deleted, so that the Add-In can disable deletion of the connector.</p>	

##### Syntax:

**Function** *EA\_OnPreDeleteConnector*(*Repository As EA.Repository, Info As EA.EventProperties*) **As Boolean**

The *EA\_OnPreDeleteConnector* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Info</b>	<a href="#">EA.EventProperties</a> <small>[2833]</small>	IN	<p>Contains the following <i>EventProperty</i> objects for the connector to be deleted:</p> <ul style="list-style-type: none"> <li><i>ConnectorID</i>: A long value corresponding to <i>Connector.ConnectorID</i></li> </ul>
<b>Repository</b>	<a href="#">EA.Repository</a> <small>[2850]</small>	IN	<p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>

##### Return Value:

Return **True** to enable deletion of the connector from the model. Return **False** to disable deletion of the connector.

#### 20.3.5.16.5 EA\_OnPreDeleteDiagram

##### Topics



Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnPreDeleteDiagram</i> notifies Add-Ins that a diagram is to be deleted from the model. It enables Add-Ins to permit or deny deletion of the diagram.</p>	

Topic	Detail	See also
	This event occurs when a user attempts to permanently delete a diagram from the Project Browser. The notification is provided immediately before the diagram is deleted, so that the Add-In can disable deletion of the diagram.	

**Syntax:**

**Function** *EA\_OnPreDeleteDiagram(Repository As EA.Repository, Info As EA.EventProperties) As Boolean*

The *EA\_OnPreDeleteDiagram* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Info</b>	<a href="#">EA.EventProperties</a> 	IN	Contains the following <i>EventProperty</i> objects for the diagram to be deleted: <ul style="list-style-type: none"> <li><i>DiagramID</i>: A long value corresponding to <i>Diagram</i>. <i>DiagramID</i></li> </ul>
<b>Repository</b>	<a href="#">EA.Repository</a> 	IN	An <i>EA.Repository</i> object representing the currently-open Enterprise Architect model.  Poll its members to retrieve model data and user interface status information.

**Return Value:**

Return **True** to enable deletion of the diagram from the model. Return **False** to disable deletion of the diagram.

### 20.3.5.16.6 EA\_OnPreDeleteDiagramObject

**Topics**

Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnPreDeleteDiagramObject</i> notifies Add-Ins that a diagram object is to be deleted from the model. It enables Add-Ins to permit or deny deletion of the element.</p> <p>This event occurs when a user attempts to permanently delete an element from a diagram. The notification is provided immediately before the element is deleted, so that the Add-In can disable deletion of the element.</p>	

**Syntax:**



**Function** *EA\_OnPreDeleteDiagramObject*(*Repository As EA.Repository, Info As EA.EventProperties*) *As Boolean*

The *EA\_OnPreDeleteDiagramObject* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Info</b>	<a href="#">EA.EventProperties</a> <small>2833</small>	IN	Contains the following <i>EventProperty</i> objects for the element to be deleted: <ul style="list-style-type: none"> <li>ID: A long value corresponding to <i>Diagram Object.ElementID</i></li> </ul>
<b>Repository</b>	<a href="#">EA.Repository</a> <small>2850</small>	IN	An <i>EA.Repository</i> object representing the currently-open Enterprise Architect model.  Poll its members to retrieve model data and user interface status information.

#### Return Value:

Return **True** to enable deletion of the element from the model. Return **False** to disable deletion of the element.

### 20.3.5.16.7 *EA\_OnPreDeletePackage*

#### Topics



Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnPreDeletePackage</i> notifies Add-Ins that a package is to be deleted from the model. It enables Add-Ins to permit or deny deletion of the package.</p> <p>This event occurs when a user attempts to permanently delete a package from the Project Browser. The notification is provided immediately before the package is deleted, so that the Add-In can disable deletion of the package.</p>	

#### Syntax:

**Function** *EA\_OnPreDeletePackage*(*Repository As EA.Repository, Info As EA.EventProperties*) *As Boolean*

The *EA\_OnPreDeletePackage* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Info</b>	<a href="#">EA.EventProperties</a>	IN	Contains the following <i>EventProperty</i> objects for the package to be deleted:

Parameter	Type	Direction	Description
			<ul style="list-style-type: none"> <li><i>PackageID</i>: A long value corresponding to <i>Package</i>. <i>PackageID</i></li> </ul>
<b>Repository</b>	<a href="#">EA.Repository</a> 	IN	<p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>

**Return Value:**

Return **True** to enable deletion of the package from the model. Return **False** to disable deletion of the package.



**20.3.5.16.8 EA\_OnPreDeleteGlossaryTerm****Topics**

Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnPreDeleteGlossaryTerm</i> notifies Add-Ins that a glossary term is to be deleted from the model. It enables Add-Ins to permit or deny deletion of the glossary term.</p> <p>The notification is provided immediately before the glossary term is deleted, so that the Add-In can disable deletion of the glossary term.</p>	

**Syntax:**

**Function** *EA\_OnPreDeleteGlossaryTerm*(*Repository* As *EA.Repository*, *Info* As *EA.EventProperties*) As *Boolean*

The *EA\_OnPreDeleteGlossaryTerm* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Info</b>	<a href="#">EA.EventProperties</a> 	IN	<p>Contains the following <i>EventProperty</i> objects for the glossary term to be deleted:</p> <ul style="list-style-type: none"> <li><i>TermID</i>: A long value corresponding to <i>Term.TermID</i></li> </ul>
<b>Repository</b>	<a href="#">EA.Repository</a> 	IN	<p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>

Parameter	Type	Direction	Description










**Return Value:**

Return **True** to enable deletion of the glossary term from the model. Return **False** to disable deletion of the glossary term.

**20.3.5.17 Pre New-Object Events**

When you create an Add-In, you can include broadcast events to intercept and respond to requests to create new objects, including elements, connectors, diagram objects, attributes, methods and packages.

**Events to intercept**

Event	Link
Creation of a new element	<a href="#">EA_OnPreNewElement</a>  <sup>[3076]</sup>
Creation of a new connector	<a href="#">EA_OnPreNewConnector</a>  <sup>[3077]</sup>
Creation of a new diagram	<a href="#">EA_OnPreNewDiagram</a>  <sup>[3078]</sup>
Creation of a new diagram object	<a href="#">EA_OnPreNewDiagramObject</a>  <sup>[3078]</sup>
Creation of a new element by dropping onto a diagram from the Project Browser or Diagram Toolbox	<a href="#">EA_OnPreDropFromTree</a>  <sup>[3079]</sup>
Creation of a new attribute	<a href="#">EA_OnPreNewAttribute</a>  <sup>[3080]</sup>
Creation of a new method	<a href="#">EA_OnPreNewMethod</a>  <sup>[3081]</sup>
Creation of a new package	<a href="#">EA_OnPreNewPackage</a>  <sup>[3082]</sup>
Creation of a new glossary term	<a href="#">EA_OnPreNewGlossaryTerm</a>  <sup>[3083]</sup>

### 20.3.5.17.1 EA\_OnPreNewElement

#### Topics

Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnPreNewElement</i> notifies Add-Ins that a new element is about to be created on a diagram. It enables Add-Ins to permit or deny creation of the new element.</p> <p>This event occurs when a user drags a new element from the Toolbox or Resources window onto a diagram. The notification is provided immediately before the element is created, so that the Add-In can disable addition of the element.</p>	<a href="#">EA_OnPostNewElement</a> [3067]

#### Syntax:

**Function** *EA\_OnPreNewElement*(*Repository As EA.Repository, Info As EA.EventProperties*) **As Boolean**

The *EA\_OnPreNewElement* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Info</b>	<a href="#">EA.EventProperties</a> [2833]	IN	<p>Contains the following <i>EventProperty</i> objects for the element to be created:</p> <ul style="list-style-type: none"> <li><i>Type</i>: A string value corresponding to <i>Element.Type</i></li> <li><i>Stereotype</i>: A string value corresponding to <i>Element.Stereotype</i></li> <li><i>ParentID</i>: A long value corresponding to <i>Element.ParentID</i></li> <li><i>DiagramID</i>: A long value corresponding to the ID of the diagram to which the element is being added</li> </ul>
<b>Repository</b>	<a href="#">EA.Repository</a> [2850]	IN	<p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>

#### Return Value:

Return **True** to enable addition of the new element to the model. Return **False** to disable addition of the new element.

### 20.3.5.17.2 EA\_OnPreNewConnector

#### Topics

Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnPreNewConnector</i> notifies Add-Ins that a new connector is about to be created on a diagram. It enables Add-Ins to permit or deny creation of a new connector.</p> <p>This event occurs when a user drags a new connector from the Toolbox or Resources window, onto a diagram. The notification is provided immediately before the connector is created, so that the Add-In can disable addition of the connector.</p>	<a href="#">EA_OnPostNewConnector</a>

#### Syntax:

**Function** *EA\_OnPreNewConnector* (*Repository As EA.Repository, Info As EA.EventProperties*) **As Boolean**

The *EA\_OnPreNewConnector* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Info</b>	<a href="#">EA.EventProperties</a>	IN	<p>Contains the following <i>EventProperty</i> objects for the connector to be created:</p> <ul style="list-style-type: none"> <li><i>Type</i>: A string value corresponding to <i>Connector.Type</i></li> <li><i>Subtype</i>: A string value corresponding to <i>Connector.Subtype</i></li> <li><i>Stereotype</i>: A string value corresponding to <i>Connector.Stereotype</i></li> <li><i>ClientID</i>: A long value corresponding to <i>Connector.ClientID</i></li> <li><i>SupplierID</i>: A long value corresponding to <i>Connector.SupplierID</i></li> <li><i>DiagramID</i>: A long value corresponding to <i>Connector.DiagramID</i></li> </ul>
<b>Repository</b>	<a href="#">EA.Repository</a>	IN	<p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>

#### Return Value:

Return **True** to enable addition of the new connector to the model. Return **False** to disable addition of the new connector.

### 20.3.5.17.3 EA\_OnPreNewDiagram

#### Topics

Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnPreNewDiagram</i> notifies Add-Ins that a new diagram is about to be created. It enables Add-Ins to permit or deny creation of the new diagram.</p> <p>The notification is provided immediately before the diagram is created, so that the Add-In can disable addition of the diagram.</p>	<a href="#">EA_OnPostNewDiagram</a> [3063]

#### Syntax:

**Function** *EA\_OnPreNewDiagram*(*Repository As EA.Repository, Info As EA.EventProperties*) **As Boolean**

The *EA\_OnPreNewDiagram* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Info</b>	<a href="#">EA.EventProperties</a> [2833]	IN	Contains the following <i>EventProperty</i> objects for the diagram to be created: <ul style="list-style-type: none"> <li><i>Type</i>: A string value corresponding to <i>Diagram.Type</i></li> <li><i>ParentID</i>: A long value corresponding to <i>Diagram.ParentID</i></li> <li><i>PackageID</i>: A long value corresponding to <i>Diagram.PackageID</i></li> </ul>
<b>Repository</b>	<a href="#">EA.Repository</a> [2850]	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.  Poll its members to retrieve model data and user interface status information.

#### Return Value:

Return **True** to enable addition of the new diagram to the model. Return **False** to disable addition of the new diagram.

### 20.3.5.17.4 EA\_OnPreNewDiagramObject

#### Topics

Topic	Detail	See also
<b>Details</b>	<i>EA_OnPreNewDiagramObject</i> notifies Add-Ins that a new	<a href="#">EA_OnPostNewDiagram</a>

Topic	Detail	See also
	<p>diagram object is about to be dropped on a diagram. It enables Add-Ins to permit or deny creation of the new object.</p> <p>This event occurs when a user drags an object from the Enterprise Architect Project Browser or Resources window onto a diagram. The notification is provided immediately before the object is created, so that the Add-In can disable addition of the object.</p>	<a href="#">Object</a> <sup>[3063]</sup>

**Syntax:**

**Function** *EA\_OnPreNewDiagramObject(Repository As EA.Repository, Info As EA.EventProperties) As Boolean*

The *EA\_OnPreNewDiagramObject* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Info</b>	<a href="#">EA.EventProperties</a> <sup>[2833]</sup>	IN	<p>Contains the following <i>EventProperty</i> objects for the object to be created:</p> <ul style="list-style-type: none"> <li><i>Type</i>: A string value corresponding to the <i>Type</i> of object being added to the diagram</li> <li><i>Stereotype</i>: A string value corresponding to the <i>Stereotype</i> of the object being added to the diagram</li> <li><i>ID</i>: A long value corresponding to the <i>ID</i> of the Element, Package or Diagram being added to the diagram</li> <li><i>DiagramID</i>: A long value corresponding to the ID of the diagram to which the object is being added</li> </ul>
<b>Repository</b>	<a href="#">EA.Repository</a> <sup>[2850]</sup>	IN	<p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>

**Return Value:**

Return **True** to enable addition of the object to the model. Return **False** to disable addition of the object.

**20.3.5.17.5 EA\_OnPreDropFromTree**

When a user drags any kind of element from the Project Browser onto a diagram, **EA\_OnPreDropFromTree** notifies the Add-In that a new item is about to be dropped onto a diagram. The notification is provided immediately before the element is dropped, so that the Add-In can override the default action that would be taken for this drag.

**Syntax**

**Function** *EA\_OnPreDropFromTree(Repository As EA.Repository, Info As EA.EventProperties) As Boolean*

The **EA\_OnPreDropFromTree** function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Info</b>	<a href="#">EA.EventProperties</a> <small>[2833]</small>	IN	Contains the following <b>EventProperty</b> objects for the element to be created: <ul style="list-style-type: none"> <li>• <b>ID</b>: A long value of the type being dropped</li> <li>• <b>Type</b>: A string value corresponding to type of element being dropped</li> <li>• <b>DiagramID</b>: A long value corresponding to the ID of the diagram to which the element is being added</li> <li>• <b>PositionX</b>: The X coordinate into which the element is being dropped</li> <li>• <b>PositionY</b>: The Y coordinate into which the element is being dropped</li> <li>• <b>DroppedID</b>: A long value corresponding to the ID of the element the item has been dropped onto</li> </ul>
<b>Repository</b>	<a href="#">EA.Repository</a> <small>[2850]</small>	IN	An <b>EA.Repository</b> object representing the currently open Enterprise Architect model.  Poll its members to retrieve model data and user interface status information.

#### Return Value

Returns **True** to allow the default behavior to be executed. Return **False** if you are overriding this behavior.

#### 20.3.5.17.6 *EA\_OnPreNewAttribute*

#### Topics

Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnPreNewAttribute</i> notifies Add-Ins that a new attribute is about to be created on an element. It enables Add-Ins to permit or deny creation of the new attribute.</p> <p>This event occurs when a user creates a new attribute on an element by either drag-dropping from the Project Browser, using the Attributes Properties dialog, or using the in-place editor on the diagram. The notification is provided immediately before the attribute is created, so that the Add-In can disable addition of the attribute.</p>	<a href="#">EA_OnPostNewAttribute</a> <small>[3064]</small>

#### Syntax:



**Function** *EA\_OnPreNewAttribute(Repository As EA.Repository, Info As EA.EventProperties) As Boolean*

The *EA\_OnPreNewAttribute* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Info</b>	<a href="#">EA.EventProperties</a> [2833]	IN	Contains the following <i>EventProperty</i> objects for the attribute to be created: <ul style="list-style-type: none"> <li><i>Type</i>: A string value corresponding to <i>Attribute.Type</i></li> <li><i>Stereotype</i>: A string value corresponding to <i>Attribute.Stereotype</i></li> <li><i>ParentID</i>: A long value corresponding to <i>Attribute.ParentID</i></li> <li><i>ClassifierID</i>: A long value corresponding to <i>Attribute.ClassifierID</i></li> </ul>
<b>Repository</b>	<a href="#">EA.Repository</a> [2850]	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.  Poll its members to retrieve model data and user interface status information.

#### Return Value:

Return **True** to enable addition of the new attribute to the model. Return **False** to disable addition of the new attribute.

#### 20.3.5.17.7 *EA\_OnPreNewMethod*

#### Topics

Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnPreNewMethod</i> notifies Add-Ins that a new method is about to be created on an element. It enables Add-Ins to permit or deny creation of the new method.</p> <p>This event occurs when a user creates a new method on an element by either drag-dropping from the Project Browser, using the method Properties dialog, or using the in-place editor on the diagram. The notification is provided immediately before the method is created, so that the Add-In can disable addition of the method.</p>	<a href="#">EA_OnPostNewMethod</a> [3065]

#### Syntax:

**Function** *EA\_OnPreNewMethod(Repository As EA.Repository, Info As EA.EventProperties) As Boolean*

The *EA\_OnPreNewMethod* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Info</b>	<a href="#">EA.EventProperties</a> [2833]	IN	Contains the following <i>EventProperty</i> objects for the method to be created: <ul style="list-style-type: none"> <li><i>ReturnType</i>: A string value corresponding to <i>Method.ReturnType</i></li> <li><i>Stereotype</i>: A string value corresponding to <i>Method.Stereotype</i></li> <li><i>ParentID</i>: A long value corresponding to <i>Method.ParentID</i></li> <li><i>ClassifierID</i>: A long value corresponding to <i>Method.ClassifierID</i></li> </ul>
<b>Repository</b>	<a href="#">EA.Repository</a> [2850]	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.  Poll its members to retrieve model data and user interface status information.

**Return Value:**

Return **True** to enable addition of the new method to the model. Return **False** to disable addition of the new method.

**20.3.5.17.8 EA\_OnPreNewPackage****Topics**

Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnPreNewPackage</i> notifies Add-Ins that a new package is about to be created in the model. It enables Add-Ins to permit or deny creation of the new package.</p> <p>This event occurs when a user drags a new package from the Toolbox or Resources window onto a diagram, or by selecting the <b>New Package</b> icon from the Project Browser. The notification is provided immediately before the package is created, so that the Add-In can disable addition of the package.</p>	<a href="#">EA_OnPostNewPackage</a> [3066]

**Syntax:**

**Function** *EA\_OnPreNewPackage*(*Repository As EA.Repository, Info As EA.EventProperties*) **As Boolean**

The *EA\_OnPreNewPackage* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Info</b>	<a href="#">EA.EventProperties</a> [2833]	IN	Contains the following <i>EventProperty</i> objects for the package to be created: <ul style="list-style-type: none"> <li><i>Stereotype</i>: A string value corresponding to <i>Package.Stereotype</i></li> <li><i>ParentID</i>: A long value corresponding to <i>Package.ParentID</i></li> <li><i>DiagramID</i>: A long value corresponding to the ID of the diagram to which the package is being added</li> </ul>
<b>Repository</b>	<a href="#">EA.Repository</a> [2850]	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.  Poll its members to retrieve model data and user interface status information.

**Return Value:**

Return **True** to enable addition of the new package to the model. Return **False** to disable addition of the new package.

**20.3.5.17.9 EA\_OnPreNewGlossaryTerm****Topics**

Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnPreNewGlossaryTerm</i> notifies Add-Ins that a new glossary term is about to be created. It enables Add-Ins to permit or deny creation of the new glossary term.</p> <p>The notification is provided immediately before the glossary term is created, so that the Add-In can disable addition of the element.</p>	<a href="#">EA_OnPostNewGlossaryTerm</a> [3067]

**Syntax:**

**Function** *EA\_OnPreNewGlossaryTerm*(*Repository As EA.Repository, Info As EA.EventProperties*) **As Boolean**

The *EA\_OnPreNewGlossaryTerm* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Info</b>	<a href="#">EA.EventProperties</a> [2833]	IN	Contains the following <i>EventProperty</i> object for the glossary term to be created: <ul style="list-style-type: none"> <li><i>Type</i>: A string value corresponding to <i>Term.TermID</i></li> </ul>

Parameter	Type	Direction	Description
<b>Repository</b>	<a href="#">EA.Repository</a> <small>[2850]</small>	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.  Poll its members to retrieve model data and user interface status information.

#### Return Value:

Return **True** to enable addition of the new glossary term to the model. Return **False** to disable addition of the new glossary term.

### 20.3.5.18 Tagged Value Broadcasts

Enterprise Architect includes the *Addin Broadcast* Tagged Value type that allows an Add-In to respond to attempts to edit it. The function that is called depends on the type of object the Tagged Value is on.

Tagged Value	Link
EA_OnAttributeTagEdit	<a href="#">EA_OnAttributeTagEdit</a> <small>[3084]</small>
EA_OnConnectorTagEdit	<a href="#">EA_OnConnectorTagEdit</a> <small>[3085]</small>
EA_OnElementTagEdit	<a href="#">EA_OnElementTagEdit</a> <small>[3086]</small>
EA_OnMethodTagEdit	<a href="#">EA_OnMethodTagEdit</a> <small>[3087]</small>

#### 20.3.5.18.1 EA\_OnAttributeTagEdit

#### Topics

Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnAttributeTagEdit</i> is called when the user clicks the ellipsis ( ...) button for a Tagged Value of type <i>AddinBroadcast</i> on an attribute.</p> <p>The Add-In displays fields to show and change the value and notes; this function provides the initial values for the Tagged Value notes and value, and takes on any changes on exit of the function.</p>	<p><a href="#">Predefined Structured Types</a> <small>[1622]</small></p> <p><a href="#">EA_OnConnectorTagEdit</a> <small>[3085]</small></p> <p><a href="#">EA_OnElementTagEdit</a> <small>[3086]</small></p> <p><a href="#">EA_OnMethodTagEdit</a> <small>[3087]</small></p>

#### Syntax

**Sub EA\_OnAttributeTagEdit(Repository As EA.Repository, AttributeID As Long, String TagName, String TagValue, String TagNotes)**

The *EA\_OnAttributeTagEdit* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>AttributeID</b>	Long	IN	The ID of the attribute that this Tagged Value is on.
<b>Repository</b>	<a href="#">EA.Repository</a> [2850]	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.  Poll its members to retrieve model data and user interface status information.
<b>TagName</b>	String	IN	The name of the Tagged Value to edit.
<b>TagNotes</b>	String	INOUT	The current value of the Tagged Value notes; if the value is updated, the new value is stored in the repository on exit of the function.
<b>TagValue</b>	String	INOUT	The current value of the tag; if the value is updated, the new value is stored in the repository on exit of the function.

#### 20.3.5.18.2 EA\_OnConnectorTagEdit

##### Topics

Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnConnectorTagEdit</i> is called when the user clicks the ellipsis ( ... ) button for a Tagged Value of type AddinBroadcast on a connector.</p> <p>The Add-In displays fields to show and change the value and notes; this function provides the initial values for the Tagged Value notes and value, and takes on any changes on exit of the function.</p>	<p><a href="#">Predefined Structured Types</a> [1622]</p> <p><a href="#">EA_OnAttributeTagEdit</a> [3084]</p> <p><a href="#">EA_OnElementTagEdit</a> [3086]</p> <p><a href="#">EA_OnMethodTagEdit</a> [3087]</p>

##### Syntax

**Sub EA\_OnConnectorTagEdit(Repository As EA.Repository, ConnectorID As Long, String TagName, String TagValue, String TagNotes)**

The *EA\_OnConnectorTagEdit* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>ConnectorID</b>	Long	IN	The ID of the connector that this Tagged Value is on.
<b>Repository</b>	<a href="#">EA.Repository</a> [2850]	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.  Poll its members to retrieve model data and user interface status information.
<b>TagName</b>	String	IN	The name of the Tagged Value to edit.
<b>TagNotes</b>	String	INOUT	The current value of the Tagged Value notes; if the value is updated, the new value is stored in the repository on exit of the function.
<b>TagValue</b>	String	INOUT	The current value of the tag; if the value is updated, the new value is stored in the repository on exit of the function.

### 20.3.5.18.3 EA\_OnElementTagEdit

#### Topics

Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnElementTagEdit</i> is called when the user clicks the ellipsis ( ...) button for a Tagged Value of type <i>AddinBroadcast</i> on an element.</p> <p>The Add-In displays fields to show and change the value and notes; this function provides the initial values for the Tagged Value notes and value, and takes on any changes on exit of the function.</p>	<p><a href="#">Predefined Structured Types</a> [1622]</p> <p><a href="#">EA_OnAttributeTagEdit</a> [3084]</p> <p><a href="#">EA_OnConnectorTagEdit</a> [3085]</p> <p><a href="#">EA_OnMethodTagEdit</a> [3087]</p>

#### Syntax

**Sub** EA\_OnElementTagEdit(*Repository* As *EA.Repository*, *ObjectID* As Long, *String* *TagName*, *String* *TagValue*, *String* *TagNotes*)

The *EA\_OnElementTagEdit* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>ObjectID</b>	Long	IN	The ID of the object (element) that this Tagged Value is on.

Parameter	Type	Direction	Description
<b>Repository</b>	<a href="#">EA.Repository</a> [2850]	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.  Poll its members to retrieve model data and user interface status information.
<b>TagName</b>	String	IN	The name of the Tagged Value to edit.
<b>TagNotes</b>	String	INOUT	The current value of the Tagged Value notes; if the value is updated, the new value is stored in the repository on exit of the function.
<b>TagValue</b>	String	INOUT	The current value of the tag; if the value is updated, the new value is stored in the repository on exit of the function.

#### 20.3.5.18.4 EA\_OnMethodTagEdit

##### Topics


Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnMethodTagEdit</i> is called when the user clicks the ellipsis ( ... ) button for a Tagged Value of type AddinBroadcast on an operation.</p> <p>The Add-In displays fields to show and change the value and notes; this function provides the initial values for the Tagged Value notes and value, and takes on any changes on exit of the function.</p>	<p><a href="#">Predefined Structured Types</a> [1622]</p> <p><a href="#">EA_OnAttributeTagEdit</a> [3084]</p> <p><a href="#">EA_OnConnectorTagEdit</a> [3085]</p> <p><a href="#">EA_OnElementTagEdit</a> [3086]</p>

##### Syntax

**Sub** *EA\_OnMethodTagEdit*(*Repository As EA.Repository, MethodID As Long, String TagName, String TagValue, String TagNotes*)







The *EA\_OnMethodTagEdit* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>MethodID</b>	Long	IN	The ID of the method that this Tagged Value is on.
<b>Repository</b>	<a href="#">EA.Repository</a>	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.

Parameter	Type	Direction	Description
	 [2850]		Poll its members to retrieve model data and user interface status information.
<b>TagName</b>	String	IN	The name of the Tagged Value to edit.
<b>TagNotes</b>	String	INOUT	The current value of the Tagged Value notes; if the value is updated, the new value is stored in the repository on exit of the function.
<b>TagValue</b>	String	INOUT	The current value of the tag; if the value is updated, the new value is stored in the repository on exit of the function.

### 20.3.5.19 Technology Events

Enterprise Architect Add-Ins can respond to the following events associated with the use of MDG Technologies:

Topic	Link
EA_OnInitializeTechnologies	<a href="#">EA_OnInitializeTechnologies</a>  [3089]
EA_OnPreActivateTechnology	<a href="#">EA_OnPreActivateTechnology</a>  [3089]
EA_OnPostActivateTechnology	<a href="#">EA_OnPostActivateTechnology</a>  [3090]
EA_OnPreDeleteTechnology (Deprecated)	<a href="#">EA_OnPreDeleteTechnology</a>  [3091]
EA_OnDeleteTechnology (Deprecated)	<a href="#">EA_OnDeleteTechnology</a>  [3092]
EA_OnImportTechnology (Deprecated)	<a href="#">EA_OnImportTechnology</a>  [3094]



### 20.3.5.19.1 EA\_OnInitializeTechnologies

#### Topics

Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnInitializeTechnologies</i> requests that an Add-In pass an MDG Technology to Enterprise Architect for loading.</p> <p>This event occurs on Enterprise Architect startup. Return your technology XML to this function and Enterprise Architect loads and enables it.</p>	

#### Syntax:

#### Function EA\_OnInitializeTechnologies(*Repository As EA.Repository*) As Object

The *EA\_OnInitializeTechnologies* function syntax contains the following element:

Parameter	Type	Direction	Description
<b>Repository</b>	<a href="#">EA.Repository</a> [2850]	IN	<p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>

#### Return Value:

Return the MDG Technology as a single XML string.

#### Example

```
Public Function EA_OnInitializeTechnologies(ByVal Repository As EA.
Repository) As Object
    EA_OnInitializeTechnologies = My.Resources.MyTechnology
End Function
```

### 20.3.5.19.2 EA\_OnPreActivateTechnology

#### Topics

Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnPreActivateTechnology</i> notifies Add-Ins that an MDG Technology resource is about to be activated in the model.</p> <p>This event occurs when a user selects to activate an MDG Technology resource in the model (by clicking on the <b>Set Active</b> button on the MDG Technologies dialog or by</p>	<p><a href="#">MDG Technologies</a> [1477]</p> <p><a href="#">Default Tools Toolbar</a> [135]</p> <p><a href="#">EA_OnPostActivateTechnology</a> [8090]</p>

Topic	Detail	See also
	<p>selecting the technology in the list box in the Default Tools toolbar).</p> <p>The notification is provided immediately after the user attempts to activate the MDG Technology, so that the Add-In can permit or disable activation of the Technology.</p>	

**Syntax:**

**Function** *EA\_OnPreActivateTechnology(Repository As EA.Repository, Info As EA.EventProperties) As Boolean*

The *EA\_OnPreActivateTechnology* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Info</b>	<a href="#">EA.EventProperties</a> <small>[2833]</small>	IN	<p>Contains the following <i>EventProperty</i> objects for the MDG Technology to be activated:</p> <ul style="list-style-type: none"> <li><i>TechnologyID</i>: A string value corresponding to the MDG Technology ID</li> </ul>
<b>Repository</b>	<a href="#">EA.Repository</a> <small>[2850]</small>	IN	<p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>

**Return Value:**

Return **True** to enable activation of the MDG Technology resource in the model. Return **False** to disable activation of the MDG Technology resource.

**20.3.5.19.3 EA\_OnPostActivateTechnology****Topics**



Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnPostActivateTechnology</i> notifies Add-Ins that an MDG Technology resource has been activated in the model.</p> <p>This event occurs when a user activates an MDG Technology resource in the model (by clicking on the <b>Set Active</b> button on the MDG Technologies dialog or by selecting the technology in the list box in the Default Tools toolbar).</p> <p>The notification is provided immediately after the user</p>	<p><a href="#">MDG Technologies</a> <small>[1477]</small></p> <p><a href="#">Default Tools Toolbar</a> <small>[135]</small></p> <p><a href="#">EA_OnPreActivateTechnology</a> <small>[3089]</small></p>

Topic	Detail	See also
	succeeds in activating the MDG Technology, so that the Add-In can update the Technology if necessary.	

**Syntax:**

**Function** *EA\_OnPostActivateTechnology(Repository As EA.Repository, Info As EA.EventProperties)*


The *EA\_OnPostActivateTechnology* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Info</b>	<a href="#">EA.EventProperties</a> 	IN	Contains the following <i>EventProperty</i> objects for the MDG Technology to be activated: <ul style="list-style-type: none"> <li><i>TechnologyID</i>: A string value corresponding to the MDG Technology ID</li> </ul>
<b>Repository</b>	<a href="#">EA.Repository</a> 	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.  Poll its members to retrieve model data and user interface status information.




**Return Value:**

Return **True** if the MDG Technology resource is updated during this notification. Return **False** otherwise.

**20.3.5.19.4 EA\_OnPreDeleteTechnology**

**Deprecated** - refers to deleting a technology through the Resources window; this process is no longer recommended. See [Deploy An MDG Technology](#) for information of recommended methods for using technologies.

See:

Topic	Link
EA_OnInitializeTechnologies	<a href="#">EA_OnInitializeTechnologies</a> 
EA_OnPreActivateTechnology	<a href="#">EA_OnPreActivateTechnology</a> 
EA_OnPostActivateTechnology	<a href="#">EA_OnPostActivateTechnology</a> 

Topics

Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnPreDeleteTechnology</i> notifies Add-Ins that an MDG Technology resource is about to be deleted from the model.</p> <p>This event occurs when a user deletes an MDG Technology resource from the model.</p> <p>The notification is provided immediately after the user confirms their request to delete the MDG Technology, so that the Add-In can disable deletion of the MDG Technology.</p>	<a href="#">EA_OnDeleteTechnology</a> [3092]

Syntax:

**Function** *EA\_OnPreDeleteTechnology*(*Repository As EA.Repository, Info As EA.EventProperties*) **As Boolean**

The *EA\_OnPreDeleteTechnology* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Info</b>	<a href="#">EA.EventProperties</a> [2833]	IN	<p>Contains the following <i>EventProperty</i> objects for the MDG Technology to be deleted:</p> <ul style="list-style-type: none"> <li><i>TechnologyID</i>: A string value corresponding to the MDG Technology ID</li> </ul>
<b>Repository</b>	<a href="#">EA.Repository</a> [2850]	IN	<p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>

Return Value:

Return **True** to enable deletion of the MDG Technology resource from the model. Return **False** to disable deletion of the MDG Technology resource.

**20.3.5.19.5 EA\_OnDeleteTechnology**

**Deprecated** - refers to deleting a technology through the Resources window; this process is no longer recommended. See [Deploy An MDG Technology](#) [1580] for information of recommended methods for using technologies.

See:

Topic	Link
EA_OnInitializeTechnologies	<a href="#">EA_OnInitializeTechnologies</a> <sup>[3089]</sup>
EA_OnPreActivateTechnology	<a href="#">EA_OnPreActivateTechnology</a> <sup>[3089]</sup>
EA_OnPostActivateTechnology	<a href="#">EA_OnPostActivateTechnology</a> <sup>[3090]</sup>

### Topics

Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnDeleteTechnology</i> notifies Add-Ins that an MDG Technology resource has been deleted from the model.</p> <p>This event occurs after a user has deleted an MDG Technology resource from the model. Add-Ins that require an MDG Technology resource to be loaded can catch this event to disable certain functionality.</p>	<a href="#">EA_OnPreDeleteTechnology</a> <sup>[3091]</sup>

### Syntax:

#### **Sub EA\_OnDeleteTechnology(Repository As EA.Repository, Info As EA.EventProperties)**

The *EA\_OnDeleteTechnology* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Info</b>	<a href="#">EA.EventProperties</a> <sup>[2833]</sup>	IN	Contains the following <i>EventProperty</i> objects: <ul style="list-style-type: none"> <li><i>TechnologyID</i>: A string value corresponding to the MDG Technology ID</li> </ul>
<b>Repository</b>	<a href="#">EA.Repository</a> <sup>[2850]</sup>	IN	<p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>

### Return Value:

None.

### 20.3.5.19.6 EA\_OnImportTechnology

**Deprecated** - refers to importing a technology into the Resources window; this process is no longer recommended. See [Deploy An MDG Technology](#)<sup>[1580]</sup> for information of recommended methods for using technologies.

See:

Topic	Link
EA_OnInitializeTechnologies	<a href="#">EA_OnInitializeTechnologies</a> <sup>[3089]</sup>
EA_OnPreActivateTechnology	<a href="#">EA_OnPreActivateTechnology</a> <sup>[3089]</sup>
EA_OnPostActivateTechnology	<a href="#">EA_OnPostActivateTechnology</a> <sup>[3090]</sup>

#### Topics

Topic	Detail	See also
<b>Details</b>	<p><i>EA_OnImportTechnology</i> notifies Add-Ins that you have imported an MDG Technology resource into the model.</p> <p>This event occurs after you have imported an MDG Technology resource into the model. Add-Ins that require an MDG Technology resource to be loaded can catch this Add-In to enable certain functionality.</p>	

#### Syntax:

**Sub EA\_OnImportTechnology(Repository As EA.Repository, Info As EA.EventProperties)**

The *EA\_OnImportTechnology* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Info</b>	<a href="#">EA.EventProperties</a> <sup>[2833]</sup>	IN	<p>Contains the following <i>EventProperty</i> objects:</p> <ul style="list-style-type: none"> <li><i>TechnologyID</i>: A string value corresponding to the MDG Technology ID</li> </ul>
<b>Repository</b>	<a href="#">EA.Repository</a> <sup>[2850]</sup>	IN	<p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>

**Return Value:**

None.

**20.3.6 Custom Views****Topics**

Topic	Detail	See also
<b>General Usage</b>	<p>Enterprise Architect enables custom windows to be inserted as a Diagram Tab within the Diagram View that appears at the center of the Enterprise Architect frame.</p> <p><b>Creating a custom view</b> enables you to easily display a custom interface within Enterprise Architect, along side other diagrams and built-in views for quick and easy access.</p> <p>Uses for this facility include:</p> <ul style="list-style-type: none"> <li>• Reports and graphs showing summary data of the model</li> <li>• Alternative views of a diagram</li> <li>• Alternative views of the model</li> <li>• Views of external data related to model data</li> <li>• Documentation tools</li> </ul>	<p><a href="#">Creating a custom view</a> <sup>[3095]</sup></p> <p><a href="#">Diagram Tabs</a> <sup>[790]</sup></p>

**20.3.6.1 Create a Custom View****Topics**

Topic	Detail	See also
<b>How to create a Custom View</b>	<p>A custom view must be designed as an ActiveX Custom Control and inserted via the automation interface.</p> <p>ActiveX Custom Controls can be created using most well-known programming tools including Microsoft Visual Studio. See the documentation provided by the relevant vendor on how to create a custom control to produce an OCX file.</p> <p>Once the custom control has been created and registered on the target system, it can be added through the <i>AddTab()</i> method of the <i>Repository</i> object.</p> <p>While it is possible to call <i>AddTab()</i> from any automation client, it is likely that you would call it from an Add-In, and that Add-In is defined in the same OCX that provides the custom view.</p> <p>Example C# code is shown below:</p> <pre>public class AddIn {     UserControl1 m_MyControl;</pre>	<p><a href="#">AddTab Method</a> <sup>[2855]</sup></p>

Topic	Detail	See also
	<pre>         public void EA_Connect ( EA. Repository Rep)         {         }          public object EA_GetMenuItems ( EA. Repository Repository, string Location, string MenuName)         {             if ( MenuName == " " )                 return "- &amp;C# Control Demo";             else             {                 String() ret = { " Show Custom View", " Show Button" };                 return ret;             }         }          public void EA_MenuClick ( EA. Repository Rep, string Location, string MenuName, string ItemName)         {             if ( ItemName == " Show Custom View" )                 m_MyControl = ( UserControl1 ) Rep. AddTab ( " C# Demo", " Cont Demo. UserControl1" );             else if ( ItemName == " Show Button" )                 m_MyControl.ShowButton ();         }     } </pre>	

### 20.3.7 Custom Docked Window

#### Topics

Topic	Detail	See also
<b>Abstract</b>	<p>Custom docked windows can be added into the Enterprise Architect user interface. Once added, they can be shown and docked in the same way as other built-in Enterprise Architect docked windows.</p> <p>A custom docked window must be designed as an ActiveX Custom Control and inserted via the automation interface.</p> <p>ActiveX Custom Controls can be created using most well-known programming tools including Microsoft Visual Studio. See the documentation provided by the relevant vendor on how to create a custom control to produce an OCX file.</p> <p>Once the custom control has been created and registered on the target system, it can be added using the <i>AddWindow()</i> method of the <i>Repository</i> object.</p> <p>While it is possible to call <i>AddWindow()</i> from any automation client, it is</p>	<p><a href="#">AddWindow Method</a> <small>[2855]</small></p> <p><a href="#">HideAddinWindow Method</a> <small>[2865]</small></p> <p><a href="#">RemoveWindow Method</a> <small>[2869]</small></p> <p><a href="#">ShowAddinWindow Method</a> <small>[2871]</small></p>



Topic	Detail	See also
	<p>likely that you would call it from an Add-In, and that Add-In is defined in the same OCX that provides the custom view.</p> <p>To view custom docked windows that have been added, select <b>Extensions   Add-In Windows</b>.</p> <p>Custom docked windows can also be made visible by the automation client or Add-in using the <i>ShowAddinWindow()</i> method, or hidden by using the <i>HideAddinWindow()</i> method.</p> <p>Example C# code is shown below:</p> <pre> public class Addin {     UserControl1 m_MyControl;      public void EA_Connect(EA.Repository Rep)     {         m_MyControl = (UserControl1) Rep.AddWindow(             "C# Demo", "ContDemo.UserControl1");     }      public object EA_GetMenuItems(EA.Repository     Repository, string Location, string MenuName)     {         if( MenuName == "" )             return "-&amp;C# Control Demo";         else         {             String() ret = {"Show Window", "S             return ret;         }     }      public void EA_MenuClick(EA.Repository Rep, str     string MenuName, string ItemName)     {         if( ItemName == "Show Window" )             Rep.ShowAddinWindow("C# Demo");         else if( ItemName == "Show Button" )             m_MyControl.ShowButton();     } } </pre>	

### 20.3.8 MDG Add-Ins

#### Topics

Topic	Detail	See also
<b>Abstract</b>	<p>MDG Add-Ins are specialized types of Add-Ins that have additional features and extra requirements for Add-In authors who want to contribute to Enterprise Architect's goal of Model Driven Generation.</p> <p>Unlike general Add-In events, MDG Add-In events are only sent to the</p>	<p><a href="#">MDG_Connect</a> [3100]</p> <p><a href="#">EA_Connect</a> [3022]</p> <p><a href="#">MDG_Events</a> [8098]</p>






Topic	Detail	See also
	<p>Add-In that has taken ownership of an Enterprise Architect model branch on a particular workstation.</p> <p>One of the additional responsibilities of an MDG Add-In is to take ownership of a branch of an Enterprise Architect model, which is done through the <i>MDG_Connect</i> event.</p> <p>MDG Add-Ins identify themselves as such during <i>EA_Connect</i> by returning the string <b>MDG</b>.</p> <p>Unlike ordinary Add-Ins, responding to <i>MDG Add-In</i> events is not optional, and methods must be published for each of the <i>MDG Events</i>.</p> <p>Two examples of MDG Add-Ins are the commercially available MDG Link for Eclipse and MDG Link for Visual Studio, published by <i>Sparx Systems</i>.</p>	<a href="#">Sparx Systems</a> (Online Resource)

### 20.3.8.1 MDG Events

An MDG Add-In must respond to all MDG Events. These events usually identify processes such as Build, Run, Synchronize, PreMerge and PostMerge, amongst others.


An MDG Link Add-In is expected to implement some form of forward and reverse engineering capability within Enterprise Architect, and as such requires access to a specific set of events, all to do with generation, synchronization and general processes concerned with converting models to code and code to models.

Topic	Link
MDG_BuildProject	<a href="#">MDG_BuildProject</a>  <sup>3099</sup>
MDG_Connect	<a href="#">MDG_Connect</a>  <sup>3100</sup>
MDG_Disconnect	<a href="#">MDG_Disconnect</a>  <sup>3101</sup>
MDG_GetConnectedPackages	<a href="#">MDG_GetConnectedPackages</a>  <sup>3102</sup>
MDG_GetProperty	<a href="#">MDG_GetProperty</a>  <sup>3102</sup>
MDG_Merge	<a href="#">MDG_Merge</a>  <sup>3103</sup>
MDG_NewClass	<a href="#">MDG_NewClass</a>  <sup>3106</sup>
MDG_PostGenerate	<a href="#">MDG_PostGenerate</a>  <sup>3107</sup>
MDG_PostMerge	<a href="#">MDG_PostMerge</a>  <sup>3108</sup>

Topic	Link
MDG_PreGenerate	<a href="#">MDG_PreGenerate</a> 
MDG_PreMerge	<a href="#">MDG_PreMerge</a> 
MDG_PreReverse	<a href="#">MDG_PreReverse</a> 
MDG_RunExe	<a href="#">MDG_RunExe</a> 
MDG_View	<a href="#">MDG_View</a> 

#### 20.3.8.1.1 MDGBuild Project


##### Topics

Topic	Detail	See also
<b>Details</b>	<p><i>MDG_BuildProject</i> enables the Add-In to handle file changes caused by generation. This function is called in response to a user selecting the <b>Extensions   Build Project</b> menu option.</p> <p>Respond to this event by compiling the project source files into a running application.</p>	<a href="#">MDG_RunExe</a> 

##### Syntax:

**Sub MDG\_BuildProject(Repository As EA.Repository, PackageGuid As String)**

The *MDG\_BuildProject* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>PackageGuid</b>	<i>String</i>	IN	The GUID identifying the Enterprise Architect package sub-tree that is controlled by the Add-In.
<b>Repository</b>	<a href="#">EA.Repository</a> 	IN	<p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>

**Return Value:**

None.

**20.3.8.1.2 MDGConnect**

An Add-In uses **MDG\_Connect** to handle a user driven request to connect a model branch to an external application. The function is called when the user attempts to connect a particular Enterprise Architect Package to an as yet unspecified external project. The Add-In calls the event to interact with the user to specify such a project.

The Add-In is responsible for retaining the connection details, which should be stored on a per-user or per-workstation basis. That is, users who share a common Enterprise Architect model over a network should be able to connect and disconnect to external projects independently of one another.

The Add-In should therefore not store connection details in an Enterprise Architect repository. A suitable place to store such details would be:

```
SHGet Folder Path( . . CSI DL_APPDATA. . ) \ Addi nName
```

The *PackageGuid* parameter is the same identifier as is required for most events relating to the MDG Add-In. Therefore it is recommended that the connection details be indexed using the *PackageGuid* value.

The *PackageID* parameter is provided to aid fast retrieval of Package details from Enterprise Architect, should this be required.

**Syntax**

**Function MDG\_Connect(*Repository As EA.Repository, PackageID as Long, PackageGuid As String*) As Long**

The *MDG\_Connect* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>PackageGuid</b>	<i>String</i>	IN	The unique ID identifying the project provided by the Add-In when a connection to a project branch of an Enterprise Architect model was first established.
<b>PackageID</b>	<i>Long</i>	IN	The <i>PackageID</i> of the Enterprise Architect package the user has requested to have connected to an external project.
<b>Repository</b>	<a href="#">EA.Repository</a> [2850]	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.  Poll its members to retrieve model data and user interface status information.

**Return Value**

Returns a non-zero to indicate that a connection has been made; a zero indicates that the user has not

nominated a project and connection should not proceed.

#### Learn more

- [MDG\\_Disconnect](#)<sup>[3101]</sup>

### 20.3.8.1.3 MDGDisconnect

#### Topics

Topic	Detail	See also
<b>Details</b>	<p><i>MDG_Disconnect</i> enables the Add-In to respond to user requests to disconnect the model branch from an external project.</p> <p>This function is called when the user attempts to disconnect an associated external project. The Add-In is required to delete the details of the connection.</p>	<a href="#">MDG_Connect</a> <sup>[3100]</sup>

#### Syntax:

**Function** *MDG\_Disconnect*(*Repository As EA.Repository, PackageGuid As String*) *As Long*

The *MDG\_Disconnect* function syntax contains the following elements:


Parameter	Type	Direction	Description
<b>PackageGuid</b>	<i>String</i>	IN	The GUID identifying the Enterprise Architect package sub-tree that is controlled by the Add-In.
<b>Repository</b>	<a href="#">EA.Repository</a> <sup>[2850]</sup>	IN	<p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>

#### Return Value:

Returns a non-zero to indicate that a disconnection has occurred enabling Enterprise Architect to update the user interface. A zero indicates that the user has not disconnected from an external project.

#### 20.3.8.1.4 MDGGetConnectedPackages


##### Topics

Topic	Detail	See also
<b>Details</b>	<p><i>MDG_GetConnectedPackages</i> enables the Add-In to return a list of current connection between Enterprise Architect and an external application.</p> <p>This function is called when the Add-In is first loaded, and is expected to return a list of the available connections to external projects for this Add-In.</p>	<a href="#">MDG_Connect</a> 

##### Syntax:

**Function** *MDG\_GetConnectedPackages*(*Repository As EA.Repository*) *As Variant*

The *MDG\_GetConnectedPackages* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Repository</b>	<a href="#">EA.Repository</a> 	IN	<p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>

##### Return Value:

Returns an array of GUID strings representing individual Enterprise Architect packages.

#### 20.3.8.1.5 MDGGetProperty

##### Topics

Topic	Detail	See also
<b>Details</b>	<p><i>MDG_GetProperty</i> provides miscellaneous Add-In details to Enterprise Architect.</p> <p>This function is called by Enterprise Architect to poll the Add-In for information relating to the <i>PropertyName</i>. This event should occur in as short a duration as possible as Enterprise Architect does not cache the information provided by the function.</p> <p>Values corresponding to the following <i>PropertyNames</i> must be provided:</p> <ul style="list-style-type: none"> <li><b>IconID</b> - Return the name of a DLL and a resource identifier in the format <i>#ResID</i>, where the resource ID indicates an Icon</li> </ul>	

Topic	Detail	See also
	<p><i>c:\program files\myapp\myapp.dll#101</i></p> <ul style="list-style-type: none"> <li>• <b>Language</b> - Return the default language that Classes should be assigned when they are created in Enterprise Architect</li> <li>• <b>HiddenMenus</b> - Return one or more values from the <i>MDGMenus</i> enumeration to hide menus that do not apply to your Add-In.</li> </ul> <pre>if (PropertyName == "HiddenMenus")     return mgBuildProject + mgRun;</pre>	

### Syntax

**Function** *MDG\_GetProperty(Repository As EA.Repository, PackageGuid As String, PropertyName As String) As Variant*

The *MDG\_GetProperty* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>PackageGuid</b>	<i>String</i>	IN	The GUID identifying the Enterprise Architect package sub-tree that is controlled by the Add-In.
<b>PropertyName</b>	<i>String</i>	IN	The name of the property that is used by Enterprise Architect. See <i>Details</i> for the possible values.
<b>Repository</b>	<a href="#">EA.Repository</a> <small>[2850]</small>	IN	An <i>EA.Repository</i> object representing the currently-open Enterprise Architect model.  Poll its members to retrieve model data and user interface status information.

### Return Value:

See *Details*, above.

## 20.3.8.1.6 MDGMerge

### Topics

Topic	Detail	See also
<b>Details</b>	<i>MDG_Merge</i> enables the Add-In to jointly handle changes to both the model branch and the code project that the model branch is	<a href="#">MDG_Connect</a> <small>[3100]</small>

Topic	Detail	See also
	<p>connected to.</p> <p>This event should be called whenever the user has asked to merge their model branch with its connected code project, or whenever the user has established a new connection to a code project.</p> <p>The purpose of this event is to enable the Add-In to interact with the user to perform a merge between the model branch and the connected project.</p>	<a href="#">MDG_PreMerge</a> <small>[3109]</small> <a href="#">MDG_PostMerge</a> <small>[3108]</small>

**Syntax:**

**Function** MDG\_Merge(*Repository As EA.Repository, PackageGuid As String, SynchObjects As Variant, SynchType As String, ExportObjects As Variant, ExportFiles As Variant, ImportFiles As Variant, IgnoreLocked As String, Language As String*) **As Long**

The *MDG\_Merge* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>ExportFiles</b>	<i>Variant</i>	OUT	<p>A string array containing the list of files for each model object chosen for export by the Add-In.</p> <p>Each entry in this array must have a corresponding entry in the <i>ExportObjects</i> parameter at the same array index, so <i>ExportFiles</i> (2) must contain the filename of the object by <i>ExportObjects</i>(2).</p>
<b>ExportObjects</b>	<i>Variant</i>	OUT	The string array containing the list of new model objects (in <i>Object ID</i> format) to be exported by Enterprise Architect to the code project.
<b>IgnoreLocked</b>	<i>String</i>	OUT	A value indicating whether to ignore any files locked by the code project (that is, "TRUE" or "FALSE").
<b>ImportFiles</b>	<i>Variant</i>	OUT	<p>A string array containing the list of code files made available to the code project to be newly imported to the model.</p> <p>Enterprise Architect imports each file listed in this array for import into the connected model branch.</p>
<b>Language</b>	<i>String</i>	OUT	The string value containing the name of the code language supported by the code project connected to the model branch.
<b>PackageGuid</b>	<i>String</i>	IN	The GUID identifying the Enterprise Architect package sub-tree that is controlled by the Add-In.



Parameter	Type	Direction	Description
<b>Repository</b>	<a href="#">EA.Repository</a> [2850]	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.  Poll its members to retrieve model data and user interface status information.
<b>SynchObjects</b>	<i>Variant</i>	OUT	A string array containing a list of objects ( <i>Object ID</i> format) to be jointly synchronized between the model branch and the project.  See below for the format of the Object IDs.
<b>SynchType</b>	<i>String</i>	OUT	The value determining the user-selected type of synchronization to take place.  See below for a list of valid values.

**Return Value:**

Return a non-zero if the merge operation completed successfully and a zero value when the operation has been unsuccessful.

**Merge:**

A merge consists of three major operations:

- **Export:** Where newly created model objects are exported into code and made available to the code project
- **Import:** Where newly created code objects, Classes and such things are imported into the model
- **Synchronize:** Where objects available both to the model and in code are jointly updated to reflect changes made in either the model, code project or both

**Synchronize Type:**

The *Synchronize* operation can take place in one of four different ways. Each of these ways corresponds to a value returned by *SynchType*:

- None: (*SynchType* = 0) No synchronization is to be performed
- Forward: (*SynchType* = 1) Forward synchronization, between the model branch and the code project is to occur
- Reverse: (*SynchType* = 2) Reverse synchronization, between the code project and the model branch is to occur
- Both: (*SynchType* = 3) Reverse, then Forward synchronization's are to occur

**Object ID Format:**

Each of the Object IDs listed in the string arrays described above should be composed in the following format:

```
( @namespace ) * ( #class ) * ( $attribute | %operation | : property ) *
```

### 20.3.8.1.7 MDGNewClass

#### Topics

Topic	Detail	See also
<b>Details</b>	<p><i>MDG_NewClass</i> enables the Add-In to alter details of a Class before it is created.</p> <p>This method is called when Enterprise Architect generates a new Class, and requires information relating to assigning the language and file path. The file path should be passed back as a return value and the language should be passed back via the language parameter.</p>	<a href="#">MDG_PreGenerate</a> <small>[3109]</small>

#### Syntax:

**Function** *MDG\_NewClass*(*Repository As EA.Repository, PackageGuid As String, CodeID As String, Language As String*) *As String*

The *MDG\_NewClass* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>CodeID</b>	<i>String</i>	IN	A string used to identify the code element before it is created. For more information, see <a href="#">MDG_View</a> <small>[3112]</small> .
<b>Language</b>	<i>String</i>	OUT	A string used to identify the programming language for the new Class. The language must be supported by Enterprise Architect.
<b>PackageGuid</b>	<i>String</i>	IN	The GUID identifying the Enterprise Architect package sub-tree that is controlled by the Add-In.
<b>Repository</b>	<a href="#">EA.Repository</a> <small>[2850]</small>	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.  Poll its members to retrieve model data and user interface status information.

#### Return Value:

Returns a string containing the file path that should be assigned to the Class.

### 20.3.8.1.8 MDGPostGenerate

#### Topics

Topic	Detail	See also
<b>Details</b>	<p><i>MDG_PostGenerate</i> enables the Add-In to handle file changes caused by generation.</p> <p>This event is called after Enterprise Architect has prepared text to replace the existing contents of a file. Responding to this event enables the Add-In to write to the linked application's user interface rather than modify the file directly.</p> <p>When the contents of a file are changed, Enterprise Architect passes <i>FileContents</i> as a non-empty string. New files created as a result of code generation are also sent through this mechanism, enabling Add-Ins to add new files to the linked project's file list.</p> <p>When new files are created Enterprise Architect passes <i>FileContents</i> as an empty string. When a non-zero is returned by this function, the Add-In has successfully written the contents of the file. A zero value for the return indicates to Enterprise Architect that the file must be saved.</p>	<a href="#">MDG_PreGenerate</a> <small>[3109]</small>

#### Syntax:

**Function** *MDG\_PostGenerate*(*Repository As EA.Repository, PackageGuid As String, FilePath As String, FileContents As String*) *As Long*

The *MDG\_PostGenerate* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>FileContents</b>	<i>String</i>	IN	A string containing the proposed contents of the file.
<b>FilePath</b>	<i>String</i>	IN	The path of the file Enterprise Architect intends to overwrite.
<b>PackageGuid</b>	<i>String</i>	IN	The GUID identifying the Enterprise Architect package sub-tree that is controlled by the Add-In.
<b>Repository</b>	<a href="#">EA.Repository</a> <small>[2850]</small>	IN	<p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>

#### Return Value:

The return value depends on the type of event that this function is responding to (see *Details*, above). This function is required to handle two separate and distinct cases.

### 20.3.8.1.9 MDGPostMerge

#### Topics

Topic	Detail	See also
<b>Details</b>	<p>MDG_PostMerge is called after a merge process has been completed.</p> <p>This function is called by Enterprise Architect after the merge process has been completed.</p> <p>File save checking should not be performed with this function, but should be handled by <i>MDG_PreGenerate</i>, <i>MDG_PostGenerate</i> and <i>MDG_PreReverse</i>.</p>	<a href="#">MDG_PreGenerate</a> [3109] <a href="#">MDG_PostGenerate</a> [3107] <a href="#">MDG_PreReverse</a> [3110] <a href="#">MDG_PreMerge</a> [3109] <a href="#">MDG_Merge</a> [3109]

#### Syntax:

**Function** MDG\_PostMerge(*Repository As EA.Repository, PackageGuid As String*) As Long

The *MDG\_PostMerge* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Repository</b>	<a href="#">EA.Repository</a> [2850]	IN	<p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>
<b>PackageGuid</b>	<i>String</i>	IN	The GUID identifying the Enterprise Architect package sub-tree that is controlled by the Add-In.

#### Return Value:

Return a zero value if the post-merge process has failed, a non-zero return indicates that the post-merge has been successful. Enterprise Architect assumes a non-zero return if this method is not implemented

### 20.3.8.1.10 MDGPreGenerate

#### Topics

Topic	Detail	See also
<b>Details</b>	<p><i>MDG_PreGenerate</i> enables the Add-In to deal with unsaved changes.</p> <p>This function is called immediately before Enterprise Architect attempts to generate files from the model. A possible use of this function would be to prompt the user to save unsaved source files.</p>	<a href="#">MDG_PostGenerate</a> <a href="#">e</a> <small>[3107]</small>

#### Syntax:

**Function** *MDG\_PreGenerate*(*Repository As EA.Repository, PackageGuid As String*) *As Long*

The *MDG\_PreGenerate* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Repository</b>	<a href="#">EA.Repository</a> <small>[2850]</small>	IN	<p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>
<b>PackageGuid</b>	<i>String</i>	IN	The GUID identifying the Enterprise Architect package sub-tree that is controlled by the Add-In.

#### Return Value:

Return a zero value to abort generation. Any other value enables the generation to continue.

### 20.3.8.1.11 MDGPreMerge

#### Topics

Topic	Detail	See also
<b>Details</b>	<p><i>MDG_PreMerge</i> is called after a merge process has been initiated by the user and before Enterprise Architect performs the merge process.</p> <p>This event is called after a user has performed their interactions with the merge screen and has confirmed the merge with the <b>OK</b> button, but before Enterprise Architect performs the merge process using the data provided by the <i>MDG_Merge</i> call, before</p>	<a href="#">MDG_PreGenerate</a> <small>[3109]</small> <a href="#">MDG_PostGenerate</a> <a href="#">e</a> <small>[3107]</small> <a href="#">MDG_PreReverse</a> <small>[3110]</small> <a href="#">MDG_Merge</a> <small>[3103]</small>

Topic	Detail	See also
	<p>any changes have been made to the model or the connected project.</p> <p>This event is made available to provide the Add-In with the opportunity to generally set internal Add-In flags to augment the <i>MDG_PreGenerate</i>, <i>MDG_PostGenerate</i> and <i>MDG_PreReverse</i> events.</p> <p>File save checking should not be performed with this function, but should be handled by <i>MDG_PreGenerate</i>, <i>MDG_PostGenerate</i> and <i>MDG_PreReverse</i>.</p>	<a href="#">MDG_PostMerge</a> [3108]

**Syntax:**

**Function** *MDG\_PreMerge*(*Repository As EA.Repository, PackageGuid As String*) *As Long*

The *MDG\_PreMerge* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Repository</b>	<a href="#">EA.Repository</a> [2850]	IN	<p>An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.</p> <p>Poll its members to retrieve model data and user interface status information.</p>
<b>PackageGuid</b>	<i>String</i>	IN	The GUID identifying the Enterprise Architect package sub-tree that is controlled by the Add-In.

**Return Value:**

A return value of zero indicates that the merge process can not occur. If the value is not zero the merge process proceeds.

If this method is not implemented then it is assumed that a merge process is used.

**20.3.8.1.12 MDGPreReverse****Topics**

Topic	Detail	See also
<b>Details</b>	<p><i>MDG_PreReverse</i> enables the Add-In to save file changes before being imported into Enterprise Architect.</p> <p>This function operates on a list of files that are about to be reverse-engineered into Enterprise Architect. If the user is working on unsaved versions of these files in an editor, you could either prompt the user or save automatically.</p>	<a href="#">MDG_PostGenerate</a> [3107]  <a href="#">MDG_PreGenerate</a> [3109]

Topic	Detail	See also

**Syntax:**

**Sub MDG\_PreReverse(*Repository As EA.Repository, PackageGuid As String, FilePaths As Variant*)**

The *MDG\_PreReverse* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Repository</b>	<a href="#">EA.Repository</a> [2850]	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.  Poll its members to retrieve model data and user interface status information.
<b>PackageGuid</b>	<i>String</i>	IN	The GUID identifying the Enterprise Architect package sub-tree that is controlled by the Add-In.
<b>FilePaths</b>	<i>String array</i>	IN	An array of filepaths pointed to the files that are to be reverse engineered.

**Return Value:**

None.

**20.3.8.1.13 MDGRunExe****Topics**

Topic	Detail	See also
<b>Details</b>	<i>MDG_RunExe</i> enables the Add-In to run the target application. This function is called when the user selects the <b>Extensions   Run Exe</b> menu option. Respond to this event by launching the compiled application.	<a href="#">MDG_BuildProject</a> [3099]

**Syntax:**

**Sub MDG\_RunExe(*Repository As EA.Repository, PackageGuid As String*)**

The *MDG\_RunExe* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Repository</b>	<a href="#">EA.Repository</a> 2850	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.  Poll its members to retrieve model data and user interface status information.
<b>PackageGuid</b>	<i>String</i>	IN	The GUID identifying the Enterprise Architect package sub-tree that is controlled by the Add-In.

**Return Value:**

None.

**20.3.8.1.14 MDGView****Topics**

Topic	Detail	See also
<b>Details</b>	<i>MDG_View</i> enables the Add-In to display user specified code elements.  This function is called by Enterprise Architect when the user asks to view a particular code element. This enables the Add-In to present that element in its own way, usually in a code editor.	

**Syntax:*****Function MDG\_View(Repository As EA.Repository, PackageGuid As String, CodeID as String) As Long***The *MDG\_View* function syntax contains the following elements:

Parameter	Type	Direction	Description
<b>Repository</b>	<i>EA.Repository</i>	IN	An <i>EA.Repository</i> object representing the currently open Enterprise Architect model.  Poll its members to retrieve model data and user interface status information.
<b>PackageGuid</b>	<i>String</i>	IN	The GUID identifying the Enterprise Architect package sub-tree that is controlled by the Add-In.
<b>CodeID</b>	<i>String</i>	IN	Identifies the code element in the following format:  <type>ElementPart<type>ElementPart...



Parameter	Type	Direction	Description
			<p>where each element is proceeded with a token identifying its type:</p> <ul style="list-style-type: none"><li>@ -namespace</li><li># - Class</li><li>\$ - attribute</li><li>% - operation</li></ul> <p>For example if a user has selected the <i>m_Name</i> attribute of <i>Class1</i> located in <i>namespace Name1</i>, the <i>class ID</i> would be passed through in the following format:</p> <p>@Name1#Class1%m_Name</p>

**Return Value:**

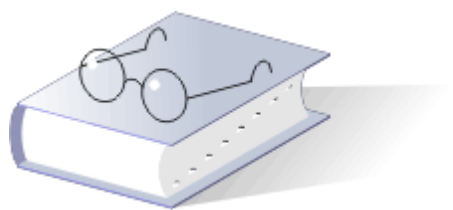
Return a non-zero value to indicate that the Add-In has processed the request. Returning a zero value results in Enterprise Architect employing the standard viewing process which is to launch the associated source file.

**Part**

---



# 21 Glossary



This glossary defines many of the terms used in Enterprise Architect.

<a href="#">A</a>	<a href="#">B</a>	<a href="#">C</a>	<a href="#">D</a>	<a href="#">E</a>	<a href="#">F</a>	<a href="#">G</a>	<a href="#">H</a>	<a href="#">I</a>	<a href="#">J</a>	<a href="#">L</a>
3116	3119	3121	3125	3128	3130	3132	3133	3134	3136	3137
<a href="#">M</a>	<a href="#">N</a>	<a href="#">O</a>	<a href="#">P</a>	<a href="#">Q</a>	<a href="#">R</a>	<a href="#">S</a>	<a href="#">T</a>	<a href="#">U</a>	<a href="#">V</a>	
3138	3141	3142	3143	3147	3148	3151	3156	3158	3160	

## 21.1 A

Name	Detail	See also
<b>Abstract Class</b>	A Class that cannot be directly instantiated.	Concrete Class
<b>Abstraction</b>	The essential characteristics of an entity that distinguish it from all other kinds of entities. An abstraction defines a boundary relative to the perspective of the viewer.	
<b>Action</b>	The specification of an executable statement that forms an abstraction of a computation procedure. An action typically results in a change in the state of the system, and can be realized by sending a message to an object or modifying a connector or a value of an attribute.	
<b>Action Sequence</b>	An expression that resolves to a sequence of actions.	
<b>Action State</b>	A state that represents the execution of an atomic action, typically the invocation of an operation.	
<b>Activation</b>	The execution of an action.	
<b>Active Class</b>	A Class whose instances are active objects. When instantiated, an active Class controls its execution. Rather than being invoked or activated by other objects, it can operate standalone, and define its own thread of behavior.	Active Object
<b>Active Object</b>	An object that owns a thread and can initiate control activity. An instance of active Class.	Active Class, Thread
<b>Activity</b>	Defines the bounds for the structural organization that contains a set of basic or fundamental behaviors. It can be used to model procedural type application development for system design through to modeling business processes in organizational structures and work flow.	
<b>Activity Diagram</b>	A diagram used to model procedural type application development for system design through to modeling business processes in organizational structures and work flow.	
<b>Activity Graph</b>	A special case of a State Machine that is used to model processes involving one or more classifiers.	State Chart Diagram

<b>Actor ( Class )</b>	A coherent set of roles that users of Use Cases play when interacting with these Use Cases. An Actor has one role for each Use Case with which it communicates.	
<b>Actual Parameter</b>	A binding for a parameter that resolves to a run-time instance.	Argument, (Formal) Parameter
<b>Aggregate ( Class )</b>	A Class that represents the 'whole' in an Aggregation (whole-part) relationship.	Aggregation
<b>Aggregation</b>	A special form of Association that specifies a whole-part relationship between the Aggregate (whole) and a component part.	Composition
<b>Analysis</b>	The part of the software development process whose primary purpose is to formulate a model of the problem domain. Analysis focuses on what to do, design focuses on how to do it.	Design
<b>Analysis Diagram</b>	A diagram used to capture high level business processes and early models of system behavior and elements. It is less formal than some other diagrams, but provides a good means of capturing the essential business characteristics and requirements.	
<b>Analysis Time</b>	Refers to something that occurs during an analysis phase of the software development process.	Modeling Time, Run Time, Compile Time, Design Time
<b>Architecture</b>	The organizational structure and associated behavior of a system. An architecture can be recursively decomposed into parts that interact through interfaces, relationships that connect parts, and constraints for assembling parts. Parts that interact through interfaces include Classes, Components and subsystems.	
<b>Argument</b>	A binding for a parameter that resolves to a run-time instance.	Actual Parameter (Formal) Parameter
<b>Artifact</b>	A physical piece of information that is used or produced by a business or development process. Examples of Artifacts include models, source files, scripts, and binary executable files. An Artifact can constitute the implementation of a deployable component.	Product, Component

<b>Assembly</b>	A connector that bridges the required interface of a component with the provided interface of a second component.	
<b>Association</b>	The semantic relationship between two or more classifiers that specifies connections among their instances.	Link
<b>Association Class</b>	A model element that has both Association and Class properties. An Association Class can be seen as an Association that also has Class properties, or as a Class that also has Association properties.	Class
<b>Association End</b>	The endpoint of an Association, which connects the Association to a classifier.	Classifier, Link End
<b>Attribute</b>	A feature within a classifier that describes a range of values that instances of the classifier can hold.	
<b>Auxiliary Class</b>	A stereotyped Class that supports another more central or fundamental Class, typically by implementing secondary logic or control flow. Auxiliary Classes are typically used together with focus Classes, and are particularly useful for specifying the secondary business logic or control flow of components during design.	Focus Class

## 21.2 B

Name	Detail	See also
<b>Behavior</b>	The observable effects of an operation or event, including its results.	
<b>Behavioral Diagram</b>	A diagram that depicts the behavioral features of a system or business process.  Behavioral diagrams include Activity diagrams, State Machine diagrams, Communication diagrams, Interaction Overview diagrams, Sequence diagrams, Timing diagrams and Use Case diagrams.	
<b>Behavioral Feature</b>	A dynamic feature of a model element, such as an operation or method.	
<b>Behavioral Model aspect</b>	A model aspect that emphasizes the behavior of the instances in a system, including their methods, collaborations, and state histories.	
<b>Binary Association</b>	An Association between two Classes; a special case of an N-ary Association.	N-ary Association
<b>Binding</b>	The creation of a model element from a template by supplying arguments for the parameters of the template.	
<b>Bookmark</b>	A marker in a Rich Text Format document that enables you to link inner sections of a document into a master document (using the Word 'Insert File' function).	
<b>Boolean</b>	An enumeration whose values are <b>true</b> and <b>false</b> .	
<b>Boolean Expression</b>	An expression that evaluates to a boolean value.	
<b>Boundary</b>	<ol style="list-style-type: none"> <li>1. A stereotyped Class that models some system boundary – typically a user interface screen; it is: <ul style="list-style-type: none"> <li>• Used in the conceptual phase to capture user interaction with the system at a screen level (or some other boundary interface type)</li> <li>• Often used in Sequence and Robustness (Analysis) diagrams</li> <li>• The View in the Model-View-Controller pattern</li> </ul> </li> <li>2. A System Boundary element used to delineate a particular</li> </ol>	

Name	Detail	See also
	part of the system.	



## 21.3 C

Name	Detail	See also
<b>C++</b>	An object-oriented programming language based on the earlier 'C' language.	
<b>Call</b>	An action state that invokes an operation on a classifier.	
<b>Cardinality</b>	The number of elements in a set.	Multiplicity
<b>CASE</b>	Computer Aided Software Engineering. A tool designed for the purpose of modeling and building software systems.	
<b>Child</b>	In a Generalization relationship, the specialization of another element, the parent.	Subclass Subtype Parent
<b>Choice</b>	A pseudo-state used to compose complex transitional paths, where the outgoing transition path is decided by dynamic, runtime conditions determined by the actions performed by the State Machine on the path leading to the choice.	
<b>Class</b>	A description of a set of objects that share the same attributes, operations, methods, relationships and semantics. A Class can use a set of interfaces to specify collections of operations it provides to its environment.	Interface Object
<b>Class Diagram</b>	A diagram that shows a collection of declarative (static) model elements, such as Classes, types, and their contents and relationships.	Object Diagram
<b>Classification</b>	The assignment of an object to a classifier.	Dynamic Classification Multiple Classification Static Classification
<b>Classifier</b>	A mechanism that describes behavioral and structural features. Classifiers include Interfaces, Classes, datatypes, and components.	
<b>Client</b>	A classifier that requests a service from another classifier.	Supplier

Name	Detail	See also
<b>Collaboration</b>	The specification of how an operation or classifier, such as a Use Case, is realized by a set of classifiers and Associations playing specific roles used in a specific way. The Collaboration defines an interaction.	Interaction
<b>Collaboration Diagram</b>	Used pre-UML 2.0. Now called a Communication diagram.	
<b>Collaboration Use</b>	Uses an occurrence to apply a pattern defined by a Collaboration to a specific situation.	
<b>Combined Fragment</b>	A combined fragment reflects a piece or pieces of interaction (called interaction operands) controlled by an interaction operator, whose corresponding boolean conditions are known as interaction constraints. It appears graphically as a transparent window, divided by horizontal dashed lines for each operand.	
<b>Comment</b>	An annotation attached to an element or a collection of elements. A comment, or note, has no semantics.	Constraint
<b>Communication Diagram</b>	A diagram that shows the interactions between elements at run-time in much the same manner as a Sequence diagram. However, Communication diagrams are used to visualize inter-object relationships, while Sequence diagrams are more effective at visualizing processing over time.	Collaboration Diagram, Object Diagram
<b>Compile Time</b>	Refers to something that occurs during the compilation of a software module.	Modeling Time, Run Time, Analysis Time, Design Time
<b>Component</b>	A modular, deployable, and replaceable part of a system that encapsulates implementation and exposes a set of interfaces. A Component is typically specified by one or more classifiers (such as implementation Classes) that reside on it, and can be implemented by one or more artifacts (such as binary, executable, or script files).	Module Artifact, Product
<b>Component Diagram</b>	A diagram that shows the organizations and dependencies among Components.	
<b>Composite ( Class )</b>	A Class that is related to one or more Classes by a Composition relationship.	Composition

Name	Detail	See also
<b>Composite State</b>	A State that consists of either concurrent (orthogonal) substates or sequential (disjoint) substates.	Substate, Concurrent Substate, Disjoint Substate
<b>Composite Structure Diagram</b>	A diagram that reflects the internal collaboration of Classes, Interfaces, or Components to describe a functionality. Composite Structure diagrams are similar to Class diagrams, except that they model a specific usage of the structure.	
<b>Composition</b>	A form of Aggregation that requires that a part instance be included in at most one Composite at a time, and that the Composite object is responsible for the creation and destruction of the parts. Composition can be recursive.	Composite Aggregation
<b>Concrete Class</b>	A Class that can be directly instantiated.	Abstract Class
<b>Concurrency</b>	The occurrence of two or more activities during the same time interval. Concurrency can be achieved by interleaving or simultaneously executing two or more threads.	Thread
<b>Concurrent Substate</b>	A substate that can be held simultaneously with other substates contained in the same composite State.	Composite State Disjoint Substate
<b>Connector</b>	A logical link between model elements. Can be structural, dynamic or possessive.	
<b>Constraint</b>	<ol style="list-style-type: none"> <li>1. A semantic condition or restriction. Certain constraints are predefined in the UML, others can be user defined. Constraints are one of three extensibility mechanisms in UML.</li> <li>2. A rule or condition that applies to some element. It is often modeled as a pre- or post- condition.</li> </ol>	Tagged Value, Stereotype Comment
<b>Container</b>	<ol style="list-style-type: none"> <li>1. An instance that exists to contain other instances, and that provides operations to access or iterate over its contents. (for example, arrays, lists, sets).</li> <li>2. A component that exists to contain other components.</li> </ol>	
<b>Containment Hierarchy</b>	A namespace hierarchy consisting of model elements, and the containment relationships that exist between them. A containment hierarchy forms a graph.	
<b>Context</b>	A view of a set of related modeling elements for a particular purpose, such as specifying an operation.	

Name	Detail	See also
<b>Continuation</b>	A Continuation is used in seq and alt combined fragments, to indicate the branches of continuation an operand follows.	
<b>Control</b>	A stereotyped Class that represents a controlling entity or manager. A Control organizes and schedules other activities and elements. It is the controller of the Model-View-Controller pattern.	
<b>Control Flow</b>	A connector linking two nodes in an activity diagram. Control Flow connectors start a node's activity when the preceding node's action is finished.	

## 21.4 D

Name	Detail	See also
<b>Database Schema</b>	The description of a database structure. It defines tables and fields and the relationship between them.	
<b>Datastore</b>	An element used to define permanently stored data. A token of data that is stored in the Datastore is stored permanently. A token of data that comes out of the Datastore is a copy of the original data. The tokens imported are kept for the life of the Activity in which it exists.	
<b>Datatype</b>	A descriptor of a set of values that lack identity and whose operations do not have side effects. Datatypes include primitive pre-defined types and user-definable types. Pre-defined types include numbers, string and time. User-definable types include enumerations.	
<b>Decision</b>	An element of an Activity diagram that indicates a point of conditional progression: if a condition is true, then processing continues one way, if not, then another.	
<b>Defining Model ( MOF )</b>	The model on which a repository is based. Any number of repositories can have the same defining model.	
<b>Delegate</b>	A connector that defines the internal assembly of a component's external ports and interfaces. Using a Delegate connector wires the internal workings of the system to the outside world, by a delegation of the external interfaces' connections.	
<b>Delegation</b>	The ability of an object to issue a message to another object in response to a message. Delegation can be used as an alternative to inheritance.  Contrast: inheritance	
<b>Dependency</b>	A relationship between two modeling elements, in which a change to one modeling element (the independent element) affects the other modeling element (the dependent element).	
<b>Deployment</b>	A type of Dependency relationship that indicates the deployment of an artifact onto a node or executable target.	
<b>Deployment Diagram</b>	A diagram that shows the configuration of run-time processing nodes and the components, processes, and objects that live	Component Diagrams

Name	Detail	See also
	on them. Components represent run-time manifestations of code units.	
<b>Deployment Specification</b>	Specifies parameters guiding deployment of an artifact, as is common with most hardware and software technologies.	
<b>Derived Element</b>	A model element that can be computed from another element, but that is shown for clarity or that is included for design purposes even though it adds no semantic information.	
<b>Design</b>	The part of the software development process whose primary purpose is to decide how the system is to be implemented. During design, strategic and tactical decisions are made to meet the required functional and quality requirements of a system.	Analysis
<b>Design Time</b>	Refers to something that occurs during a design phase of the software development process.	Modeling Time Run Time Compile Time Analysis Time
<b>Development Process</b>	A set of partially ordered steps performed for a given purpose during software development, such as constructing models or implementing models.	
<b>Diagram</b>	A graphical presentation of a collection of model elements, most often rendered as a connected graph of arcs (relationships) and vertices (other model elements). UML supports 14 diagram types, and Enterprise Architect extends these with seven more. Add-Ins, technologies and profiles can provide further diagram types.	
<b>Diagram Gate</b>	A simple graphical way to indicate the point at which messages can be transmitted into and out of Interaction Fragments.	
<b>Diagram View</b>	The Enterprise Architect workspace area where the UML diagrams are displayed.	
<b>Disjoint Substate</b>	A substate that cannot be held simultaneously with other substates contained in the same composite State.	Composite State Substate Concurrent Substate

Name	Detail	See also
<b>Distribution Unit</b>	A set of objects or components that are allocated to a process or a processor as a group. A distribution unit can be represented by a run-time composite or an Aggregate.	
<b>Domain</b>	An area of knowledge or activity characterized by a set of concepts and terminology understood by practitioners in that area.	
<b>Dynamic Classification</b>	A semantic variation of Generalization in which an object can change its classifier.	Multiple Classification Static Classification

## 21.5 E

Name	Detail	See also
<b>Element</b>	<ol style="list-style-type: none"> <li>1. An atomic constituent of a model.</li> <li>2. A model object of any type, such as Class, Component, Node or Object.</li> </ol>	
<b>Endpoint</b>	Used in Interaction diagrams to reflect a lost message in sequence.	
<b>Entity</b>	A store or persistence mechanism that captures the information or knowledge in a system. It is the Model in the Model-View-Controller pattern.	
<b>Entry Action</b>	An action executed upon entering a state in a State Machine regardless of the transition taken to reach that state.	
<b>Entry Point</b>	Used to define where external states can enter a Sub Machine.	
<b>Enumeration</b>	A list of named values used as the range of a particular attribute type. For example, RGBColor = {red, green, blue}. Boolean is a predefined enumeration with values from the set {false, true}.	
<b>Event</b>	The specification of a significant occurrence that has a location in time and space. In the context of State diagrams, an event is an occurrence that can trigger a transition.	
<b>Exception Handler</b>	An element that defines the group of operations to carry out when an exception occurs.	
<b>Exit Action</b>	An action executed upon exiting a State in a State Machine regardless of the transition taken to exit that State.	
<b>Exit Point</b>	Used in Sub Machine states and State Machines to denote the point where the machine is exited and the transition sourcing this exit point, for Sub Machines, is triggered. Exit points are a type of pseudo-state used in the State Machine diagram.	
<b>Export</b>	In the context of packages, to make an element visible outside its enclosing namespace.	Visibility



Name	Detail	See also
		Import
<b>Expose Interface</b>	A toolbox icon that is a graphical way to depict the required and supplied interfaces of a Component, Class or Part.	
<b>Expression</b>	<p>A string that evaluates to a value of a particular type. For example, the expression <math>(7 + 5 * 3)</math> evaluates to a value of type number.</p> <p>A relationship from an extension Use Case to a base Use Case, specifying how the behavior defined for the extension Use Case augments (subject to conditions specified in the extension) the behavior defined for the base Use Case. The behavior is inserted at the location defined by the extension point in the base Use Case. The base Use Case does not depend on performing the behavior of the extension Use Case.</p>	Extend Include
<b>Extend</b>	A connector used to indicate that an element extends the behavior of another. Extensions are used in Use Case models to indicate one Use Case (optionally) extends the behavior of another.	Expression Include

## 21.6 F

Name	Detail	See also
<b>Facade</b>	A stereotyped package containing only references to model elements owned by another package; it is used to provide a 'public view' of some of the contents of a package.	
<b>Feature</b>	A property, like an operation or attribute, that is encapsulated within a classifier such as an Interface, Class, or Datatype.	
<b>Final</b>	A pseudo-state that indicates an end.	
<b>Final State</b>	A special kind of State signifying that the enclosing composite State or the entire State Machine is completed.	
<b>Fire</b>	To execute a State transition.	Transition
<b>Flow Final</b>	An element that depicts an exit from the system, as opposed to the Activity Final, which represents the completion of the activity.	
<b>Focus Class</b>	A stereotyped Class that defines the core logic or control flow for one or more auxiliary Classes that support it.  Focus Classes are typically used together with one or more auxiliary Classes, and are particularly useful for specifying the core business logic or control flow of components during design.	Auxiliary Class
<b>Focus of Control</b>	A symbol on a Sequence diagram that shows the period of time during which an object is performing an action, either directly or through a subordinate procedure.	
<b>Forward Engineering</b>	The process of generating source code from the UML model.	
<b>Fork</b>	Used in State Machine diagrams as pseudo-states.  With respect to State Machine diagrams, a Fork pseudo-state signifies that its incoming transition comes from a single State, and it has multiple outgoing transitions.	Join
<b>Framework</b>	A stereotyped package containing model elements that specify a reusable architecture for all or part of a system.	Pattern

Name	Detail	See also
	<p>Frameworks typically include Classes, Patterns or templates.</p> <p>When frameworks are specialized for an application domain, they are sometimes referred to as Application frameworks.</p>	

## 21.7 G

Name	Detail	See also
<b>Generalizable Element</b>	A model element that can participate in a Generalization relationship.	Generalization
<b>Generalization</b>	A taxonomic relationship between a more general element and a more specific element. The more specific element is fully consistent with the more general element and contains additional information. An instance of the more specific element can be used where the more general element is allowed.	Generalizable Element Inheritance
<b>Guard Condition</b>	A condition that must be satisfied in order to enable an associated transition to fire.	

## 21.8 H

Name	Detail	See also
<b>History State</b>	<p>There are two types of History pseudo-states defined in UML: shallow History and deep History.</p> <p>A shallow History sub-state is used to represent the most recently active sub-state of a composite State.</p> <p>A deep History sub-state, in contrast, reflects the most recent active configuration of the composite State.</p>	

## 21.9 I

Name	Detail	See also
<b>Implementation</b>	A definition of how something is constructed or computed. For example, a Class is an implementation of a type, a method is an implementation of an operation.	
<b>Implementation Class</b>	A stereotyped Class that specifies the implementation of a Class in some programming language (for example, C++, Smalltalk, Java) in which an instance can not have more than one Class. An Implementation Class is said to realize a type if it provides all of the operations defined for the type with the same behavior as specified for the type's operations.	Type
<b>Implementation Inheritance</b>	The inheritance of the implementation of a more general element. Includes inheritance of the interface.	Interface Inheritance
<b>Import</b>	In the context of packages, a dependency that shows the packages whose Classes can be referenced within a given package (including packages recursively embedded within it).	Visibility , Export
<b>Include</b>	A relationship from a base Use Case to an inclusion Use Case, specifying how the behavior for the base Use Case contains the behavior of the inclusion Use Case. The behavior is included at the location that is defined in the base Use Case. The base Use Case depends on performing the behavior of the inclusion Use Case, but not on its structure (that is, attributes or operations).	Extend, Expression
<b>Inheritance</b>	The mechanism by which more specific elements incorporate the structure and behavior of more general elements related by behavior.	Generalization, Delegation
<b>Initial State</b>	A pseudo-state used to denote the default state of a composite State; there can be one initial vertex in each region of the composite State.	
<b>Instance</b>	An entity that has a unique identity, a set of operations that can be applied to it, and a state that stores the effects of the operations.	Object
<b>Interaction</b>	A specification of how stimuli are sent between instances to perform a specific task. The interaction is defined in the context of a collaboration.	Collaboration

Name	Detail	See also
<b>Interaction Diagram</b>	A generic term that applies to several types of diagrams that emphasize object interactions. These include Timing diagrams, Sequence diagrams, Interaction Overview diagrams and Communication diagrams.	
<b>Interaction Occurrence</b>	A reference to an existing interaction element. Interaction occurrences are visually represented by a frame, with ref in the frame's title space. The diagram name is indicated in the frame contents.	
<b>Interaction Overview Diagram</b>	A diagram that visualizes the cooperation between other Interaction diagrams to illustrate a control flow serving an encompassing purpose. As Interaction Overview diagrams are a variant of Activity diagrams, most of the diagram notation is similar, as is the process in constructing the diagram.	
<b>Interface</b>	A named set of operations that characterize the behavior of an element.	Class , Type
<b>Interface Inheritance</b>	The inheritance of the interface of a more general element. Does not include inheritance of the implementation.	Implementation Inheritance
<b>Internal Transition</b>	A transition signifying a response to an event without changing the state of an object.	
<b>Interrupt Flow</b>	An Enterprise Architect-defined toolbox icon used to define the exception handler and interruptible activity region concepts.	

## 21.10 J

Name	Detail	See also
<b>Java</b>	A fully object-oriented, cross platform language based on elements from Smalltalk, C++ and other OO languages.	
<b>Join</b>	Used in State Machine diagrams and in Activity diagrams to synchronize multiple flows.	Fork
<b>Junction</b>	Junction pseudo-states are used to design complex transitional paths. A Junction can be used to combine, or merge, multiple paths into a shared transition path or to split an incoming path into multiple paths.	



## 21.11 L

Name	Detail	See also
<b>Layer</b>	The organization of classifiers or packages at the same level of abstraction. A layer represents a horizontal slice through an architecture, whereas a partition represents a vertical slice.	Partition
<b>Lifeline</b>	An individual participant in an interaction (that is, Lifelines cannot have multiplicity). A Lifeline represents a distinct connectable element.	
<b>Link</b>	A semantic connector among a tuple of objects. An instance of an Association.	Association
<b>Link End</b>	An instance of an Association end.	Association End, Classifier
<b>Local Path</b>	A relative path on a local machine, enabling developers to store shared source code in machine specific directories, but still generate and synchronize code.	

## 21.12 M

Name	Detail	See also
<b>Maintenance</b>	The support of a software system after it is deployed.	
<b>Manifest</b>	A relationship that indicates that the artifact source embodies the target model element. Stereotypes can be added to Enterprise Architect to classify the type of manifestation of the model element.	
<b>Message</b>	Messages indicate a flow of information, or transition of control, between elements. Messages are used by Communication diagrams, Sequence diagrams, Interaction Overview diagrams and Timing diagrams.	
<b>Message Endpoint</b>	An element that defines an endpoint of a Lifeline, such as a State or Value Lifeline in a Timing diagram.	
<b>Message Label</b>	Used for messages sent between Lifelines to make the diagram appear less cluttered. Labels with the same name indicate that a message can be interrupted.	
<b>Metaclass</b>	A Class whose instances are Classes. Metaclasses are typically used to construct metamodels.	
<b>Metafile</b>	A vector-based image format native to Windows. Supports high detail and excellent scaling. Typically used for saving diagram images for placement in documents. Comes in Placeable (an older format) and Enhanced (current standard format).	
<b>Meta-Metamodel</b>	A model that defines the language for expressing a metamodel. The relationship between a meta-metamodel and a metamodel is analogous to the relationship between a metamodel and a model.	
<b>Metamodel</b>	A model that defines the language for expressing a model.	
<b>Meta-Object</b>	A generic term for all meta-entities in a meta-modeling language. For example, meta-types, meta-classes, meta-attributes, and meta-associations.	
<b>Meta-Object Facility (MOF)</b>	An Object Management Group (OMG) standard. MOF originated in the UML, when the OMG required a Meta-	

Name	Detail	See also
	Modeling architecture to define the UML. MOF is designed as a four-layered architecture.	
<b>Method</b>	The implementation of an operation. It specifies the algorithm or procedure associated with an operation.	
<b>Model ( MOF )</b>	An abstraction of a physical system with a certain purpose.Usage note: In the context of the MOF specification, which describes a meta-metamodel, the meta-metamodel is for brevity frequently referred to simply as the model.	Physical System
<b>Model Aspect</b>	A dimension of modeling that emphasizes particular qualities of the metamodel. For example, the structural model aspect emphasizes the structural qualities of the metamodel.	
<b>Model Elaboration</b>	The process of generating a repository type from a published model. Includes the generation of interfaces and implementations which enables repositories to be instantiated and populated based on, and in compliance with, the model elaborated.	
<b>Model Element ( MOF )</b>	An element that is an abstraction drawn from the system being modeled. In the MOF specification model elements are considered to be meta-objects.	View Element
<b>Model Library</b>	A stereotyped package containing model elements that are intended to be reused by other packages. A model library differs from a profile in that a model library does not extend the metamodel using stereotypes and tagged definitions. A model library is analogous to a Class library in some programming languages.	
<b>Modeling Time</b>	Refers to something that occurs during the modeling phase of the software development process. It includes analysis time and design time.Usage note: When discussing object systems, it is often important to distinguish between modeling-time and run-time concerns.	Analysis Time Design Time Compile time Run time
<b>Module</b>	A software unit of storage and manipulation. Modules include source code modules, binary code modules and executable code modules.	Component
<b>MOF</b>	Meta-Object Facility, an Object Management Group (OMG)	

Name	Detail	See also
	standard. MOF originated in the UML, when the OMG required a Meta-Modeling architecture to define the UML. MOF is designed as a four-layered architecture.	
<b>Multiple Classification</b>	A semantic variation of Generalization in which an object can belong directly to more than one classifier.	Static Classification Dynamic Classification
<b>Multiple Inheritance</b>	A semantic variation of Generalization in which a type can have more than one supertype.	Single Inheritance
<b>Multiplicity</b>	A specification of the range of enableable cardinalities that a set can assume. Multiplicity specifications can be given for roles within Associations, Parts within Composites, repetitions and other purposes. Essentially a multiplicity is a (possibly infinite) subset of the non-negative integers.	Cardinality
<b>Multi-Valued ( MOF )</b>	A model element with multiplicity defined whose Multiplicity Type:: upper attribute is set to a number greater than one. The term multi-valued does not pertain to the number of values held by, for example, an attribute or parameter, at any point in time.	Single-Valued

## 21.13 N

Name	Detail	See also
<b>Name</b>	A string used to identify a model element.	
<b>Namespace</b>	A part of the model in which the names can be defined and used. Within a namespace, each name has a unique meaning.	Name
<b>N-ary Association</b>	An Association among three or more Classes. Each instance of the Association is an n-tuple of values from the respective Classes.	Binary Association
<b>Nesting</b>	A connector used as an alternative membership notation to indicate nested members within an element; for example, a package that has nested members. The nested members of a package could also be shown inside the package rather than linked by the Nesting connector.	
<b>Node</b>	A classifier that represents a run-time computation resource, which generally has at least a memory and often processing capability. Run-time objects and components can reside on nodes.	

## 21.14 O

Name	Detail	See also
<b>Object</b>	An entity with a well-defined boundary and identity that encapsulates state and behavior. State is represented by attributes and relationships, behavior is represented by operations, methods and State Machines. An Object is an instance of a Class.	Class Instance
<b>Object Diagram</b>	A diagram that encompasses objects and their relationships at a point in time. An Object diagram can be considered as a special case of a Class diagram or Communication diagram.	Class Diagram Communication Diagram
<b>Object Flow</b>	A sub type of the State flow or transition. It implies the passing of an object instance between elements at run-time.	
<b>Object Flow State</b>	A state in an Activity graph that represents the passing of an object from the output of actions in one State to the input of actions in another State.	
<b>Object Lifeline</b>	A line in a Sequence diagram that represents the existence of an object over a period of time.	Sequence Diagram
<b>Object Management Group (OMG)</b>	The standards body responsible for the UML specification and management.	Their website is <a href="http://www.omg.org">www.omg.org</a> - follow the links to the UML pages.
<b>Occurrence</b>	A relationship that indicates that a Collaboration represents a classifier. An Occurrence connector is drawn from the collaboration to the classifier.	
<b>Operation</b>	A service that can be requested from an object to effect behavior. An operation has a signature, which could restrict the actual parameters that are possible.	

## 21.15 P

Name	Detail	See also
<b>Package</b>	<p>1. A namespace, as well as an element that can be contained in other packages' namespaces. Packages can own or merge with other packages, and their elements can be imported into a package's namespace.</p> <p>2. A logical container of model elements. It groups elements and can also contain other packages.</p> <p>The <i>OMG UML specification (UML Superstructure Specification, v2.1.1, p. 109)</i> states:</p> <p><i>A package is used to group elements, and provides a namespace for the grouped elements.</i></p> <p><i>A package is a namespace for its members, and can contain other packages. Only packageable elements can be owned members of a package. By virtue of being a namespace, a package can import either individual members of other packages, or all the members of other packages.</i></p> <p><i>In addition a package can be merged with other packages.</i></p> <p>Note that packages own model elements and are the basis for configuration control, storage and access control. Each element can be directly owned by a single package, so the package hierarchy is a strict tree. However, packages can reference other packages, modeled by using one of the stereotypes «import» and «access» of Permission dependency, so the usage network is a graph. Other kinds of dependencies between packages usually imply that one or more dependencies among the elements exist.</p> <p>A package is represented by the common folder icon - a large rectangle with a small rectangle (a 'tab') attached to the left side on top.</p>	
<b>Package Diagram</b>	Used to reflect the organization of packages and their elements, and provide a visualization of their corresponding namespaces.	
<b>Package Import</b>	A package import relationship is drawn from a source package to a package whose contents are imported. Private members of a target package cannot be imported.	
<b>Package Merge</b>	Indicates a relationship between two packages whereby the contents of the target package are merged with those of the source package. Private contents of a target package are not merged.	

Name	Detail	See also
<b>Parameter</b>	The specification of a variable that can be changed, passed, or returned. A parameter can include a name, type, and direction. Parameters are used for operations, messages and events.	Formal Parameter Argument Actual parameter
<b>Parameterized element</b>	The descriptor for a Class with one or more unbound parameters.	Template Parameterized class
<b>Parent</b>	In a generalization relationship, the generalization of another element, the child.	Subclass Subtype Child
<b>Part</b>	A run-time instance of a Class or Interface.	
<b>Participate</b>	The connection of a model element to a relationship or to a reified relationship. For example, a Class participates in an Association, an Actor participates in a Use Case.	
<b>Partition</b>	<ol style="list-style-type: none"> <li>1. activity graphs: A portion of an activity graph that organizes the responsibilities for actions.</li> <li>2. architecture: A set of related classifiers or packages at the same level of abstraction or across layers in a layered architecture. A partition represents a vertical slice through an architecture, whereas a layer represents a horizontal slice.</li> </ol>	Layer Swim Lane
<b>Pattern</b>	A template collaboration.	Framework
<b>Persistent Object</b>	An object that exists after the process or thread that created it has ceased to exist.	
<b>Physical System</b>	<ol style="list-style-type: none"> <li>1. The subject of a model.</li> <li>2. A collection of connected physical units, which can include software, hardware and people, that are organized to accomplish a specific purpose. A physical system can be described by one or more models, possibly from different viewpoints.</li> </ol>	Model (MOF) System
<b>Port</b>	Defines the interaction between a classifier and its environment. Interfaces controlling this interaction can be depicted using the 'Expose Interface' toolbox icon.	
<b>Postcondition</b>	A constraint that must be true at the completion of an	



Name	Detail	See also
	operation.	
<b>Precondition</b>	A constraint that must be true when an operation is invoked.	
<b>Primitive Type</b>	A pre-defined basic datatype without any substructure, such as an integer or a string.	
<b>Process</b>	<ol style="list-style-type: none"> <li>1. A heavyweight unit of concurrency and execution in an operating system. Thread, which includes heavyweight and lightweight processes. If necessary, an implementation distinction can be made using stereotypes.</li> <li>2. A software development process - the steps and guidelines by which to develop a system.</li> <li>3. To execute an algorithm or otherwise handle something dynamically.</li> </ol>	Thread
<b>Product</b>	A physical piece of information that is produced by a business or development process. Examples of products include models, source files, scripts, and binary executable files. An product can constitute the implementation of a deployable component.	Artifact Component
<b>Profile</b>	A stereotyped package that contains model elements that have been customized for a specific domain or purpose using extension mechanisms, such as stereotypes, tagged definitions and constraints. A profile can also specify model libraries on which it depends and the metamodel subset that it extends.	
<b>Project Browser</b>	The workspace window where the model contents are displayed in 'tree' format. Displays structures such as packages, diagrams and model elements.	
<b>Projection</b>	A mapping from a set to its subset.	
<b>Property</b>	A named value denoting a characteristic of an element. A property has semantic impact. Certain properties are predefined in the UML; others can be user defined.	Tagged Value
<b>Pseudo-State</b>	A vertex in a State Machine that has the form of a State, but doesn't behave as a State. Pseudo-states include initial and history vertices.	

Name	Detail	See also
<b>Published Model (MOF)</b>	A model that has been frozen, and that becomes available for instantiating repositories and for support in defining other models. A frozen model's model elements cannot be changed.	

## 21.16 Q

Name	Detail	See also
<b>Qualifier</b>	An Association attribute or tuple of attributes whose values partition the set of objects related to an object across an Association.	

## 21.17 R

Name	Detail	See also
<b>Realize</b>	A source object realizes the destination object.  Realize is used to express traceability and completeness in the model – a business process or requirement is realized by one or more Use Cases which are in turn realized by some Classes which in turn are realized by a Component, and so on.	
<b>Receive ( A message )</b>	The handling of a stimulus passed from a sender instance.	Sender, Receiver
<b>Receive</b>	An element used to define the acceptance or receipt of a request; movement on to the next action occurs until it has received what is defined.	
<b>Receiver ( Object )</b>	The object handling a stimulus passed from a sender object.	Sender
<b>Reception</b>	A declaration that a classifier is prepared to react to the receipt of a signal.	
<b>Recursion</b>	A type of message used in Sequence diagrams to indicate a recursive function.	
<b>Reference</b>	1. A denotation of a model element. 2. A named slot within a classifier that facilitates navigation to other classifiers.	Pointer
<b>Region</b>	UML 2.x supports both Expansion Regions and Interruptible Activity Regions. <ul style="list-style-type: none"><li>• An Expansion Region defines the bounds of a region consisting of one or more sets of input collections, where an input collection is a set of elements of the same type</li><li>• An Interruptible Activity Region contains Activity nodes - when a token leaves an interruptible region, this terminates all of the region's tokens and behaviors</li></ul>	
<b>Refinement</b>	A relationship that represents a fuller specification of something that has already been specified at a certain level of detail; for example, a design Class is a refinement of an analysis Class.	

Name	Detail	See also
<b>Relationship</b>	A semantic connection among model elements; examples of relationships include Associations and Generalizations.	
<b>Repository</b>	A facility for storing object models, interfaces and implementations.	
<b>Represents</b>	A connector that indicates a Collaboration Use is used in a classifier; the connector is drawn from the Collaboration Use to its owning classifier.	
<b>Requirement</b>	A required feature, property or behavior of a system (external requirement).	
<b>Responsibility</b>	A contract or obligation of a classifier (internal requirement).	
<b>Reuse</b>	The use of a pre-existing artifact.	
<b>Reverse Engineering</b>	The process of importing source code into the model as standard UML model objects (such as Classes, attributes and operations).	
<b>Rich Text Format</b>	A standard mark-up language for creating word processor documents, frequently associated with Microsoft Word.	
<b>Robustness Diagram</b>	Enterprise Architect supports business process modeling extensions from the UML business process model profile; Robustness diagrams are used in ICONIX.	Read more about this at <a href="http://www.sparxsystems.com/iconix/iconixsw.htm">www.sparxsystems.com/iconix/iconixsw.htm</a> .
<b>Role</b>	<ol style="list-style-type: none"> <li>1. The named detail and rules associated with one end of an association; a role can indicate name, constraints, multiplicity and collection details.</li> <li>2. The named specific behavior of an entity participating in a particular context; a role can be static (such as an Association end) or dynamic (such as a Collaboration role).</li> </ol>	
<b>Role Binding</b>	The mapping between a Collaboration Use's internal roles and the respective parts required to implement a specific situation; the associated parts can have	

Name	Detail	See also
	properties defined to enable the binding to occur, and the collaboration to take place.	
<b>Run Time</b>	The period of time during which a computer program executes.	Analysis Time, Compile Time, Design Time, Modeling Time

## 21.18 S

Name	Detail	See also
<b>Scenario</b>	<p>1. A specific sequence of actions that illustrates behaviors. A scenario can be used to illustrate an interaction or the execution of a Use Case instance.</p> <p>2. A sequence of operations carried out in some order to produce a known result. Can apply to Use Cases where it is the equivalent of a Sequence diagram, or to other objects to describe how they are used at run-time.</p>	Interaction
<b>Schema ( MOF )</b>	In the context of the MOF, analogous to a package that is a container of model elements. Schema corresponds to a MOF package.	Metamodel, Package
<b>Self-Message</b>	Reflects a new process or method invoked within the calling Lifeline's operation. It is a specification of a message.	
<b>Semantic Variation Point</b>	A point of variation in the semantics of a metamodel. It provides an intentional degree of freedom for the interpretation of the metamodel semantics.	
<b>Send (A message )</b>	The passing of a stimulus from a sender instance to a receiver instance.	Sender, Receiver
<b>Sender ( Object )</b>	The object passing a stimulus to a receiver object.	Receiver
<b>Sequence Diagram</b>	<p>A diagram that shows object interactions arranged in time sequence. In particular, it shows the objects participating in the interaction and the sequence of messages exchanged.</p> <p>Unlike a Communication (Collaboration) diagram, a Sequence diagram includes time sequences but does not include object relationships. A Sequence diagram can exist in a generic form (describes all possible scenarios) and in an instance form (describes one actual scenario).</p> <p>Sequence diagrams and Communication diagrams express similar information, but show it in different ways.</p>	Communication Diagram, Object Lifeline
<b>Signal</b>	The specification of an asynchronous stimulus communicated between instances. Signals can have parameters.	
<b>Signature</b>	The name and parameters of a behavioral feature. A signature can include an optional returned parameter.	

Name	Detail	See also
<b>Single Inheritance</b>	A semantic variation of Generalization in which a type can have only one supertype.	Multiple Inheritance
<b>Single Valued ( MOF )</b>	A model element with multiplicity defined is single valued when its Multiplicity Type: upper attribute is set to 1.  The term single-valued does not pertain to the number of values held by, for example, an attribute or parameter at any point in time, since a single-valued attribute (for instance, with a multiplicity lower bound of zero) could have no value.	Multi-Valued
<b>Specification</b>	A declarative description of what something is or does.	Implementation
<b>State</b>	A condition or situation during the life of an object during which it satisfies some condition, performs some activity, or waits for some event.	State (OMA)
<b>State Invariant</b>	A condition applied to a Lifeline that must be fulfilled for the Lifeline to exist.	
<b>State Machine</b>	A behavior that specifies the sequences of States that an object or an interaction goes through during its life in response to events, together with its responses and actions.	
<b>State Machine Diagram</b>	A diagram that illustrates how an element, often a Class, can move between States classifying its behavior, according to transition triggers, constraining guards and other aspects of State Machine diagrams that depict and explain movement and behavior.	
<b>State Chart</b>	A diagram that shows a State Machine.	State Machine, Activity Graph
<b>State Continuation</b>	A symbol that serves two different purposes for Interaction diagrams - as State Invariants and as Continuations.  A State Invariant is a condition applied to a Lifeline that must be fulfilled for the Lifeline to exist.  A Continuation is used in seq and alt combined fragments to indicate the branches of continuation that an operand follows.	
<b>State Lifeline</b>	A State Lifeline follows discrete transitions between States, which are defined along the y-axis of the timeline. Any transition has optional attributes of timing constraints, duration constraints and observations.	



Name	Detail	See also
<b>Static Classification</b>	A semantic variation of Generalization in which an object can not change classifier.	Multiple Classification, Dynamic Classification
<b>Stereotype</b>	<p>A new type of modeling element that extends the semantics of the metamodel.</p> <p>Stereotypes must be based on certain existing types or Classes in the metamodel.</p> <p>Stereotypes can extend the semantics, but not the structure of pre-existing types and Classes. Certain stereotypes are predefined in the UML, others can be user defined. Stereotypes are one of three extensibility mechanisms in UML.</p>	Constraint, Tagged value
<b>Stimulus</b>	The passing of information from one instance to another, such as raising a signal or invoking an operation. The receipt of a signal is normally considered an event.	Message
<b>String</b>	A sequence of text characters. The details of string representation depend on implementation, and can include character sets that support international characters and graphics.	
<b>Structural Diagram</b>	<p>A diagram that depicts the structural elements composing a system or function.</p> <p>These diagrams can reflect the static relationships of a structure, as do Class or Package diagrams, or run-time architectures, such as Object or Composite Structure diagrams.</p> <p>Structural diagrams include Class diagrams, Composite Structure diagrams, Component diagrams, Deployment diagrams, Object diagrams and Package diagrams.</p>	
<b>Structural Feature</b>	A static feature of a model element, such as an attribute.	
<b>Structural Model Aspect</b>	A model aspect that emphasizes the structure of the objects in a system, including their types, Classes, relationships, attributes and operations.	
<b>Subactivity State</b>	A State in an activity graph that represents the execution of a non-atomic sequence of steps that has some duration.	
<b>Subclass</b>	In a Generalization relationship, the specialization of another Class; the superclass.	Generalization, Child, Parent Superclass

Name	Detail	See also
<b>Submachine State</b>	A State in a State Machine that is equivalent to a composite State but its contents are described by another State Machine.	
<b>Subpackage</b>	A package that is contained in another package.	
<b>Substate</b>	A State that is part of a composite State.	Composite State, Concurrent Substate, Disjoint Substate
<b>Subsystem</b>	A grouping of model elements that represents a behavioral unit in a physical system.  A subsystem offers interfaces and has operations. In addition, the model elements of a subsystem can be partitioned into specification and realization elements.	Package, Physical System
<b>Subtype</b>	In a Generalization relationship, the specialization of another type; the supertype.	Generalization, Child, Parent Supertype
<b>Superclass</b>	In a Generalization relationship, the generalization of another Class; the subclass.	Generalization Subclass
<b>Supertype</b>	In a Generalization relationship, the generalization of another type; the subtype.	Generalization Subtype
<b>Supplier</b>	A classifier that provides services that can be invoked by others.	Client
<b>Swimlane</b>	A partition on an Activity diagram for organizing the responsibilities for actions. Swimlanes typically correspond to organizational units in a business model.	Partition
<b>Synch</b>	A State used for indicating that concurrent paths of a State Machine are synchronized. After bringing the paths to a synch state, the emerging transition indicates unison.	
<b>Synchronize Code</b>	The process of importing and exporting code changes to ensure the model and source code match.	
<b>System</b>	A top-level subsystem in a model.	Physical System

Name	Detail	See also
<b>System Boundary</b>	An element used to delineate a particular part of the system.	

## 21.19 T

Name	Detail	See also
<b>Table</b>	A relational table (composed of columns).	
<b>Tagged Value</b>	<p>The explicit definition of a property as a name-value pair.</p> <p>In a Tagged Value, the name is referred to as the tag. Certain tags are predefined in the UML; others can be user defined.</p> <p>Tagged Values are one of three extensibility mechanisms in UML.</p>	Constraint, Property, Stereotype
<b>Template</b>	The descriptor for a Class with one or more unbound parameters.	Parameterized Element, Parameterized Class
<b>Terminate</b>	A pseudostate indicating that upon entry of its pseudostate, the State Machine's execution ends.	
<b>Thread ( of Control )</b>	A single path of execution through a program, a dynamic model, or some other representation of control flow. Also, a stereotype for the implementation of an active object as a lightweight process.	Active Object, Process, Concurrency
<b>Time Event</b>	An event that denotes the time elapsed since the current state was entered.	Event
<b>Time Expression</b>	An expression that resolves to an absolute or relative value of time.	
<b>Timing Diagram</b>	A diagram that defines the behavior of different objects within a time-scale, with visual depictions of those objects changing state and interacting over time.	
<b>Toolbox</b>	The main toolbar running down the center of Enterprise Architect, from which you can select model elements to insert into diagrams. This is also known as the Diagram Toolbox and the Object toolbar.	
<b>Top Level</b>	<p>A stereotype of package denoting the top-most package in a containment hierarchy.</p> <p>The topLevel stereotype defines the outer limit for looking up names, as namespaces 'see' outwards. For example, <b>opTopLevelsubsystem</b> represents the top of the subsystem containment hierarchy.</p>	

Name	Detail	See also
<b>Trace</b>	A dependency that indicates a historical or process relationship between two elements that represent the same concept without specific rules for deriving one from the other.	
<b>Transient Object</b>	An object that exists only during the execution of the process or thread that created it.	
<b>Transition</b>	A relationship between two States indicating that an object in the first State performs certain specified actions and enters the second State when a specified event occurs and specified conditions are satisfied. On such a change of State, the transition is said to fire.	Fire, Object Flow
<b>Type</b>	<p>A stereotyped Class that specifies a domain of objects together with the operations applicable to the objects, without defining the physical implementation of those objects.</p> <p>A type can not contain any methods, maintain its own thread of control, or be nested. However, it can have attributes and associations.</p> <p>Although an object can have at most one implementation Class, it can conform to multiple different types.</p>	Implementation Class, Interface
<b>Type Expression</b>	An expression that evaluates to a reference to one or more types.	

## 21.20 U

Name	Detail	See also
<b>UML</b>	The Unified Modeling Language, a notation and specification for modeling software systems in an Object-Oriented manner.  You can read more about UML on the OMG home page or in our UML Tutorial.	
<b>UML Diagrams</b>	Diagrams used to model different aspects of the system under development. They include various elements and connectors, all of which have their own meanings and purposes.  UML 2.4.1 includes 14 diagrams: Use Case diagram, Activity diagram, State Machine diagram, Timing diagram, Sequence diagram, Interaction Overview diagram, Communication diagram, Package diagram, Class diagram, Profile diagram, Object diagram, Composite Structure diagram, Component diagram and Deployment diagram.	
<b>UML Toolbox</b>	The main toolbar running down the center of Enterprise Architect from which you can select model elements to insert into diagrams. This is also known as the Diagram Toolbox and the Object toolbar.	
<b>Uninterpreted</b>	A placeholder for a type or types whose implementation is not specified by the UML. Every uninterpreted value has a corresponding string representation.	Any ( CORBA )
<b>Usage</b>	A dependency in which one element (the client) requires the presence of another element (the supplier) for its correct functioning or implementation.	
<b>Use</b>	A connector that indicates that one element requires another to perform some interaction. The Usage relationship does not specify how the target supplier is used, other than that the source client uses it in definition or implementation.	
<b>Use Case ( Class )</b>	A UML model element that describes how a user of the proposed system interacts with the system to perform a discrete unit of work. It describes and signifies a single interaction over time that has meaning for the end user (person, machine or other system), and is required to leave the system in a complete state: either the interaction completed or was rolled back to the initial state.	Use Case Instance
<b>Use Case Diagram</b>	A diagram that captures Use Cases and Actor interactions. It describes the functional requirements of the system, the manner in which outside things (Actors) interact at the system boundary, and the response of the system.	

Name	Detail	See also
<b>Use Case Estimation</b>	The technique of estimating project size and complexity based on the number of Use Cases and their difficulty.	
<b>Use Case Instance</b>	The performance of a sequence of actions being specified in a Use Case. An instance of a Use Case.	Use Case Class
<b>Use Case Model</b>	A model that describes a system's functional requirements in terms of Use Cases.	
<b>Utility</b>	<p>A stereotype that groups global variables and procedures in the form of a Class declaration. The utility attributes and operations become global variables and global procedures, respectively.</p> <p>A utility is not a fundamental modeling construct, but a programming convenience.</p>	

## 21.21 V

Name	Detail	See also
<b>Value</b>	An element of a type domain.	
<b>Value Lifeline</b>	A Lifeline that shows the Lifeline's state across the diagram, within parallel lines indicating a steady state.  A cross between the lines indicates a transition or change in state.	
<b>Vertex</b>	A source or a target for a transition in a State Machine; a vertex can be either a State or a pseudo-state.	State, Pseudo-State
<b>View</b>	A projection of a model, which is seen from a given perspective or vantage point and omits entities that are not relevant to this perspective.	
<b>View Element</b>	An element that is a textual and/or graphical projection of a collection of model elements.	Model Element (MOF)
<b>View Projection</b>	A projection of model elements onto view elements.  A View Projection provides a location and a style for each view element.	
<b>Visibility</b>	An enumeration whose value (public, protected, package or private) denotes how the model element to which it refers can be seen outside its enclosing namespace.	Export, Import
<b>Visual Basic</b>	A rapid application development programming language.  Windows' only scripting language based on COM.	



**Part**

---



## 22 License Management



The License Management dialog in Enterprise Architect enables you to upgrade Enterprise Architect and to register Add-Ins. The dialog lists:



- The currently-registered keys
- The product each key applies to
- When each private key expires
- When each shared key is to be reactivated in the keystore for issue to another user

**Access**   **Help | Register and Manage License Key(s)**

### License Management options

Option	Action	See also
<b>Add Key</b>	Display the Add Registration Key dialog, which you can use to: <ul style="list-style-type: none"> <li>• Register your new Enterprise Architect license</li> <li>• Obtain a shared key from the Sparx Systems Key Store (available for version 4.51 and above)</li> <li>• Add a new key to update to a higher version of Enterprise Architect or to register an Add-In</li> </ul>	<a href="#">Register a Full License</a> <sup>[36]</sup> <a href="#">Add License Keys</a> <sup>[3165]</sup> <a href="#">Upgrade an Existing License</a> <sup>[3169]</sup> <a href="#">Register Add-In</a> <sup>[317]</sup>
<b>Remove Key</b>  <b>Release Key</b>	<ul style="list-style-type: none"> <li>• (Private Key) Make the Add-In or current version of Enterprise Architect inoperable</li> <li>• (Shared Key) Release the key to the keystore; however, the Keystore Manager is normally configured to release keys automatically when the user logs off</li> </ul>	
<b>Copy</b>	Place the highlighted key into the clipboard.	
<b>Close</b>	Close the dialog.	
<b>Help</b>	Display the help for this topic.	

[Learn more](#)

- [Finding Your License Information](#) 
- [Keystore Troubleshooting](#) 

## 22.1 *Finding Your License Information*

You can find information on your Enterprise Architect license in the About Enterprise Architect dialog.

[Access](#) [Help](#) | [About EA](#)

## 22.2 Adding License Keys

Two types of license key can be used in conjunction with Enterprise Architect: Private and Shared.

### Reference

Topic	Detail	See also
<b>Private Keys</b>	<p>Private keys enable you to register, indefinitely, on the machine and user account that you are currently using:</p> <ul style="list-style-type: none"> <li>• An Enterprise Architect license for the edition of Enterprise Architect you have purchased, or</li> <li>• An Add-In product license (such as for MDG Link for Eclipse or MDG Link for Visual Studio.NET).</li> </ul>	<a href="#">Enterprise Architect Editions</a> <sup>18</sup>
<b>Shared Keys</b>	<p>Shared keys are obtained from a central shared key store, to give you temporary access to the installed 'suite' edition of Enterprise Architect.</p> <p>Shared Keys are available with the purchase of a floating license, and require Enterprise Architect version 4.51 or higher.</p> <p>Shared Keys require a shared license key store to be configured by your license administrator; the key store can be either file-based or network based (preferred).</p> <p>Only the Key Administrator is required to install the Sparx Enterprise Key Store application; end users simply connect to the configured key file - advised by the administrator - as described below.</p> <p>No additional software is required to be installed.</p>	<a href="#">Enterprise Architect Floating License</a> .
<b>Notes on Keys</b>	<p>Some license keys can override and remove others</p> <ul style="list-style-type: none"> <li>• The key for a more advanced edition of Enterprise Architect replaces the key for a simpler edition</li> <li>• The key for MDG Integration for Visual Studio replaces the key for MDG Link for Visual Studio</li> </ul> <p>Shared keys and Private keys have different formats and cannot be used interchangeably.</p>	

**Access** [Help | Register and Manage License Key\(s\) > License Management: Add Key > Add Registration Key](#)

### How to

To add a private key

Step	Action	See also
1	Click on the Enter Private Key tab.	<a href="#">License Management</a> 3162
2	In the <b>Name</b> and <b>Company</b> fields, type your user name and company name.	
3	In the <b>Copy registration key into space below...</b> field, copy the registration key (including any parentheses around the key).	
4	Click on the <b>OK</b> button to confirm the key selection.	

To add a shared key

Step	Action	See also
1	Click on the Get Shared Key tab.	<a href="#">License Management</a> 3162
2	In the <b>Name</b> and <b>Company</b> fields, type your user name and company name.	
3	In the <b>Shared Keystore</b> field, click on the ( ... ) (Browse) button. The Shared Keystore Selection dialog displays.	
4	If your keystore is file-based: <ul style="list-style-type: none"> <li>• Select the <b>File Based Keystore</b> radio button</li> <li>• Click on the <b>Browse</b> button, and</li> <li>• Locate and select the keystore file</li> </ul> Go to step 7.	
5	If the keystore is network-based: <ul style="list-style-type: none"> <li>• Click on the <b>Sparx Keystore Server</b> radio button</li> <li>• In the <b>Server Address</b> field, type the server address of the keystore.</li> </ul>	
6	If necessary, type in the password (advised by your administrator) and/or click on the <b>Test</b> button to ensure that you have a connection to the keystore.	

Step	Action	See also
7	Click on the <b>OK</b> button to return to the Get Shared Key tab. This now shows the name of the keystore in the <b>Shared Keystore</b> field.	
8	In the <b>Select a Product</b> field, click on the appropriate product name.	
9	Click on the <b>OK</b> button. The License Management dialog redisplay, indicating that the shared key is registered for the selected product until the key expiry date.	
10	Click on the <b>Close</b> button.	

## 22.3 Keystore Troubleshooting

Message Displayed:	Explanation
<i>Error reading Key Store file: (Access is denied)</i>	<p>All users who are to use the shared key facility require <b>Read, Write and Modify</b> access to the <code>sskeys.dat</code> file containing the shared keys.</p> <p>Please verify that all required users have sufficient permissions to the file and try again.</p> <p>Review the effective permissions calculated at the location of the key file for the user account reporting the problem - you should closely examine the permissions for both the Network Share and the File System; it is possible that these permissions have been overwritten at some point.</p> <p>If the problem continues, contact Sparx Support.</p>
<i>Error reading Key Store file: (Key File has been moved)</i>	<p>In the key store, as a security measure the hard drive serial number is recorded when the file is created; the file then cannot be moved from the original location in which it was created.</p> <p>If the key store has to be re-located for any reason, the administrator should re-create the key store in the new location using the original license keys.</p> <p>This issue is commonly seen after a file server has undergone a hardware upgrade in which the physical hard drives have been replaced; problems could also occur if the drive used is part of a RAID configuration.</p> <p>The message can also appear where the key store exists on a Novell-based file system - when creating the key store, the administrator is prompted to confirm that the key store is to be located on a Novell Netware file server; if they click on the <b>Yes</b> button, the key store instead records the logical path used to create it, and all users must connect to the key store using this same path.</p> <p>The recorded path is case-sensitive and must be an exact match.</p>



## 22.4 Upgrade an Existing License

This topic explains how to upgrade your installation of Enterprise Architect to a more powerful edition.

Topic	Detail	See also
<b>Editions</b>	<p>There are six Enterprise Architect editions:</p> <ul style="list-style-type: none"> <li>• Desktop</li> <li>• Professional</li> <li>• Corporate</li> <li>• Business and Software Engineering</li> <li>• Systems Engineering</li> <li>• Ultimate</li> </ul> <p>The installation file provides the facilities of all editions; your registration key gives access to the facilities of the edition you have purchased.</p>	<a href="#">Editions Available</a> <sup>[20]</sup>
<b>Upgrade</b>	<p>If you have purchased a less powerful edition, such as Professional or Desktop, you can subsequently upgrade your existing license to a more powerful edition by purchasing a special upgrade key from Sparx Systems.</p> <p>Once you have purchased and received the appropriate key, use the procedure below to gain access to the additional features.</p>	<a href="#">Enterprise Architect Purchase Details</a>
<b>Trial and Lite Editions</b>	<p>The Lite edition and the Trial version cannot be registered or upgraded.</p> <p>If you have used either of these versions and then purchased Enterprise Architect, you must download the registered version from:</p> <p style="text-align: center;"><a href="http://www.sparxsystems.com/securedownloads/easetupfull.exe">www.sparxsystems.com/securedownloads/easetupfull.exe</a></p> <p>Then you can enter your registration key.</p>	<a href="#">The Trial Version</a> <sup>[18]</sup> <a href="#">The Read-only 'Lite' Edition</a> <sup>[23]</sup> <a href="#">Registered User software download site</a>
<b>Application</b>	<p>This topic is mainly applicable to users with private keys.</p> <p>If you are an end-user with a shared key, you would simply be allocated the relevant key next time you requested one.</p> <p>If, however, you want to upgrade while using a shared key on a long lease, you would simply click on the <b>Release Key</b> button and then the <b>Add Key</b> button.</p>	<a href="#">Adding License Keys</a> <sup>[3165]</sup>

[Access](#)   **Help | Register and Manage License Key(s) > License Management: Add Key**

### How to

To upgrade from one license edition to another

Step	Action	See also
1	Make sure you have a valid upgrade key purchased from Sparx Systems; you typically receive this in an email or PDF format.	
2	On the Add Registration Key dialog, in the <b>Name</b> and <b>Company</b> fields, type your name and company name.	<a href="#">License Management</a> [3162]
3	Copy the key you received for the upgraded edition of Enterprise Architect - including the { and } bracket characters - from the email.	
4	Paste the key into the <b>Copy registration key</b> field.	
5	Click on the <b>OK</b> button.  Enterprise Architect displays the message:  <i>Registration succeeded – Thank you for purchasing Enterprise Architect xxxx Edition</i>	
6	Click on the <b>OK</b> button, and then on the <b>Close</b> button to continue working in Enterprise Architect.	
7	Select the <b>Help   About EA</b> menu option.  Copy the registration key shown and store it somewhere safe; this is a key to the full license of the edition you have upgraded to.  If you ever have to reinstall Enterprise Architect, you can register it with this key, so you won't have to go through the upgrade process again.	

## 22.5 Register Add-In

Apart from registering Enterprise Architect, you must also register any Add-Ins you purchase to use with the application. You can register either a private key as explained here, or a shared Add-In key as explained in *Add An Add-In Key*.

**Access**   [Help | Register and Manage License Key\(s\)](#)

### Register Add-Ins for Enterprise Architect

Step	Action	See also
1	Purchase one or more licenses for the Add-In from your Add-In provider.  Once you have paid for a licensed version of the Add-In, you receive (via email or other suitable means) a license key for the product.	
2	Save the license key and the latest full version of the Add-In.	
3	Run the Add-In's setup program to install the Add-In.	
4	In Enterprise Architect, select the menu option: <ul style="list-style-type: none"> <li><b>Register and Manage License Key(s)</b></li> </ul> The License Management dialog displays.	<a href="#">License Management</a> t <sub>[3162]</sub>
5	Click on the <b>Add Key</b> button.  The Add Registration Key dialog displays, showing the Enter Private Key tab.	
6	Type or copy-and-paste in the key you received with the Add-In, including the { and } characters.	
7	Click on the <b>OK</b> button.	
8	When the Add-In has been added successfully, close down Enterprise Architect and restart it to apply the integration changes imposed by the Add-In.	

### Learn more

- [Add an Add-In Key](#)<sub>[3172]</sub>
- [Add-In License Management Events](#)<sub>[3037]</sub>

## 22.6 Add an Add-In Key

When your organization is using a keystore to distribute keys you can retrieve a key for a third party add-in in the same way as you would for a key for any Sparx Systems product.

**Access**   [Help](#) | [Register and Manage License Key\(s\): Add Key > Get Shared Key](#)

### Add a shared Add-In key

Step	Action	See also
1	In the <b>Name</b> and <b>Company</b> fields, type your user name and company name.	
2	In the <b>Shared Keystore</b> field, click on the <b>Browse</b> button ( ... ). The Shared Keystore Selection dialog displays.	
3	Click on the <b>Sparx Keystore Server</b> radio button, and in the <b>Server Address</b> field type the server address of the keystore.	
4	If necessary, in the <b>Password</b> field type the password (advised by your administrator) and/or click on the <b>Test</b> button to check the connection to the keystore.	
5	Click on the <b>OK</b> button to return to the Get Shared Keystore tab, which now shows the name of the keystore in the <b>Shared Keystore</b> field.	
6	In the <b>Select a Product</b> field, look for the appropriate product name, and click on it.	
7	Click on the <b>OK</b> button. The License Management dialog redisplay with the shared Add-In key in the <b>Currently Registered Keys</b> list.	
8	Click on the <b>Close</b> button.	

### Notes

- Shared Add-In keys require Enterprise Architect version 11 or higher

### Learn more

- [License Management](#) 

- [Register Add-In](#) 

# Index

■ ■ ■

.EAP  
    Create Project File 210  
.EMX  
    Import 478  
.FEAP  
    Create Project File 210  
.NET  
    ASP, Debug 2203  
    Debug 2200  
    Debug Another Process 2243  
    Debug CLR Versions 2201  
    Debug With COM Interop Process 2202  
    Debug, System Requirements 2186  
    Garbage Collect (Exit Method) 2850  
    Object Workbench, Set Up 2216  
    Set Up Debug Session 2200  
.UML2  
    Import 478

## - A -

Abstract  
    Complex Models 1376  
Abstract Class Model  
    Generate XML Schema From 2413  
Abstraction  
    Connector 1391  
    Relationship 1391  
Acceptance Testing 2607  
AcceptEvent Action  
    Simulation 1268  
    Triggers Tab 1268  
Access  
    Email 170  
    Internet Search Engine 170  
    MDG Technologies Remote From Enterprise Architect 1479  
    Web Site 170  
Access 2007  
    Convert EAP to ACCDB Format 241  
    Transfer EAP To ACCDB Repository 241  
Acknowledgement  
    CXImage Library 32  
    Of Contributions 32  
    Of Trademarks 29

Print Listview 32

### Action

Accept Event, Simulation 1268  
AcceptEvent, Triggers Tab 1268  
As Hyperlink 2004  
BroadcastSignal, Signal Tab 1268  
Call 2134  
DestroyObject Action 2483  
Element 1266  
Expansion Node 1301  
In Simulation 1268, 2497  
Invocation 2134  
Invocation, Simulation 1268  
Local Pre/Post Conditions 1274  
Notation 1268  
Operations 1266, 1275  
ReadSelf 1268  
SendSignal, Signal Tab 1268  
StructuralFeature 1268  
Trigger, AcceptEvent 1268  
Type, Set 1268  
Update Operation 1266, 1275  
Variable 1273

### Action Pin

Add To Action 1277  
As Action Property 1278  
As Argument For Call Action 1277  
Assign To Action 1278  
Properties 1277

### ActionScript

Code Generation Language Options 2264  
Import, Reverse Engineering 2139  
Modeling Conventions 2081  
Options 2264  
Versions Supported 2264

### Activate

MDG Technologies 1477

### Activate Recording Markers 2224

### Activation

End 1255  
Extend Down 1255  
Extend Up 1255  
Lower 1255  
Raise 1255  
Sequence Element 1255  
Suppress 1255

### Activation Layer

Sequence Diagram Lifelines 1256

### Activation Levels

Sequence Diagram Lifeline Self Messages 1256

### Active

- Active
  - Classes 1365
  - State Configuration 1330
- Active Directory
  - Import User ID From 324
- Active State Logic
  - Model State Machine For HDL 2131
- Activity
  - Behavioral Aspects 1026
  - Default Object Variable, In Simulation 2479
  - Edit Parameters 1026
  - Element 1279
  - Elements And Connectors 808
  - Expansion Node 1301
  - Generate Code From Decision Table 1935
  - Instance 1009
  - Notation 1281
  - Parameter Node 1281
  - Parameter, In Return Simulation 2501
  - Parameters, Define 1028
  - Partition 1283, 1325
  - Process Element 2008
  - Region Element 1328
  - Simulate Behavior 2489
  - Simulate Return 2501
  - Structured 1340
  - Structured, Conditional Node 1338, 1345
  - Structured, Loop Node 1338, 1341
  - Structured, Sequential 1338
  - Toolbox Pages 808
- Activity (BPEL 1.1)
  - Create 1881
  - Loop 1881
  - Model 1881
  - Sub-Process 1881
  - Task 1881
  - Types 1881
- Activity (BPEL 2.0)
  - Create 1904
  - Loop 1904
  - Model 1904
  - Sub-Process 1904
  - Task 1904
  - Types 1904
- Activity Diagram
  - Create Object From Attribute 1008
  - Description 1199
  - Elements And Connectors 1199
  - Example 1200
  - Generate Code From 2121, 2134
  - Generate From Scenario 976, 978
  - Generate Scenario From 985
  - Object Flows 1436
  - Operations 1266, 1275
- Activity Edge
  - Connector 1415
  - Relationship 1415
- Activity Final
  - Element 1305
- Activity Parameter
  - In Activity Return Simulation 2501
- Activity Parameter Node
  - Exception 1281
  - Stream 1281
- Activity Partition
  - Docking 1325
  - Element 1325
  - Horizontal 1325
  - Vertical 1325
- Activity Return
  - Simulation 2501
- Actor
  - Element 1284
- Ada 2005
  - Code Generation Language Options 2264
  - Modeling Conventions 2082
  - Options 2264
- Adaptive Server Anywhere
  - Data Repository, Connect To 239
  - ODBC Driver, Set Up 237
  - Repository, Create 237
- Add
  - Category To Team Review 350
  - Code Modules In MDG Technology Wizard 1552
  - Connectors Between Locked Elements 388
  - Connectors To UML Model, Quick Start 55
  - Custom Compartments 1604
  - Diagram Properties Note 848
  - Diagram To Package 50
  - Diagram To Project 822
  - Diagram To UML Model, Quick Start 50
  - Diagram Type In MDG Technology Wizard 1550
  - Document Report Template In MDG Technology Wizard 1555
  - Document To Team Review 353
  - Element Changes 2625
  - Element Defects 2625
  - Element Directly To Package 903
  - Element In Specification Manager 1738
  - Element Issues 2625
  - Element Tasks 2625
  - Element To Diagram 52
  - Element To UML Model, Quick Start 52

- Add
  - Elements To Diagram From Project Browser 833
  - Enumeration Tags To Stereotypes 1493
  - Filters To Search 716
  - Images In MDG Technology Wizard 1556
  - Instance Variable 1380
  - Key 3165
  - License Key 3165
  - Line Points 1114
  - Linked Document Template In MDG Technology Wizard 1555
  - Linked Note To Custom Compartment 1604
  - MDA Transforms In MDG Technology Wizard 1554
  - Model Searches In MDG Technology Wizard 1559
  - Model To Project 199
  - Model Views In MDG Technology Wizard 1558
  - New Code Sections To Existing Features 1640
  - Note To Connector 1111
  - Note To Link 1111
  - Package In Project Browser 772
  - Package To UML Model, Quick Start 50
  - Packages To Model Document 2675
  - Pattern In MDG Technology Wizard 1549
  - Pattern To Diagram 1467, 1469
  - Port To Element 1385
  - Profile Attribute To Diagram 1472
  - Profile Connector To Diagram 1472
  - Profile Element To Diagram 1472
  - Profile In MDG Technology Wizard 1548
  - Profile Operation To Diagram 1472
  - Project Items Via Toolbar 136
  - Project Task 527
  - Property Value to Part 1383
  - Scripts In MDG Technology Wizard 1556
  - Shape Script To Stereotype In Profile 1501
  - Tagged Value Types In MDG Technology Wizard 1551
  - Tagged Values 1136
  - Test Details 2607
  - Toolbox In MDG Technology Wizard 1551
  - Topic To Team Review 351
  - UML Diagram 822
  - UML Pattern To Diagram 1467, 1469
  - Views 770
  - Workspace Layouts In MDG Technology Wizard 1557
- Add And Delete Attributes
  - Automation Interface Code Example 3000
- Add And Delete Methods
  - Automation Interface Code Example 3000
- Add And Manage Diagrams
  - Automation Interface Code Example 2999
- Add And Manage Elements
  - Automation Interface Code Example 2997
- Add And Manage Packages
  - Automation Interface Code Example 2996
- Add Connector
  - Automation Interface Code Example 2998
- Add Diagram
  - Package Context Menu, Project Browser 651
- Add Element
  - Package Context Menu, Project Browser 651
- Add Model
  - Package Context Menu, Project Browser 651
- Add Package
  - Package Context Menu, Project Browser 651
- Add Stereotypes
  - Automation Interface Code Example 3006
- Add Submenu
  - Element Context Menu, Project Browser 662
- Add to Project Clipboard
  - Menu Option (Edit Menu) 80
- AddDocumentationPath
  - Loading Learning Center Documentation 74
- Add-In
  - Add Custom Window 3096
  - Add Shared Key 3172
  - And Enterprise ArchitectDeadlocks (.NET) 3016
  - COM Interoperability 3016
  - Concurrent Method Calls 3016
  - Connect To 117
  - Create 3012
  - Create Learning Center Pages To Support 74
  - Create, Define Menu Items 3013
  - Deploy 3014
  - Disable 3018
  - Display Help On 117
  - Enable 3018
  - Events 3021
  - Holding State Information 3016
  - Manage 3018
  - Manager 3018
  - MDG 1481
  - Menu 117
  - Model Driven Generation 1481
  - Pre-2004 3016
  - Re-entrancy 3016
  - Register 3171
  - Search 711, 3019
  - Search Data 3020



- Add-In
  - Submenu 117
  - Tasks 3011
  - Visual Basic Issues 3016
- Add-In Event
  - EA\_Connect 3022
  - EA\_Disconnect 3023
  - EA\_GetMenuItems 3023
  - EA\_GetmenuState 3024
  - EA\_MenuClick 3025
  - EA\_OnOutputItemClicked 3027
  - EA\_OnOutputItemDoubleClicked 3028
  - EA\_ShowHelp 3029
- Add-In License Management Event
  - EA\_AddinLicenseGetDescription 3032
  - EA\_AddinLicenseValidate 3031
  - EA\_GetSharedAddinName 3033
- Add-In License Management Events
  - Add-In Model 3031
- Add-In Model
  - Add-In Event, EA\_Connect 3022
  - Add-In Event, EA\_Disconnect 3023
  - Add-In Event, EA\_GetMenuItems 3023
  - Add-In Event, EA\_GetMenuState 3024
  - Add-In Event, EA\_MenuClick 3025
  - Add-In Event, EA\_OnOutputItemClicked 3027
  - Add-In Event, EA\_OnOutputItemDoubleClicked 3028
  - Add-In Event, EA\_ShowHelp 3029
  - Add-In Event, Overview 3021
  - Add-In License Management Event, EA\_AddinLicenseGetDescription 3032
  - Add-In License Management Event, EA\_AddinLicenseValidate 3031
  - Add-In License Management Event, EA\_GetSharedAddinName 3033
  - Add-In License Management Events 3031
  - Add-In Tasks 3011
  - Benefits 3010
  - Broadcast Event, EA\_FileClose 3040
  - Broadcast Event, EA\_FileNew 3041
  - Broadcast Event, EA\_FileOpen 3042
  - Broadcast Event, EA\_OnPostCloseDiagram 3042
  - Broadcast Event, EA\_OnPostInitialized 3043
  - Broadcast Event, EA\_OnPostOpenDiagram 3044
  - Broadcast Event, EA\_OnPostTransform 3044
  - Broadcast Event, EA\_OnPreExitInstance 3045
  - Broadcast Event, EA\_OnRetrieveModelTemplate 3046
  - Broadcast Event, EA\_OnTabChanged 3047
  - Broadcast Events 3029
  - Compartment Events 3034
  - Compartment Events, EA\_GetCompartmentData 3035
  - Compartment Events, EA\_QueryAvailableCompartments 3034
  - Context Item Events 3037
  - Context Item Events, EA\_OnContextItemChanged 3037, 3039
  - Context Item Events, EA\_OnContextItemDoubleClicked 3038
  - Create Add-In 3012
  - Create Add-In, Tricks and Traps 3016
  - Create Custom View 3095
  - Custom View 3095
  - EA\_AddinLicenseGetDescription 3032
  - EA\_AddinLicenseValidate 3031
  - EA\_Connect 3022
  - EA\_Disconnect 3023
  - EA\_FileClose 3040
  - EA\_FileNew 3041
  - EA\_FileOpen 3042
  - EA\_GetCompartmentData 3035
  - EA\_GetMenuItems 3023
  - EA\_GetMenuState 3024
  - EA\_GetSharedAddinName 3033
  - EA\_MenuClick 3025
  - EA\_OnAttributeTagEdit 3084
  - EA\_OnConnectorTagEdit 3085
  - EA\_OnContextItemChanged 3037, 3039
  - EA\_OnContextItemDoubleClicked 3038
  - EA\_OnDeleteTechnology 3092
  - EA\_OnElementTagEdit 3086
  - EA\_OnEndValidation 3050
  - EA\_OnImportTechnology 3094
  - EA\_OnInitializeTechnologies 3089
  - EA\_OnInitializeUserRules 3048
  - EA\_OnMethodTagEdit 3087
  - EA\_OnOutputItemClicked 3027
  - EA\_OnOutputItemDoubleClicked 3028
  - EA\_OnPostActivateTechnology 3090
  - EA\_OnPostCloseDiagram 3042
  - EA\_OnPostInitialized 3043
  - EA\_OnPostNewAttribute 3064
  - EA\_OnPostNewConnector 3062
  - EA\_OnPostNewDiagram 3063
  - EA\_OnPostNewDiagramObject 3063
  - EA\_OnPostNewElement 3061
  - EA\_OnPostNewGlossaryTerm 3067
  - EA\_OnPostNewMethod 3065
  - EA\_OnPostNewPackage 3066
  - EA\_OnPostOpenDiagram 3044
  - EA\_OnPostTransform 3044

## Add-In Model

EA\_OnPreActivateTechnology 3089  
 EA\_OnPreDeleteAttribute 3069  
 EA\_OnPreDeleteConnector 3071  
 EA\_OnPreDeleteDiagram 3071  
 EA\_OnPreDeleteDiagramObject 3072  
 EA\_OnPreDeleteElement 3068  
 EA\_OnPreDeleteGlossaryTerm 3074  
 EA\_OnPreDeleteMethod 3070  
 EA\_OnPreDeletePackage 3073  
 EA\_OnPreDeleteTechnology 3091  
 EA\_OnPreDropFromTree 3079  
 EA\_OnPreExitInstance 3045  
 EA\_OnPreNewAttribute 3080  
 EA\_OnPreNewConnector 3077  
 EA\_OnPreNewDiagram 3078  
 EA\_OnPreNewDiagramObject 3078  
 EA\_OnPreNewElement 3076  
 EA\_OnPreNewGlossaryTerm 3083  
 EA\_OnPreNewMethod 3081  
 EA\_OnPreNewPackage 3082  
 EA\_OnRetrieveModelTemplate 3046  
 EA\_OnRunAttributeRule 3053  
 EA\_OnRunConnectorRule 3053  
 EA\_OnRunDiagramRule 3052  
 EA\_OnRunElementRule 3050  
 EA\_OnRunMethodRule 3054  
 EA\_OnRunPackageRule 3051  
 EA\_OnRunParameterRule 3055  
 EA\_OnStartValidation 3049  
 EA\_OnTabChanged 3047  
 EA\_QueryAvailableCompartments 3034  
 EA\_ShowHelp 3029  
 Interface 3010  
 Introduction 3010  
 MDG Add-Ins 3097  
 MDG Add-Ins, MDG Events 3098  
 MDG Events, MDG\_BuildProject 3099  
 MDG Events, MDG\_Connect 3100  
 MDG Events, MDG\_Disconnect 3101  
 MDG Events, MDG\_GetConnectedPackages 3102  
 MDG Events, MDG\_GetProperty 3102  
 MDG Events, MDG\_Merge 3103  
 MDG Events, MDG\_NewClass 3106  
 MDG Events, MDG\_PostGenerate 3107  
 MDG Events, MDG\_PostMerge 3108  
 MDG Events, MDG\_PreGenerate 3109  
 MDG Events, MDG\_PreMerge 3109  
 MDG Events, MDG\_PreReverse 3110  
 MDG Events, MDG\_RunExe 3111  
 MDG Events, MDG\_View 3112

Model Validation Broadcasts 3047  
 Model Validation Broadcasts,  
 EA\_OnEndValidation 3050  
 Model Validation Broadcasts,  
 EA\_OnInitializeUserRules 3048  
 Model Validation Broadcasts,  
 EA\_OnRunAttributeRule 3053  
 Model Validation Broadcasts,  
 EA\_OnRunConnectorRule 3053  
 Model Validation Broadcasts,  
 EA\_OnRunDiagramRule 3052  
 Model Validation Broadcasts,  
 EA\_OnRunElementRule 3050  
 Model Validation Broadcasts,  
 EA\_OnRunMethodRule 3054  
 Model Validation Broadcasts,  
 EA\_OnRunPackageRule 3051  
 Model Validation Broadcasts,  
 EA\_OnRunParameterRule 3055  
 Model Validation Broadcasts,  
 EA\_OnStartValidation 3049  
 Model Validation Example 3056  
 Post-New Events 3060  
 Post-New Events, EA\_OnPostNewAttribute 3064  
 Post-New Events, EA\_OnPostNewConnector 3062  
 Post-New Events, EA\_OnPostNewDiagram 3063  
 Post-New Events, EA\_OnPostNewDiagramObject 3063  
 Post-New Events, EA\_OnPostNewElement 3061  
 Post-New Events, EA\_OnPostNewGlossaryTerm 3067  
 Post-New Events, EA\_OnPostNewMethod 3065  
 Post-New Events, EA\_OnPostNewPackage 3066  
 Pre New-Object Events 3075  
 Pre-Deletion Events 3067  
 Pre-Deletion Events, EA\_OnPreDeleteAttribute 3069  
 Pre-Deletion Events, EA\_OnPreDeleteConnector 3071  
 Pre-Deletion Events, EA\_OnPreDeleteDiagram 3071  
 Pre-Deletion Events,  
 EA\_OnPreDeleteDiagramObject 3072  
 Pre-Deletion Events, EA\_OnPreDeleteElement 3068  
 Pre-Deletion Events,  
 EA\_OnPreDeleteGlossaryTerm 3074  
 Pre-Deletion Events, EA\_OnPreDeleteMethod 3070

- Add-In Model
  - Pre-Deletion Events, EA\_OnPreDeletePackage 3073
  - Pre-New Events, EA\_OnPreDropFromTree 3079
  - Pre-New Events, EA\_OnPreNewAttribute 3080
  - Pre-New Events, EA\_OnPreNewConnector 3077
  - Pre-New Events, EA\_OnPreNewDiagram 3078
  - Pre-New Events, EA\_OnPreNewDiagramObject 3078
  - Pre-New Events, EA\_OnPreNewElement 3076
  - Pre-New Events, EA\_OnPreNewGlossaryTerm 3083
  - Pre-New Events, EA\_OnPreNewMethod 3081
  - Pre-New Events, EA\_OnPreNewPackage 3082
  - Search Data, XML Format 3020
  - Tagged Value Broadcast Events 3084
  - Tagged Value Broadcast Events, EA\_OnAttributeTagEdit 3084
  - Tagged Value Broadcast Events, EA\_OnConnectorTagEdit 3085
  - Tagged Value Broadcast Events, EA\_OnElementTagEdit 3086
  - Tagged Value Broadcast Events, EA\_OnOperationTagEdit 3087
  - Technology Event, EA\_OnInitializeTechnologies 3089
  - Technology Events 3088
  - Technology Events, EA\_OnDeleteTechnology 3092
  - Technology Events, EA\_OnImportTechnology 3094
  - Technology Events, EA\_OnPostActivateTechnology 3090
  - Technology Events, EA\_OnPreActivateTechnology 3089
  - Technology Events, EA\_OnPreDeleteTechnology 3091
- Additional Collection Class 2260
- AddVariableValue Action
  - Simulation Variable Action 1273
- Administrator
  - Security Permissions 318
- Advanced
  - Element Submenu 100
  - Settings, Connector 1133
  - Settings, Elements 961
  - Tag Management 1140
- Advanced Options
  - Document Report 2654
- Advanced Search Options 715
- Advanced Submenu
  - Package Context Menu, Project Browser 654
- Aggregate
  - Connector 1392
  - Relationship 1392
- Aggregation Connector
  - Change Form 1392
- Alias
  - Element, Apply Auto Naming 906
- Align
  - Elements, Diagram Toolbar 138
  - Multiple Elements 918, 951
- Alignment
  - Submenu 953
- All User Permissions
  - Dialog, User Security 328
  - View, User Security 328
- Allocated Resources
  - View, Project Calendar 578
- Allocated Work
  - Tab, PersonalTasks View 555
- Alternate Path
  - Add To Scenario (Toolbar) 970
- Alternate Path, Scenario 965
- Alternative Image
  - For Element 860
  - In Diagram 860
  - Select 860
  - Stereotype 1459
- Analysis
  - Elements and Connectors 812
  - Models 1794
  - Models, Business Processes 1809
  - Stereotypes 1800
  - Toolbox Pages 812
- Analysis Diagram
  - Description 1801
  - Diagram 1801
  - Elements And Connectors 1801
  - Example 1802
- Analyzer
  - Menu 112
  - Record Submenu 115
- Analyzer Script 2216
  - Options, Filters 2219
- Analyzer Scripts
  - Add New 2175
  - Build 2175
  - Clean 2175
  - Copy 2175
  - Create 2175
  - Debug 2175
  - Delete 2175

- Analyzer Scripts
  - Deploy 2175
  - Edit 2175
  - Export 2175
  - Import 2175
  - Manage, Visual Execution Analyzer 2175
  - Paste 2175
  - Rebuild 2175
  - Rename 2175
  - Run 2175
  - Set As Default 2175
- and any breakpoints should 2203
- Android Debugger
  - Local 2210
  - Remote 2210
  - Script Set Up 2210
- Android Devices
  - Debug Java Applications On 2210
- Android Mobile
  - Application Pattern 2077
- Animation
  - Of Menu Display 162
- ANSI C
  - Code Generation Language Options 2266
  - Modeling Conventions 2084
  - Options 2266
- Anti-Aliased Rendering
  - Of Diagrams 622
- Anti-Aliased Text
  - In Diagrams 622
- Any Attribute
  - XML, Toolbox Icon 2407
- Any Element
  - XML, Toolbox Icon 2405
- Apache Tomcat
  - Server Configuration 2199
  - Server, Debugging 2195
  - Service Configuration 2199
- App Object
  - Automation Interface 2814
- APPDATA
  - Import MDG Technologies 1480
- Appearance
  - Connectors Context Menu Section 1107
  - Default, Of Element 927
- Appearance (Element)
  - Autosize 99
  - Submenu 99
- Appearance, Element
  - Apply Alternate Image 948
  - Apply From Clipboard 948
  - Copy For Element 948
  - Element Context Menu Option 948
  - Hide Name Under Image 948
  - Set Text Font 948
- Applets
  - Java, In Internet Browsers, Debug 2194
- Application Look
  - Dialog 168
- Application Pattern
  - Android Mobile 2077
  - ASP.NET Web Service 2077
  - Generate 2077
  - Java Program 2077
  - MFC Windows 2077
- Application Schema
  - GML, Generate 1987
- Application Schema Files
  - GML 1983
- Application Workspace 67
- Apply
  - RTF Report Filter (Legacy) 2722
  - Stereotype To Dependency Relationship 1405
  - Stereotype To Element 1453
  - Stereotype To UML Object 1453
  - User Lock, Require User Lock To Edit 339
- ArcGIS
  - Attributes 1946
  - Code Generation Language Options 2265
  - Connectivity Rule Connector 1952
  - Connectors 1946
  - Coordinate System, Set 1958
  - Core 1946
  - Disable 1944
  - Elements 1946
  - Export XML Workspace 1961
  - Geometric Network 1946
  - Import from XMI 478
  - Import XML Workspace 1968
  - MDG Technology For 1944
  - Modular Schemas, Export 1962
  - Options 2265
  - Partial Schemas, Modeling 1962
  - Profile 1944
  - Relationship Rule Connector 1956
  - Relationships 1946
  - Schema View 1962
  - Schema, Export 1961
  - Schema, Import 1968
  - Show/Hide System Attributes 1944
  - Show/Hide System Fields 1944
  - Show/Hide System Stereotypes 1944
  - Spatial References 1944
  - Support For In Enterprise Architect 1944

- ArcGIS
  - Toolbox Pages 1946
  - Topology 1946
  - Topology Example 1954
  - Topology Package 1954
  - TopologyMembership Attribute 1954
  - TopologyRule Connector 1954
  - Validate Workspace 1970
  - Vertical Coordinate System, Set 1958
- ArchiMate
  - 1.0 1926
  - 2.0 1926
  - Concept 1926
  - Diagram 1926
  - Disable 1926
  - Elements 1926
  - Enable 1926
  - Extend Profile 1491
  - MDG Technology 1926
  - Migrate 1.0 Model To 2.0 1927
  - Migration 1.0 To 2.0 1926
  - Relationships 1926
  - Toolbox Pages 1926
- Argument
  - Call, Synchronize With Behavior Parameter 1027
  - For Behavioral Parameter 1027
  - Invocation, Synchronize With Behavior Parameter 1027
- Arrange
  - Connectors 1112
- Array
  - View Elements Of, Debugger 2238
- Arrow
  - Flow Direction (SysML Port) 2306
  - Quick Linker 896
- Artifact
  - Base 1358
  - Document 1358
  - Element 1358
  - Element, File 1362
  - Executable Statemachine 1358, 2122
  - Matrix Specification 1358
  - Model View 1358
  - Report Specification 1358
  - Standard Chart 1358
  - Time Series Chart 1358
  - User Story 1358
  - Working Set 561, 1358
- Artifact External
  - Create 1362
- Artifact Internal
  - Create 1362
- ASA
  - Data Repository, Connect To 239
  - ODBC Driver, Set Up 237
  - Repository, Create 237
- ASP .NET
  - Debug 2203
- ASP.NET
  - Application Pattern 2077
- Assembly
  - Connector 1393
  - Relationship 1393
- Asset Contents
  - Package, Reusable Asset Service 286
- Asset Dependencies
  - Package, Reusable Asset Service 288
- Asset Package
  - RAS, Import To Model 291
- Asset Properties
  - Package, Reusable Asset Service, Review 286
- Assets
  - Browse In Reusable Asset Service 284
  - Check Dependencies 302
  - Register Files in Reusable Asset Service 298
  - Register Packages In Reusable Asset Service 298
  - Register Packages Needed By Asset 302
  - Register Technologies in Reusable Asset Service 298
  - Types In Reusable Asset Service 282
  - Update in Reusable Asset Service 304
- Assign
  - Information to Tagged Values 1138
  - People To Changes 2631
  - People To Issues 2631
  - Tagged Values To Item 1137
- Assign Activity
  - BPEL 2.0, Model 1912
- Assignment
  - BPEL 1.1, Create 1889
- Associate
  - Calls With Behaviors 1027
  - Connector 1393
  - Invocations With Behaviors 1027
  - Relationship 1393
- Associated Files
  - Elements 988
- Association 1126
  - Bind To Part (SysML Block) 2304
  - Bind To Property (SysML Block) 2304
  - Bind To Property (SysML) 1393
  - Class 2007

- Association 1126
  - Connector 1393
  - Connector, Set Collection Class 2259
  - Details 1126
  - Direction Indicator 866
  - Dot On Connector 1130
  - Generate Part From (SysML Block) 2304
  - Generate Property From (SysML Block) 2304
  - N-Ary 2007
  - Properties 1126
  - Relationship 1393
  - Set Derived Property 1105
  - Specialisation, Set Up 1123
- Association Class
  - Connector 1398
  - Find In Project Browser 1103
  - Link New Class To Association 1400
  - Locate In Project Browser 1398
  - Relationship 1398
  - Shape Scripts 1501
- Association End
  - Qualifiers 1395, 1396
- Association Role
  - GML 1984
- AST
  - Attributes, In Grammars 1711
  - Grammar, Parsing 1722
  - Nodes, In Grammars 1711
- Asynchronous Signal Message
  - Associate With Signal 1427
  - Connector 1427
  - Relationship 1427
- Attach
  - Note To Link 1111
- Attach To Process Dialog 2243
- Attribute
  - Add And Delete, Automation Interface Code Example 3000
  - Add To Element, In-place Editor 1040
  - ArcGIS 1946
  - As Executable Statemachine Artifact 667
  - Collections 1003
  - Connect To 1110
  - Constraints 1004
  - Context Menu, Project Browser 667
  - Copy Between Elements 932
  - Create 999
  - Create Fast, Option 2253
  - Create Object From 1008
  - Definition 999
  - Delete 1001
  - Delete If Not In Code In Reverse Synchronization 2253
  - Derived 1001
  - Dialog 999
  - Dialog, Constraints Tab 1004
  - Dialog, Detail Tab 1003
  - Dialog, General Tab 1001
  - Disconnect From 1110
  - Drag Onto New Element 989
  - Edit Keyword 1037
  - Edit Name, In-Place Editor 1034
  - Edit Scope 1036
  - Edit Stereotype, In-place Editor 1035
  - Fast Create 999
  - Field Substitution Macros 1651
  - GML 1984
  - Imported, Default Name Generated From 2253
  - Inherited, Display 1007
  - Inherited, Show 845
  - Initializer, Override 1007
  - Introduction 999
  - Link To Element Via Object 1008
  - Message Part, WSDL 2431
  - Modify 999
  - Move Between Elements 933
  - Multiplicity 1003
  - Of Custom Toolbox Page 1563
  - PData & StyleEx, Diagram Profiles 1573
  - Private, Icon 670
  - Properties, Create 1006
  - Protected, Icon 670
  - Qualifiers 1003, 1396
  - Redefine Property 1003
  - Show On Diagram 845
  - Special, Define Child Diagram Types 1512
  - Special, Define Composite Elements 1511
  - Special, Define Tag Grouping 1514
  - Special, Metatype In Profiles 1503
  - Special, Stereotype In Profiles 1503
  - Stereotyped, For Columns 2345
  - Subsetted Property 1003
  - Tagged Values 1005
  - UML Property, isUnique 1003
  - Work With, Automation Interface Code Example 3006
  - XML Schema Local Attribute 2398
  - XML Schema, Any 2407
- Attribute Class
  - Alias, Automation Interface 2911
  - Automation Interface, ElementFeatures Package 2911
- Attribute Group Element

- Attribute Group Element
  - XML, Toolbox Icon 2399
- AttributeConstraint
  - Automation Interface, ElementFeatures Package 2915
- Attributes
  - AST, In Grammars 1711
- AttributeTag
  - Automation Interface, ElementFeatures Package 2916
- Audit
  - Requirements 1785
  - Scope 448
- Audit History Tab
  - Description 455
  - How To Display 455
  - On Output Window 679
  - On System Output Window 169
- Audit Options
  - All 448
  - Connectors Audited 448
  - Core Structural 448
  - Custom 448
  - Elements Audited 448
  - Maintenance 448
- Audit View
  - Advanced Mode 453
  - Audit Changes 453
  - Controls 453
  - Custom Time Periods 453
  - Deleted Mode 453
  - Display Database Changes 453
  - Filter By Time 453
  - Performance Problems 457
  - Raw Mode 453
  - Refresh 453
  - Search 453
  - Slow Loading 457
  - Slow Navigation 457
  - Sort 453
  - Standard Mode 453
- Auditing
  - Alternative To Differencing 457
  - And Document Reporting 446
  - And Performance Of Enterprise Architect 446
  - Audit Tree 451
  - Clear Logs 448
  - Display Audit Results 451
  - Enable 448
  - How To Invoke 447
  - Include Reverse Engineering 448
  - Include XMI Export 448
  - Include XMI Import 448
  - Introduction 446
  - Large Deletion Issue 456
  - Level, Core 448
  - Level, Extended 448
  - Level, Standard 448
  - Load Logs 448
  - Performance Issues 456
  - Quick Start 447
  - Record Display 451
  - Reverse Engineering Issue 456
  - Revert Changes 451
  - Roll Back Changes 451
  - Save Logs 448
  - Settings 448
  - Use Database Timestamp 448
  - View 451
  - XMI Import Issue 456
- Authentication
  - Accept Windows Authentication 323
  - Automatic Delete On Relocation Of Project 323
- Author
  - Attributes 2827
  - Define 1153
  - From Windows Active Directory 1153
  - Methods 2827
- Author Collection
  - Automation Interface Repository 2827
- Auto Counter
  - Elements, Apply 906
  - Of Elements, Set Up 904
- Auto Increment Column Values
  - Impact Of Replication 598
  - Impact Of XMI Import 598
  - Reset 598
- Auto Naming
  - And Requirements 1773
  - Element Aliases, Apply 906
  - Elements, Apply 906
  - Maintenance Items 2623
  - Of Element, Set Up 904
  - Task Items 527
  - Test Items 2605
- Auto Numbering
  - And Requirements 1773
  - Apply 529
  - Elements, Apply 906
  - Maintenance Items 2623
  - Of Elements, Set Up 904
  - Task Items 527
  - Test Items 2605
- Auto Route

- Auto Route
  - Diagram Layout 890
- Auto Save
  - Diagram Changes 625
- Auto Spell
  - In Document Editor 1089
- Autocompletion List
  - Code Editor, Common 2160
- Autohide
  - Docked Windows 132
  - Floating Windows (Dock First) 132
  - Reveal Autohidden Window 132
  - Turn Off 132
  - Turn On 132
- Autolayout
  - Diagram 891
  - Options 891
- Automatic
  - Save Of Diagram Changes 625
- Automatic Exclusive Edit Lock 318
- Automatically Hidden Windows
  - Animate 84
- Automation
  - Of Enterprise Architect 2790
- Automation Interface
  - App Object 2814
  - Attribute Class, ElementFeatures Package 2911
  - AttributeConstraint, ElementFeatures Package 2915
  - AttributeTag, ElementFeatures Package 2916
  - Available Resources 2811
  - Call Executables From Enterprise Architect 2804
  - Call From Enterprise Architect 2810
  - Code Example, Add And Delete Attributes 3000
  - Code Example, Add And Delete Methods 3000
  - Code Example, Add And Manage Diagrams 2999
  - Code Example, Add And Manage Elements 2997
  - Code Example, Add And Manage Packages 2996
  - Code Example, Add Connector 2998
  - Code Example, Add Stereotypes 3006
  - Code Example, Iterate Through EAP File 2996
  - Code Example, Open The Repository 2995
  - Code Example, Use Element Extras 3001
  - Code Example, Use Repository Extras 3004
  - Code Example, Work With Attributes 3006
  - Code Example, Work With Methods 3007
  - Code Examples, Introduction 2994
  - Connect From Borland Delphi 7.0 2805
  - Connect From Java 2805
  - Connect From MS C# 2805
  - Connect From MS Visual Basic 6.0 2805
  - Connect To 2804
  - Connector Package Diagram 2933
  - Connector, Connector Package 2933
  - ConnectorConstraint, Connector Package 2939
  - ConnectorEnd, Connector Package 2940
  - ConnectorTag, Connector Package 2943
  - ConstLayoutStyles Enum 2816
  - Constraint, Element Package 2879
  - CreateBaselineFlag Enum 2817
  - CreateModelType Enum 2818
  - CustomProperties Collection, ElementFeatures Package 2918
  - Diagram Package 2948
  - Diagram, Diagram Package 2948
  - DiagramLinks, Diagram Package 2955
  - DiagramObject, Diagram Package 2956
  - DocumentBreak Enum 2818
  - DocumentGenerator Class 2985
  - DocumentGenerator Interface 2985
  - DocumentPageOrientation Enum 2818
  - DocumentType Enum 2819
  - EAEditionTypes Enum 2819
  - Effort, Element Package 2880
  - Element Package Diagram 2877
  - Element Package, File 2895
  - Element, Element Package 2881
  - ElementFeatures Package Diagram 2910
  - EmbeddedElements Collection, ElementFeatures Package 2918
  - Enumerations 2815
  - EnumRelationSetType Enum 2820
  - Examples 2804
  - Examples and Tips 2809
  - ExportPackageXMIFlag Enum 2820
  - Introduction 2804
  - Issue, Element Package 2896
  - Mail Interface Package 2990
  - MailInterface Class, Mail Interface Package 2990
  - MDGMenus Enum 2821
  - MessageFlag 2821
  - Method, ElementFeatures Package 2919
  - MethodConstraint, ElementFeatures Package 2923
  - MethodTag, ElementFeatures Package 2924
  - Metric, Element Package 2898
  - Model 2813
  - Models Collection 2850



## Automation Interface

- ObjectType Enum 2822
- Package 2813
- Parameter, ElementFeatures Package 2925
- ParamTag, ElementFeatures Package 2927
- Partitions Collection, ElementFeatures Package 2928
- Project Interface 2962
- Project, Project Interface 2962
- Properties, ElementFeatures Package 2929
- Property ElementFeatures Package 2929
- PropType Enum 2823
- Reference 2812
- ReloadType Enum 2823
- Repository Class 2850
- Repository Package 2826, 2835
- Repository, Author Collection 2827
- Repository, Client Collection 2828
- Repository, Collection Class 2829
- Repository, Datatype 2831
- Repository, EventProperties 2833
- Repository, EventProperty 2833
- Repository, ModelWatcher 2834
- Repository, ProjectIssues 2844
- Repository, ProjectResource 2846
- Repository, ProjectRole 2847
- Repository, PropertyType 2848
- Repository, Reference 2848
- Repository, Stereotype 2873
- Repository, Task 2874
- Repository, Term 2876
- Requirement, Element Package 2899
- Resource, Element Package 2900
- Risk, Element Package 2902
- RoleTag, Connector Package 2944
- Scenario, Element Package 2903
- ScenarioDiagramType Enum 2824
- ScenarioExtension, Element Package 2904
- ScenarioStep, Element Package 2905
- ScenarioStepType Enum 2825
- ScenarioTestType Enum 2825
- Set Up Visual Basic 2808
- Simulation Class, Simulation Package 2993
- Simulation Package 2993
- Swimlane, Diagram Package 2961
- SwimlaneDef, Diagram Package 2958
- Swimlanes, Diagram Package 2960
- TaggedValue, Element Package 2907
- TemplateBinding Class, 2946
- TemplateParameter, ElementFeatures Package 2931
- Test, Element Package 2908

- Transitions Collection, ElementFeatures Package 2932

- Using 2805

- VB GetObject Support 2814

- XMIType Enum 2825

## Autonaming

- Add In Specification Manager 1758

- Apply In Specification Manager 1759

- Apply, Package Context Menu 654

## Autonumbering

- Add In Specification Manager 1758

- Apply In Specification Manager 1759

## Autosize

- Element, Single 851

- Elements, As Group 851

## Available Resources

- Automation Interface 2811

**- B -**

## Background Tile

- Diagram Background, Set In Theme 611

## Bar Chart

- 2D, Change Appearance 2775

- 3D, Change Appearance 2778

## Base

- Artifact 1358

## Base Project

- Copy 211

- New 211

## Baseline

- And Differences, Overview 457

- Comparison In Reusable Asset Service 291

- Considerations 459

- Create 462

- Delete 461

- Diagram Comparison 466

- Export 461

- Import 461

- Include XMI Stubs 462

- Load From Alternative Model 461

- Manage 461

- Merge With Current Model, Overview 457

- Model 459

- Overview 459

- Requirements 1785

- Scenarios 459

- Versions 462

## Baseline Comparison

- Context Menu 470

- Keyboard Options 470

- Merge Options 470

- Baseline Comparison
  - Toolbar 470
- Baseline Diagram Compare
  - Dialog 466
- BaseModel Script
  - InnoDB 220
  - MyISAM 220
- Basic Path, Scenario 965
- Batch Generate
  - Elements With Code 103
  - Resource Documents, Automatically 2668
  - Resource Documents, Manually 2668
- Batch XML
  - Export 492
  - Import 493
- Behavior
  - Associate With Calls 1027
  - Associate With Invocations 1027
  - Instance 1009
  - Of Operation, Display In Diagram 1018
  - Operation, Associate With 1018
  - Parameter, Tagged Values 1030
  - Parameters, Define 1028
  - Synchronize Call Arguments With Parameters 1027
  - Synchronize Invocation Arguments With Parameters 1027
- Behavior Call
  - Actions For RuleTask Operations 1830
  - Interaction Occurrence 1026
  - Modeling 1026
  - Parameter Arguments 1027
- Behavioral Code Generation
  - From State Machines 2122
- Behavioral Diagram
  - Elements 1265
  - Overview 1198
- Behavioral Model Templates 1688
- Behavioral Modeling 1014
- Behavioral Models
  - Generate Code From 2121
- Behavioral Parameter
  - Edit 1028
  - Extend 1028
  - Reassign 1028
  - Set 1028
- Behaviors
  - Of Activities 1026
  - Of Interactions 1026
- Bend
  - Connector At Cursor 1114
- Bend Connector 55
- Bezier Lines 1114
- Binary Module
  - Import, Reverse Engineering 2143
  - Languages Supported 2143
- Bind
  - Breakpoints 2229
  - Fail To, Breakpoints 2229
- Bind Package
  - To Team Review Folder (Topic), Specification Manager 1749
- Bind Topic
  - To Project Browser Package, Team Review 346
- Binding
  - HTTP, WSDL Model 2437
  - In Connector 1129
  - SOAP, WSDL Model 2437
  - WSDL Diagram 2437
  - WSDL Element 2437
- Binding Class
  - And Binding Connector 1444
- Binding Expression
  - Binding Connector 1444
- Bitmap Image
  - In Diagrams 860
- Block (SysML)
  - Generate Part From Association 2304
  - Generate Property From Association 2304
  - Port, Flow Direction 2306
- Block Element
  - Compartments 2305
- Blue
  - Exclamation Mark 340
- Bookmark
  - Clear 80
  - Clear All 80
  - Code Editor, Common 2157
  - Create Reference To 1083
  - Delete 1083
  - For Document Report 2730
  - Go To 1083
  - In Master Document Element 2669
  - Insert In Document 1083
  - Multiple Elements 591
  - Package As 2730
  - Selected Element 80
  - Triangle 591
- Bookmark Selected
  - Menu Option (Edit Menu) 80
- Border
  - Element, Red 340
  - Width, Set In Theme 611
- Boundary

**Boundary**

- Element 1997
- Element Settings 1348
- Element, Create 1998
- Insert Image As, In Diagram 1362
- Object Settings 1348
- Properties 1348

**Box**

- Diagram Layout 879

**BPEL**

- 1.1 Assignments, Create 1889
- 1.1 Create Model 1873
- 1.1 End Event, Model 1885
- 1.1 Example 1871
- 1.1 Generate 1893
- 1.1 Intermediate Event, Model 1878
- 1.1 Model 1871
- 1.1 Model, Create 1873
- 1.1 Modeling Restrictions 1871
- 1.1 Package Structure 1873
- 1.1 Package Structure, Create 1871
- 1.1 Process, Capture Via BPMN 1.1 1871
- 1.1 Process, Properties 1876
- 1.1 Start Event, Model 1877
- 1.1, BPMN Elements Not Mapable 1876
- 1.1, Create Assignments 1889
- 1.1, Looping Constructs 1881
- 1.1, Model A Process 1876
- 1.1, Model Activity 1881
- 1.1, Model Gateway 1884
- 1.1, Model Pool 1888
- 1.1, Model Sequence Flow Connector 1887
- 1.1, Sub Process Types 1881
- 1.1, Task Types 1881
- 1.1, Web Service, Create 1890
- 2.0 Assign Activity, Model 1912
- 2.0 End Event, Model 1907
- 2.0 Example 1894
- 2.0 Intermediate Event, Model 1901
- 2.0 Looping Constructs 1904
- 2.0 Model 1894
- 2.0 Model Activity 1904
- 2.0 Model Gateway 1906
- 2.0 Model, Create 1897
- 2.0 Modeling Restrictions 1894
- 2.0 Package Structure 1897
- 2.0 Package Structure, Create 1894
- 2.0 Process, Capture Via BPMN 2.0 1894
- 2.0 Process, Model 1898
- 2.0 Sub Process Types 1904
- 2.0 Task Types 1904
- 2.0 WSO, Create 1914

- 2.0, Generate Code 1916
- 2.0, Model Data Object 1909
- 2.0, Model Property 1910
- 2.0, Model Sequence Flow Connector 1910
- 2.0, Start Event, Model 1900
- 2.0, Use Of BPMN Pool 1911
- Concepts 1870
- Create 2.0 Model 1897
- Diagram 1870
- Disable 1870
- In Enterprise Architect 1870
- MDG Technology 1870
- Model Validation 1917
- Modeling Restrictions 1870
- Process Element, BPMN 1.1 1848
- Toolbox, Via BPMN 1.1 1871
- Toolbox, Via BPMN 2.0 1894

**BPMN**

- 1.0 1845
- 1.0 Elements 1848
- 1.0 Relationships 1848
- 1.0 Toolbox Pages 1848
- 1.1 1845
- 1.1 Elements 1848
- 1.1 Relationships 1848
- 1.1 Toolbox Pages 1848
- 2.0 1845
- 2.0 Business Process Elements 1852
- 2.0 Business Process Relationships 1852
- 2.0 Business Process Toolbox Pages 1852
- 2.0 Choreography Elements 1854
- 2.0 Choreography Relationships 1854
- 2.0 Choreography Toolbox Pages 1854
- 2.0 Collaboration Elements 1856
- 2.0 Collaboration Relationships 1856
- 2.0 Collaboration Toolbox Pages 1856
- 2.0 Conversation Elements 1859
- 2.0 Conversation Relationships 1859
- 2.0 Conversation Toolbox Pages 1859
- 2.0 Elements 1851
- 2.0 Model 1869
- 2.0 Relationships 1851
- 2.0 Toolbox Pages 1851
- 2.0 Type Elements 1861
- 2.0 Type Relationships 1861
- 2.0 Type Toolbox Pages 1861
- 2.0 XML 1869
- Change Element Appearance 1862
- Concepts 1845
- Core Page 1848
- Create Simulation Model 2521
- Diagram 1845

**BPMN**

- Diagram Flow Orientation, 2.0 1845
- Disable 1845
- Element Appearance, Change 1862
- Extend Profile 1491
- MDG Technology 1845
- Message Flow Rules, 2.0 1845
- Migrate 1.0 Model To 1.1 1866
- Migrate 1.1 Model To 2.0 1868
- Sequence Flow Rules, 2.0 1845
- Serializing 2.0 In XML 1869
- Simulation 2521
- Simulation Of Process, Compare With UML Activity 2524
- Simulation, Control Via Tagged Values 2523
- Simulation, Initialization 2523
- Tagged Values In Simulation 2523
- Toolbox Page 1845
- Types Page 1848
- Update Version Via Toolbox Element 1862
- Version Differences In Appearance 1862

**BPMN 1.1**

- Activity Types 1881
- BPEL 1.1 Sequence Flow 1887
- Elements Not Mappable to BPEL 1.1 1876
- End Event, Types 1885
- Gateway Types 1884
- Intermediate Event, Types 1878
- Pool 1888
- Start Event, Types 1877
- Sub-Process Types 1881
- Task Types 1881

**BPMN 2.0**

- Activity Types 1904
- BPEL Sequence Flow 1910
- Data Object 1909
- End Event, Types 1907
- Gateway Types 1906
- Intermediate Event, Types 1901
- Property 1910
- Start Event, Types 1900
- Sub-Process Types 1904
- Task Types 1904

**Brace Matching**

- Code Editor, Common 2157

**Branching**

- Version Control 394

**Branching Macros**

- Code Template Syntax 1684

**BREAD Matrix**

- Introduction 732

**Breakpoint**

- Behavior 2226

- Constraints 2226

- Delete 2224

- Difference From Recording Marker 2538

- Disable 2224

- Enable 2224

- Fail To Bind 2229

- Failure Message 2224

- Line Of Action 2226

- Management 2224

- Marker On Element During Simulation 2477

- Model Simulation, Using 2477

- Properties 2226

- Set For Modifiable Data 2229

- Set In Code 2226

- Sets, Model Simulation 2477

- States 2224

- Trace Statements 2226

- Breakpoints And Markers Window 2224, 2540

**Broadcast Event**

- Add-In Model 3029

- EA\_FileClose 3040

- EA\_FileNew 3041

- EA\_FileOpen 3042

- EA\_OnPostCloseDiagram 3042

- EA\_OnPostInitialized 3043

- EA\_OnPostOpenDiagram 3044

- EA\_OnPostTransform 3044

- EA\_OnPreExitInstance 3045

- EA\_OnRetrieveModelTemplate 3046

- EA\_OnTabChanged 3047

**Broadcast Events**

- EA\_OnPreNew 3075

- Pre New-Object Events 3075

**BroadcastSignal Action**

- Signal Tab 1268

**Browser**

- Element 989

**Build**

- Application 2221

- Application, Analyzer Menu Option 112

- Clean Commands, Add 2181

- Code 2172

- Execute Command 2221

- Locate Compiler Error In Code 2221

- Output 2221

- Toolbar 147

**Build Script**

- Create 2180

- Deploy Script, Create 2218

- Execute, Build Toolbar Option 147

- Introduction 2180

- Build Script
  - Recursive 2180
  - Run Script, Create 2217
  - Test Script, Create 2182
- Build Systems Using UML
  - Enterprise Architect 9
- Built-In
  - Diagram Types 1572
- Business
  - Models 1804
  - Scenarios & Requirements 992
- Business Analyst
  - And Enterprise Architect 185
  - Project Role 185
- Business and Software Engineering Edition
  - Of Enterprise Architect 20
- Business Domain Model
  - Create 1824
- Business Interaction Diagram
  - Description 1805
  - Elements And Connectors 1805
  - Example 1805
- Business Model
  - Analysis 1809
- Business Modeling
  - Business Process Outline 1813
  - Events 1812
  - Example 1807
  - Goals 1813
  - Information 1811
  - Inputs 1811
  - Outputs 1812
  - Process Element 1810
  - Process Modeling Notation 1810
  - Processes 1807
  - Resources 1811
  - Traceability 723
- Business Modeling Diagram
  - Description 1805
  - Elements And Connectors 1805
  - Example 1805
- Business Process
  - Analysis 755, 758
  - Model Template 755
  - Model, Template 758
  - Outline 1813
- Business Process Execution Language (BPEL) 1870
- Business Process Modeling 1801
- Business Process Modeling Notation (BPMN) 1845
- Business Rule
  - Add Rule Action To 1837
  - Add To Rule Task 1833, 1834
  - Bind To Actions 1839
  - Bind To Conditions 1839
  - Define Computation Rules 1840
  - Define Rule Conditions 1836
  - Dependent Rules 1840
  - Element 1821
  - Export To CSV File 1842
  - Generate Code 1843
  - Modeling 1821
  - Parameters As Variables 1829
  - Remove From Rule Task 1834
  - Rule Expression 1840
  - Validate 1841
- Business Rule Modeling
  - Add Rule To Task 1834
  - Advantages 1814
  - Bind Rules To Conditions And Actions 1839
  - Business Domain Model, Create 1824
  - Business Rules 1821
  - Compose Rules 1833
  - Computation Rule Table 1833, 1840
  - Decision Table 1833
  - Decision Table, Rule Actions 1837
  - Define Rule Conditions 1836
  - Develop A Business Rules Model 1816
  - Export Rules To CSV File 1842
  - Generate Code 1843
  - In Class Operations 1830
  - In RuleTask Actions 1821
  - Introduction 1814
  - Overview 1814
  - Remove Rule From Task 1834
  - Rule Composer 1833
  - Rule Flow Diagram 1826
  - Rule Model 1821
  - Vocabulary 1824
- Business Rules
  - Advantages 1814
  - Develop A Model 1816
  - Diagram 1816
  - Generate Model 1818
  - Introduction To 1814
  - Managing 1814
  - Model Structure 1816
  - Modeling 1814
  - Overview 1814
  - Using 1814
- Business Rules Model
  - Diagram 1818
  - Generate 1818
  - Structure 1818

# - C -

## C

- Code Generation Language Options 2266
- Debugging, General Setup 2189
- Import, Reverse Engineering 2139
- Modeling Conventions 2084, 2086
- Object Oriented Programmiing 2086
- Options 2266

## C#

- Code Generation Language Options 2268
- Import, Reverse Engineering 2139
- Modeling Conventions 2088
- Options 2268
- Transformation 2021

## C++

- Code Generation 2269
- Debug Symbols 2191
- Debugging, General Setup 2189
- Implementation Files 2269
- Import, Reverse Engineering 2139
- Language Options 2269
- Modeling Conventions 2090
- Modeling Conventions, CLI Extensions 2094
- Modeling Conventions, Managed 2093
- Sample Model, Visual Execution Analysis 2529
- Set Up Debug Session 2189
- Transformation 2023

## Calendar

- Allocated Resources View 578
- Calendar Panel 571
- Calendar Panel, Configure Event Subtypes 577
- Diary Panel 571
- Display Events 574
- Options 571
- Overview 571
- Project 571
- Project CalendarView 574
- Project Tasks View 581
- Recurring Events 574
- Toolbar 571

## Calibration

- Of Project Factors 584

## Call

- Associate With Behaviors 1027
- Automation Interface From Enterprise Architect 2810
- Re-associate With Behavior 1027
- Self Message 1422
- Synchronize Arguments With Behavior Parameters 1027

## Call Action

- Pin As Argument 1277

## Call Actions 2134

## Call Behaviors

- In Simulation 2489
- Simulate Activity Behavior 2489
- Simulate Class Behavior 2489

## Call Stack

- Create Sequence Diagram 2241
- Options 2239
- View 2239
- Window 2239

## CallBehavior

- In Activity Return Simulation 2501

## CallBehavior Action

- In Simulation, Instantiate Member Object 2480

## CallOperationAction 2134

## Calltips

- Code Editor, Common 2160

## Camel Case

- Naming Format 2067

## Cancel

- Default Diagram, Model 844
- Validation 89

## Capture State Changes

- Setup To, Visual Execution Analyzer 2551

## Cardinality (Multiplicity)

- Define 1151

## CASE Tool

- Enterprise Architect 3

## Category

- Add To Team Review 350
- Create 350
- Delete 346
- Password Protect Text 357

## Central Buffer Node

- Element 1285

## Chaining

- Transformations 2019

## Change

- Add 2633
- BPMN Element Appearance 1862
- Connector Source Or Target 1113
- Connector Type 1113
- Diagram Type 840
- Element 2631, 2633
- Element Type 100, 918
- Elements And Requirements 1785
- Form Of Aggregation Connector 1392
- Hide Stereotype Letter 2633
- Items, Element Maintenance 2623
- Roll Back In Audit Log 451

- Change
  - Show Stereotype Letter 2633
  - Tracking 445, 446
- Change Conflicts
  - Resolve 315
- Change Management
  - And Requirements 1785
  - Auditing 382
  - Baselines And Differences 382
  - In Enterprise Architect 382
  - Introduction 382
  - Maintenance Diagram 2634
  - Model Transfer 473
  - Project Data Transfer 382
  - Tracking Changes 445
  - Version Control 382, 383
- Change Tracking In Document Editor
  - Accept Changes 1091
  - Hide Changes 1091
  - Locate Changes 1091
  - Reject Changes 1091
- Changes Report
  - On Elements, Generate 2740
- Character Set
  - Set Up For Document Report 2666
  - Set Up For RTF Report (Legacy) 2725
- Chart
  - Appearance Options 2774
  - Appearance, Change 2774
  - Details, Appearance Page 2774
  - Diagrams-Only Report 2788
  - Doughnut 2782
  - Doughnut, 3D 2782
  - Dynamic Generation 2763
  - Element 2763
  - From Resources Window, Create 2763
  - From Toolbox, Create 2763
  - Host On Web 2788
  - Include In Reports 2788
  - Introduction 2762
  - Key Features 2762
  - Overview 2762
  - Pie, 3D 2782
  - Pie, Change Appearance Of 2782
  - Pie, Flat 2782
  - Pie, Options To Change 2782
  - Print 2788
  - Refresh 2763
  - Report On 2788
  - System Provided 2763
  - Torus, 3D 2782
  - Types Available 2762
- Chart, Model View
  - Define 2765
  - Source Of Data - SQL Query 2765
  - SQL Query As Source Of Data 2765
- Chart, Standard
  - 2D Bar, Change Appearance Of 2775
  - 2D Bar, Options To Change 2775
  - 3D Bar, Change Appearance Of 2778
  - 3D Bar, Options To Change 2778
  - Apply Element Filters To Package Search 2772
  - Cluster, 2D 2775
  - Cluster, 3D 2778
  - CSV Data As Source 2773
  - Data 2770
  - Data Source, Define 2770
  - Data Types, Specify 2770
  - Filter Elements 2772
  - Horizontal Bar, 2D 2775
  - Horizontal Bar, 3D 2778
  - Package As Source Of Data 2771
  - Source Of Data - CSV File 2773
  - Source Of Data - Package 2771
  - Source Of Data - SQL Query 2773
  - SQL Query As Source Of Data 2773
  - Stack Percent, 2D 2775
  - Stack Percent, 3D 2778
  - Stack, 2D 2775
  - Stack, 3D 2778
  - Type, Specify 2770
  - Vertical Column, 2D 2775
  - Vertical Column, 3D 2778
- Chart, Time Series
  - Change Appearance 2785
  - Define 2767
  - Line 2785
  - Percentage Stacked Line 2785
  - Source Of Data - SQL Query 2767
  - SQL Query As Source Of Data 2767
  - Stacked Line 2785
  - Stacked Spline 2785
  - Stepped Line 2785
- Check
  - Data Integrity 597
  - Model Integrity 597
  - Project Integrity 597
- Check Constraint
  - Create 2370
- Check In
  - Explanation 427
  - Model Branch 434
  - Multiple Packages 434
  - Offline Packages 392

- Check In
  - Packages 432
- Check Out
  - Explanation 427
  - Model Branch 433
  - Multiple Packages 433
  - Package, Undo 432
  - Packages 431
  - Packages Offline 392
  - Revision Of Package 440
- Check Spelling
  - Auto Spell 1089
  - In Document Editor 1089
- Checked In Package
  - Icon 670
- Checked Out Package
  - Icon 670
- Child
  - Confirm Element As Parent 911
- Child Diagram
  - Of Composite Element 936
  - Switch On Composite Element 936
- Child Element
  - Paste Object As 833
- ChildElement
  - Shape Script 1604
- Choice
  - Element 1286
- Circle
  - Diagram Layout 875
- Circle Notation
  - For Interface Element 1377
  - For Shaped Elements 1355
- Class
  - Active Classes 1365
  - Collection, Set 2259
  - Created In Transformation, Connect To 2063
  - Default Object Variable, In Simulation 2479
  - Element 1363
  - Elements And Connectors 803
  - Elements, Imported 2379
  - Field Substitution Macros 1652
  - In Simulation 2479
  - Instance 1009
  - Make Into Association Class 1400
  - Members, Show/Hide On Diagram 831
  - Model Template 755, 761
  - Parameterized Classes (Templates) 1365
  - Partial 2088
  - Partial, Generate 2115
  - Reset Options 2279
  - Show Realised Interfaces On Diagram 865
  - Simulate Behavior 2489
  - Source Code Generation 2111
  - Structured Tagged Value, Profile 1495
  - Toolbox Pages 803
  - View 769
- Class Diagram
  - Description 1184
  - Elements And Connectors 1184
  - Example 1185
- Class Markup Selection Dialog 2542
- Class Model
  - Generate XML Schema From 2413
- CLASSGUID
  - Add-In Hidden Field 3019
- Classifier
  - Behavior 1314
  - Drop As Attribute 833
  - Drop As Link 833
  - Drop As New Instance 833
  - Examples 1009
  - Features 1009
  - Instance Of 1009
  - Item Conveyed 1413
  - Of Object, Methods 1010
  - Of Objects 1009
  - Properties 1189
- Classify
  - Objects Based On Class Associations 1012
- CLASSTYPE
  - Add-In Hidden Field 3019
- Clean
  - Data 597
  - Project 597
- Clean Commands
  - Add To Build Script 2181
- Clean Project
  - Prior To Build 2181
- Clear All Bookmarks
  - Menu Option (Edit Menu) 80
- Clear Project Clipboard
  - Menu Option (Edit Menu) 80
- Clear Selection
  - Menu Option (Edit Menu) 80
- ClearVariable Action
  - Simulation Variable Action 1273
- CLI Extensions
  - C++ Modeling Conventions 2094
- Client
  - Define 1157
- Client Collection
  - Automation Interface Repository 2828
- Clipboard



- Clipboard
  - Copy Team Review Path To 346
- Clipboard File Format
  - Define 605
- Close
  - Full Screen 121
- Close Project
  - Menu Option (File Menu) 79
- Cloud Connection
  - Access To 68
  - In Registry Asset Service 283
- Cloud Server
  - Configure Models 262
  - Configure Server 262
  - General Settings 262
  - HTTP Ports 262
  - IIS Integration 262
  - Installation 262
  - Security Considerations 262
  - Set Up 262
  - TCP Connections 262
- Cloud Service Connection
  - In Reusable Asset Service 282, 293
- Cloud Services
  - Benefits 262
  - Connect To Project Via 266
  - Connecting To Model Via 262
  - Facilities Available Using 267
  - Introduction 262
  - Use 262
- CLR Versions
  - Debug .NET 2201
- Clustered
  - Index 2365
- Code
  - Breakpoint, Set 2226
  - Build 2172
  - Debug 2172
  - Debug, Step Into Function Calls 2231
  - Debug, Step Out Of Functions 2231
  - Debug, Step Over Lines Of Code 2231
  - Debug, Step Through Function Calls 2548
  - Delete From Features In Model In Fwd Synchronization 2253
  - Generate For Business Rule 1843
  - Generated From State Machine 2128
  - Generation, Toolbar 144
  - Import, Select Language 144
  - Imported, Writing Grammar For 1705
  - Language, Set Default 144
  - Synchronize 1638
- Code Breakpoint
  - Set 2226
- Code Editor
  - Common 133
  - Compare 2148
  - Customize 133
  - Different Types Of 133
  - External 2148
  - Functions 133
  - Functions, Editing 2157
  - Functions, Overview 2157
  - Internal (External Code) 2148
  - Languages Supported 2147
  - Overview 133
  - Supported Languages 2147
  - Variants 2148
  - View 2148
  - Window 2148
- Code Editor, Common
  - Autocompletion List 2160
  - Automatic Indentation 2157
  - Bookmarks 2157
  - Brace Matching 2157
  - Calltips 2160
  - Commenting Selections 2157
  - Configure Search Options 2152
  - Context Menu 2152
  - Create Use Case For Method Option 2155
  - Cursor History 2157
  - Debug Tooltips 2236
  - Find And Replace 2161
  - IME 2152
  - Input Method Editor 2152
  - Intelli-sense 2160
  - Key Bindings 2167
  - Line Selection 2157
  - Manual Indentation 2157
  - Mouseovers 2160
  - Scope Guides 2157
  - Search Facility 2152
  - Syntax Highlighting 2157
  - Tooltips, Debug 2236
  - Zooming 2157
- Code Engineering
  - And MDG Integration 2079
  - Broad View Of, In Enterprise Architect 2076
  - Code, Reverse Engineer 2136
  - Default UML To XML Schema Mappings 2415
  - Default UML To XSD Mappings 2415
  - Eclipse 2079
  - From Behavioral Models, Hardware Description Languages Supported 2121

## Code Engineering

- From Behavioral Models, Software Languages Supported 2121
- Generate Code For Single Class 2113
- Generate Group of Classes 2114
- Generate Package 2115
- Generate Package Source Code 2115
- Generate Source Code 2111
- Hardware Description Languages Supported 2121
- Introduction 2073
- Languages Supported, Hardware Description Languages 2131
- Namespaces 2120
- Object Lifetimes 2253
- Package Contents, Update 2117
- Referenced XML Schema 2420
- Reverse Engineer Source Code 2136
- Settings 2246
- Settings, Attribute/Operation Options 2253
- Settings, Code Generation Constructor/Destructor Options 2253
- Settings, Code Page for Source Editing 2255
- Settings, General Code Options 2246
- Settings, Import Component Types 2249
- Settings, Source Code Options 2247
- Supported Hardware Description Languages 2131
- Supported Software Languages 2111
- Synchronization 2136
- Synchronize Model And Code 2118
- Synchronize Package Tree 2117
- Unicode Character Set 2255
- Update Package Contents 2117
- Visual Studio 2079
- With Enterprise Architect 2076
- XML Schema 2387
- XML Schema, Generate XSD 2417
- XML Schema, Import XSD 2420
- XML Schema, Model XSD 2387
- XSD 2387

## Code Engineering Submenu

- Element Context Menu, Project Browser 664
- Package Context Menu, Project Browser 657

## Code Generation

- ActionScript Language Options 2264
- Ada 2005 Language Options 2264
- ArcGIS Language Options 2265
- Behavioral, From State Machines 2122
- C Language Options 2266
- C# Language Options 2268
- C++ Language Options 2269

- Delphi Language Options 2271
- From Activity Diagrams 2121, 2134
- From Behavioral Models 2121
- From Behavioral Models, Hardware Description Languages Supported 2121
- From Behavioral Models, Software Languages Supported 2121
- From Decision Model 1935
- From Interaction Diagrams 2121, 2133
- From Legacy State Machine Diagrams 2126
- From Sequence Diagrams 2121, 2133
- From State Machine Diagrams 2121
- From State Machines 2122
- Hardware Description Languages Supported 2121
- Java Language Options 2272
- Language Options 2262
- Languages Supported, Hardware Description Languages 2131
- Languages Supported, Options 2262
- Live, Menu Option 657
- MDG Technology Language Options 2278
- PHP Language Options 2274
- Python Language Options 2273
- Supported Hardware Description Languages 2121, 2131
- Supported Languages 2080
- Supported Languages, Options 2262
- SystemC Language Options 2275
- Template, Call From Other Template 1700
- VB.NET Language Options 2275
- Verilog Language Options 2276
- VHDL Language Options 2277
- Visual Basic Language Options 2277
- XML Schema, From Abstract Class Model 2413

## Code Generation Template

- Transfer Between Models, Export 1637
- Transfer Between Models, Import 1637

## Code Language

- Create Properties As Attributes 1006
- Supported By Code Editor 2147

## Code Language Options 2251

## Code Module

- Add To MDG Technology 1552

## Code Options

- Additional 1553
- Define 1553
- XML File 1553

## Code Sections

- Synchronize 1640

## Code Template

- Code Template
  - Base Templates 1633
  - Custom Templates, Create 1701
  - Customize 1702
  - Default Templates 1704
  - Editor 1641, 1701
  - Editor, Add New Stereotyped Templates 1703
  - Editor, In MDG Development 1701
  - Framework, In SDK 1631
  - Framework, Overview 2281
  - Overview 1632
  - Syntax, Introduction 1643
  - Syntax, Literal Text 1643
  - Syntax, Macros 1646
  - Syntax, Template Substitution Macros 1647
- Code Template Syntax
  - Variable Definitions 1644
  - Variable References 1644
  - Variables 1644
- Codepage
  - Set Up For Document Report 2666
  - Set Up For RTF Report (Legacy) 2725
- Collaboration
  - Element 1367
  - Elements and Connectors, Now Communication 805
- Collaboration Diagram (Now Communication) 1259
  - Example 1260
  - Message Colors 1261
- Collaboration Use
  - Element 1368
- Collaborative Development 2255
- Collection Class
  - Additional 2268, 2269, 2272
  - Additional, Example Of Use 2260
  - Automation Interface Repository 2829
  - Example Of Use 2260
  - Set 2259
- Collections, EASL
  - Action 1690
  - Behaviors 1690
  - Classifier 1690
  - Construct 1690
  - Node 1690
  - State 1690
  - State Machine 1690
  - Transition 1690
  - Trigger 1690
  - Vertex 1690
- Color
  - Define Connector Custom Colors 619
  - Define Element Custom Colors 619
  - Diagram Background, Set In Theme 611
  - Element Fill, Set In Theme 611
  - Of Communication Messages 638
  - Color Code External Requirements 1776
  - Color Query
    - Shape Scripts 1598
  - Column
    - Change Order 2347
    - Create In Data Modeling 2345
    - Data Modeling 2338
    - Definition 2345
    - In UML Data Modeling Profile 2345
    - Introduction 2338
    - Order, Change 2345
    - Properties 2345
    - Re-order 2347
    - Sequence Entries 2345
    - Stereotyped Attribute 2345
    - Unique 2345
  - Column-Based
    - Index 2365
  - COM Interop
    - Debug .NET 2202
  - COM Object
    - .NET Garbage Collect (Exit Method) 2850
  - Combine
    - Windows In One Frame 128
  - Combined Fragment
    - Create 1290
    - Element 1287
    - Interaction Operator 1290
  - Combo Box Resize
    - In MDG Win32® User Interface Technology 1996
  - Comma Separated Value
    - Export 497, 501
    - Import 497, 503
  - Command
    - Deploy, Create 2218
    - Run, Create 2217
    - Unit Test, Create 2182
    - Unit Test, Introduction 2182
  - Command Set
    - Copy 165
    - Customize 165
    - Default 165
    - Delete 165
    - Export 165
    - Import 165
    - Save 165
    - Select 167
  - Commands

- Commands
  - Add To Toolbar 153
  - Change Icon Appearance 153
  - Customize In Toolbar 153
  - Remove From Toolbar 153
- Comment
  - On Document In Team Review 355
- Commenting Selections
  - Code Editor, Common 2157
- Common
  - Connectors 800
  - Elements 800
  - Reference Data 375
  - Relationships 800
  - Repository 375
  - System Resources 375
  - Toolbox Page 800
- Common Source Code Control (SCC)
  - Create Environment For Version Control 411
- Communication
  - Connector 1402
  - Elements and Connectors 805
  - Message 1428
  - Message, Create 1429
  - Message, Level 1430
  - Message, Properties 1429, 1430
  - Message, Sequence 1430
  - Relationship 1402
  - Toolbox Pages 805
- Communication Diagram
  - Description 1259
  - Elements And Connectors 1259
  - Example 1260
  - Labelled Associations 1259
  - Message Colors 1261
  - Numbering In 1259
  - Transformation To Sequence Diagram 2041
- Communication Message
  - Colors 638
- Communication Path
  - Connector 1400
  - Relationship 1400
- Community Site 38
  - Access To 68
- Compact
  - Project .EAP File 601
- Compare
  - Data 507
  - DDL With Database 2381
  - Diagram With Baseline 466
  - Models 507
  - Projects 507
  - Utility 457, 463
- Compare Utility
  - Options 464
  - Output 468
  - Tab 468
- Compartment
  - Block Element 2305
  - ChildElement 1604
  - Constraint 955
  - Constraint Block 2305
  - Custom, Add To Elements 1604
  - Custom, In-place Editor 1039
  - Custom, Show/Hide 845
  - DependsOn 1604
  - Element 955
  - Maintenance 955
  - Maintenance, Show In Diagram 2630
  - Part, In Block 2305
  - Port, In Block 2305
  - Property, In Block 2305
  - RealizedInterfaces 1604
  - Responsibility 955
  - Tag 955
  - Test Script, Show In Diagram 2616
  - Testing 955
- Compartment Events
  - Add-In Model 3034
  - EA\_GetCompartmentData 3035
  - EA\_QueryAvailableCompartments 3034
- Compass Point Icons
  - For Docking Windows 128
- Compiled April 17 2013 3
- Compiler Error
  - Build, Locate In Code 2221
- Complex Modeling
  - Enterprise Architect 9
- Complex Type Element
  - XML, Toolbox Icon 2401
- Component
  - Description 1194
  - Diagram 1194
  - Element 1370
  - Elements And Connectors 809, 1194
  - Example 1195
  - Model Template 755, 763
  - Toolbox Pages 809
  - View 769
- Compose
  - Connector 1401
  - Relationship 1401
- Composite
  - Diagram, In Simulation 2518

- Composite
  - Element, In Simulation 2518
  - Elements And Connectors 804
  - Elements, Copy Between Diagrams 912
  - Foreign Key 2361
  - State 1329, 1330
  - State Regions 1209
  - Toolbox Pages 804
- Composite Aggregation
  - Connector 1401
  - Relationship 1401
- Composite Diagram
  - In Simulation 2518
  - Include Internal Embedded Elements 938
  - Shape Script 1608
  - Show In Element, Shape Script 1608
- Composite Element
  - Child Diagram 936
  - Drill Down 936
  - Icon 936
  - In Simulation 2518
  - Include Internal Embedded Elements 938
  - Linked Diagram 936
  - Metaclass, Define With Special Attributes 1511
  - Switch Child Diagrams 936
- Composite Structure Diagram
  - Description 1188
  - Elements And Connectors 1188
  - Example 1189
- Compress
  - Timeline 1246
  - Transition 1246
- Computation Rule Table
  - Business Rule Modeling 1833, 1840
- Concept
  - ArchMate 1926
  - BPEL 1870
  - BPMN 1845
  - Eriksson-Penker MDG Technology 1929
  - Mind Mapping 1794
  - SoaML 2449
  - SOMF 2454
  - SPEM 1919
  - SysML 2294
- Conclusion
  - For Decision Table 1931
- Concurrent Method Calls
  - In Add-Ins 3016
- Concurrent State Region
  - In Simulation, Multi-Threaded 2517
- Concurrent Substate
  - Regions 1209
- Concurrent Versions System (CVS)
  - Create Environment For Version Control 404
  - TortoiseCVS 407
  - Verify Workspace 406
  - Version Control, Create Local Working Copy 405
- Condition
  - For Decision Table 1931
- Conditional Node
  - Simulate 2499
  - Structured Activity 1338, 1345
- Conditional Substitution
  - Field Substitution Macros, Code Template Syntax 1649
- Configuration
  - Apache Tomcat Server 2199
  - JBOSS Server 2198
  - Tomcat Server 2199
  - Tomcat Service 2199
- Configure
  - Controlled Packages With XMI 488
  - Export To XMI Stubs 488
  - Local Options 604
  - Model Validation 2596
  - Options 604
  - Package For Version Control 424
  - Packages 488
  - User Registry Settings 604
- Configure Event Subtypes
  - Project Calendar 577
- Configure Timeline Dialog
  - States Tab 1237
  - Transitions Tab 1239
- Confirm
  - Parent Element 911
- Connect
  - Elements 1109
  - Objects 1109
  - To ASA Data Repository 239
  - To Automation Interface 2805
  - To Element Feature 1110
  - To Model Via Cloud Services 262
  - To MySQL Data Repository 225
  - To Oracle 10g Data Repository, Via ODBC 255
  - To Oracle 10g Data Repository, Via OLE DB 257
  - To Oracle 11g Data Repository, Via ODBC 255
  - To Oracle 11g Data Repository, Via OLE DB 257
  - To Oracle 9i Data Repository, Via ODBC 255
  - To Oracle 9i Data Repository, Via OLE DB 257
  - To PostgreSQL Data Repository 233

- Connect
  - To Progress OpenEdge Data Repository 248
  - To Project Via Cloud Services 266
  - To SQL Server Data Repository 218
- Connect To Asset Service 283
- ConnectionPointReference
  - Create 1210
  - For State Machine 1210
  - To Entry/Exit Point 1210
- Connections
  - In Relationships Window 742
  - In Team Review 363
  - To Other Team Reviews 363
  - Window 742
- Connectivity Rule
  - Connectors 1952
  - Edge-Edge 1952
  - Edge-Junction 1952
  - Examples 1952
- Connector 1126
  - Abstraction 1391
  - Activity Edge 1415
  - Add Between Locked Elements 388
  - Add Note 1111
  - Add To Diagram 55
  - Add To UML Model, Quick Start 55
  - Add, Automation Interface Code Example 2998
  - Advanced Settings 1133
  - Advanced, Menu Section 1105
  - Aggregate 1392
  - Analysis Diagram 1801
  - Appearance 1107
  - Appearance Options On Diagram 831
  - ArcGIS 1946
  - Arrange 1112
  - Assembly 1393
  - Associate 1393
  - Association 1393
  - Association Class 1398
  - Asynchronous Signal Message 1427
  - At Page Boundaries 1114
  - Automation Interface, Connector Package 2933
  - Bend At Cursor 1114
  - Bend Connector 55
  - Binding 1129
  - Break Connection To Feature 1110
  - Change Source Or Target 1113
  - Change Type 1113
  - Characteristics, Edit 1105
  - Collaboration Numbers, Show 831
  - Communication 1402
  - Communication Path 1400
  - Compose 1401
  - Composite Aggregation 1401
  - Connect To Element Feature 1110
  - Connectivity Rule, ArcGIS 1952
  - Connector 1402
  - Constraints 1128
  - Context Menu 1102
  - Control Flow 1403
  - Copy Between Instances Of Elements From Project Browser 1118
  - Create Between Elements 1118
  - Create From Project Browser 1118
  - Create From Toolbox 792
  - Create In MDA-Style Transformation 2060, 2063
  - Create With Element Using Quick Linker 897
  - Create With Quick Linker 898
  - Create, Same Type As Previous 93
  - Crossed 1114
  - Custom Properties, Set Value 1105
  - Data Flow 1797
  - Define Custom Line Color 619
  - Delegate 1404
  - Delete 1119
  - Delete From Model 62
  - Dependency 1404
  - Dependency, Apply Stereotype 1405
  - Deployment 1406
  - Destination Role 1132
  - Details 1126
  - Direction Indicator 866
  - Display Options 635
  - Duplication In Transformation 2063
  - Edge 1114
  - Entity Relationship Diagram 1939
  - Extend 1406
  - Extension (Profile Toolbox) 810
  - Field Substitution Macros 1660
  - Filter Display On Diagram 718, 720
  - Filter For Hidden Or Visible 720
  - Generalization 1409
  - Generalize 1409
  - Generalize (Profile Toolbox) 810
  - Get Project Custom Colors 621
  - GML 1984
  - Hide 143, 1119
  - Hide On Diagram 62
  - Hide Type On Diagram 718, 720
  - Hide/Show 1107
  - Hide/Show On Diagram 831
  - Highlight Incoming 1109
  - Highlight Outgoing 1109

- Connector 1126
  - Implements 1440
  - Include 1410
  - Incoming Point Shapescrypt 1614
  - Information Flow 1410, 1411
  - Inheritance 1409
  - In-place Editor Options 1123
  - Insert New From Toolbar 140
  - Interrupt Flow 1415
  - Invoke 801
  - Jumps 1114
  - Labels 866
  - Labels, Edit 1123
  - Labels, Hide/Show On Diagram 831
  - Line Color 616
  - Line Width, Set In Theme 611
  - List For Requirements 1779
  - List, On Context References Tab 987
  - Locked 388
  - Manifest 1416
  - Message 1417
  - Move 57, 1112
  - Multiplicity 1130
  - Nesting 1434
  - Non-Navigable Ends, Show 831
  - Notation, Set 831
  - Notelink 1435
  - Notes 1142
  - Object Flow 1435
  - Occurrence 1438
  - Off Page 1114
  - Off-Page 1102
  - Outgoing Point Shapescrypt 1614
  - Overview 1389
  - Package Import 1438
  - Package Merge 1439
  - Page Boundary 1114
  - Pin Ends 143
  - Pkg Import 1438
  - Pkg Merge 1439
  - Precede 801
  - Properties 1126
  - Properties In Transformation 2060
  - Properties Menu 1103
  - Realization, Common Diagram 1779
  - Realization, Quick Generation Of 1779
  - Realize 1440
  - Recursion 1441
  - Relationship 1402
  - Relationship Matrix 727
  - Relationship Rule, ArcGIS 1956
  - Representation 1443
  - Represents 1442
  - Reverse Direction 143, 1123
  - Role Binding 1441
  - Role, Context Menu 1102
  - Select By Property 718
  - Self-Message 1421
  - Set Project Custom Colors 621
  - Shape Script Properties 1599
  - Show 143
  - SoaML 2451
  - Source Role Properties 1130
  - SPEM 1920
  - Stereotype Labels, Show On Diagram 831
  - Styles 1106, 1114
  - Substitute 1443
  - Substitution 1443
  - Switch To Different Element 61
  - SysML Activity 2295
  - SysML Block Definition 2300
  - SysML Interaction 2306
  - SysML Internal Block 2308
  - SysML Model 2310
  - SysML Parametric 2312
  - SysML Requirement 2314
  - SysML State Machine 2316
  - SysML Use Case 2318
  - Tagged Value (Profile Toolbox) 810
  - Tagged Value, Use 1498
  - Tagged Values 1132
  - Target Role Properties 1132
  - Tasks 1108
  - Template Binding 1444
  - To Class Created In Transformation 2063
  - Toolbar 143
  - TopologyRule, ArcGIS 1954
  - Trace 1446
  - Transform 2060, 2063
  - Transition 1446
  - Type Specific Menu Section 1103
  - Type, Change 1113
  - Types In Transformation 2060
  - Usage 1450
  - Use (Class Diagram Usage) 1450
  - Use (Use Case Connector) 1450
  - Visibility 1107, 1119, 1121
  - What Is A? 1389
  - Working With 1102
- Connector Package
  - Connector, Automation Interface 2933
  - ConnectorConstraint, Automation Interface 2939
  - ConnectorEnd, Automation Interface 2940

- Connector Package
  - ConnectorTag, Automation Interface 2943
  - RoleTag, Automation Interface 2944
  - TemplateBinding, Automation Interface 2946
- Connector Package Diagram
  - Automation Interface 2933
- Connector Styles
  - Auto Routing 1114
  - Bezier 1114
  - Custom 1114
  - Direct 1114
  - Lateral 1114
  - Line Style 1114
  - Set 1114
  - Tree Style 1114
- ConnectorConstraint
  - Automation Interface, Connector Package 2939
- ConnectorEnd
  - Automation Interface, Connector Package 2940
- ConnectorTag
  - Automation Interface, Connector Package 2943
- Console
  - Commands, Scripting Window 2796
  - Tab, Scripting Window 2796
- Constant
  - Project, For Document Reports 2664
- ConstLayoutStyles Enum
  - Automation Interface 2816
- Constraint
  - Attach To Connector 898
  - Automation Interface, Element Package 2879
  - Compartment, Element 955
  - Default, For SQL Server Table 2371
  - Delete 1004
  - Drag Onto New Element 989
  - Element 963
  - Element Border Color 616
  - Element Fill Color 616
  - Element Text Color 616
  - Field Substitution Macros 1664
  - In Connector 1128
  - In Scenarios & Requirements Window 992
  - Include In Document Report 2693
  - Inherited, Show 845
  - Internal, Import As Test 2614
  - Note, Element 1324
  - Of Attributes, Create 1004
  - Post Condition On Actions 1274
  - Precondition On Actions 1274
  - Profile 1500
  - Status Type, Define 1161
  - Stereotype 1500
  - Synchronize, And Tagged Values 1473
  - Synchronize, From MDG Toolbox Pages 1473
  - Tab, Scenario 988
  - Testpoint, Constraint Composition 2581
  - Type, Define 1160
- Constraint Block
  - Compartments 2305
- Contents
  - Table, Add To Report In MS Word 2733
- Contents Submenu
  - Package Context Menu, Project Browser 655
- Context Diagram 1797
- Context Element
  - Highlight 930
  - In Multiple Selection 930
- Context Filter
  - A Diagram 789
  - Impact On Diagram Filters 789
  - Impact On Name Filter 789
- Context Item Events
  - Add-In Model 3037
  - EA\_OnContextItemChanged 3037, 3039
  - EA\_OnContextItemDoubleClicked 3038
- Context Menu
  - Apply «Stereotype» Option 792, 1473
  - Attribute, Project Browser 667
  - Code Editor, Common 2152
  - Configure Code Editor Search Options 2152
  - Connector 1102
  - Connector Label 866
  - Connector Role 1102
  - Diagram 778
  - Diagram, Advanced 783
  - Diagram, Project Browser 665
  - Element Label 866
  - Element, Add Supporting Diagrams and Elements 939
  - Element, Multiple Selection 951
  - Element, Project Browser 660
  - Item, Structural Specification 972
  - Linked Document Editor 1048
  - Main 77
  - Method, Project Browser 667
  - Model Views 690
  - Model, Project Browser 648
  - New Element Or Connector 782
  - Operation, Project Browser 667
  - Package, Project Browser 651
  - Project Browser 648
  - Record & Analyze 2546
  - Scripting Window 2792
  - Selected Text, Structural Specification 974



- Context Menu
  - Structural Specification Entry Points 976
  - Team Review Options 346
- Context Reference
  - Add Comments, On Context References Tab 987
  - Add, On Context References Tab 987
  - Delete, On Context References Tab 987
  - List, On Context References Tab 987
  - Tab 987
- Continuation
  - Element 1332
- Control
  - Create 1999
  - Element 1999
- Control Flow
  - Connector 1403
  - Guard 1403
  - Relationship 1403
  - Weight 1403
- Control Macros
  - Code Template Syntax 1683
- Control Recording
  - Execution Analysis, Recording Sequence Diagrams 2544
- Control Stack Depth 2536
- Controlled Package 424
  - Batch Export To XMI 492
  - Batch Import From XMI 493
  - Disconnect 490
  - Load 492
  - Menu, XMI 485
  - Recovery 494
  - Report Deletion Of Cross Package References In XMI 496
  - Version Control 494
  - With XMI 484
- Converge
  - Diagram Layout 886
- Convert
  - Linked Element To Local Copy 931
  - Names In MDA Transformations 2067
- CONVERT\_DB\_TYPE 2066
- CONVERT\_NAMES
  - Macro 2067
- CONVERT\_TYPE 2066
- Convey
  - Information Item 1413
- Coordinate System
  - ArcGIS, Set 1958
  - Vertical, ArcGIS, Set 1958
- Copy
  - Attributes Between Elements 932
  - Base Project 211
  - Diagram Image To Clipboard 842
  - Diagram Image To Disk File 841
  - Diagram, Deep 842
  - Diagram, Shallow 842
  - Document Bookmark To Clipboard 656, 665
  - Element Between Packages 915
  - Element From/To Floating Diagram 837
  - Element On Diagram 836
  - Elements Between Diagrams 57, 912
  - Existing Project 212
  - Maintenance Item Between Categories 2628
  - Model 204, 206, 207
  - Operations Between Elements 932
  - Package 775
  - Packages Between Projects 506
  - Test Between Categories 2610
  - UML Diagram, Deep 842
  - UML Diagram, Shallow 842
- Copy/Paste Element Options 663
- Copy/Paste Package Options 654
- Copy/Paste Submenu
  - Element Context Menu, Project Browser 663
  - Package Context Menu, Project Browser 654
- Copyright Notice 25
- CORBA
  - MDG Technology For, Enterprise Architect 1481
- Co-Region Notation 1427
- Corporate Edition
  - Of Enterprise Architect 20
- Correct Spell Checked Words 554
- Corrupt EAP file 602
- Cover Page
  - Create For Document Reporting 2707
- Cover Sheet
  - For Document Reports 2654
  - Selecting 2654
  - Templates 2654
- Create
  - .EAP File 210
  - .FEAP file 210
  - A Project In Enterprise Architect 47
  - Activity, BPEL 1.1 1881
  - Activity, BPEL 2.0 1904
  - Adaptive Server Anywhere Repository 237
  - Add-In 3012
  - Attribute Properties 1006
  - Baselines 462
  - Boundary Element 1998
  - BPEL 1.1 Assignments 1889

## Create

- BPEL 1.1 Model 1873
- BPEL 1.1 Web Service 1890
- BPEL 2.0 Model 1897
- BPEL 2.0 Web Service Operation 1914
- BPEL 2.0 WSO 1914
- Build Script 2180
- Columns In Data Modeling 2345
- Combined Fragment 1290
- Communication Messages 1429
- Composite Foreign Key 2361
- Connector With Quick Linker 898
- Control Element 1999
- Cover Page, Document Reports 2707
- Custom Diagram Background, Image Manager 863
- Custom Tagged Values 1626
- Custom View, Add-In Model 3095
- Data Object, BPEL 2.0 1909
- Design Master, Replication 311
- Diagram 822
- Diagram From Linked Document 1096
- Diagram, Automatically 772
- Document Artifact For Linked Document 1046
- Document In Team Review 353
- Document Report (Enhanced Generator) 2642
- Document Template 2681
- Documentation 2638
- Element And Connector With Quick Linker 897
- Element From Linked Document 1096
- Element From Maintenance Item 2629
- Element From Text File 902
- Element From Text File Text 1773
- Element In Diagram 902
- Element In Specification Manager 1738
- Element Template 929
- Elements 902
- Elements On Diagram From Project Browser 833
- End Event, BPEL 1.1 1885
- End Event, BPEL 2.0 1907
- Entity 2001
- Favorites Folder, Model View 688
- Favorites Folder, Model View (Context Menu) 690
- Foreign Key 2359
- Gateway, BPEL 1.1 1884
- Gateway, BPEL 2.0 1906
- Hidden Submenu In Toolbox Profile 1564
- Intermediate Event, BPEL 1.1 1878
- Intermediate Event, BPEL 2.0 1901
- Link Between Elements 1118
- Linked Document Template 1098
- Masked Tagged Values 1626
- MDG Technologies, Overview 1545
- Model 210
- MySQL Repository 222
- Notes 923
- Oracle 10g Server Repository 253
- Oracle 11g Server Repository 253
- Oracle 9i Server Repository 253
- Pattern 1464
- Pool, BPEL 1.1 1888
- PostgreSQL Repository 229
- Primary Key 2355, 2356
- Primary Key Name Template 2355
- Profiles 1487
- Progress OpenEdge Repository 245
- Project 210
- Project File 199
- Property, BPEL 2.0 1910
- Reference Data Tagged Values 1628
- Relationship Using Matrix 737
- Relationship With Overlay 737
- Replicas 312
- Requirement From Text File Text 1773
- Requirements 1773
- Root Node, Model View 688
- Root Node, Model View (Context Menu) 690
- RTF Style Template (Legacy) 2726
- Search Definition 711
- Sequence Flow, BPEL 1.1 1887
- Sequence Flow, BPEL 2.0 1910
- Slideshow, Model View 688
- Slideshow, Model Views 695
- SQL Server Repository 217
- Start Event, BPEL 1.1 1877
- Start Event, BPEL 2.0 1900
- Structured Tagged Values 1622
- Stylesheet For Document Reports 2705
- Table in Data Modeling 2339
- Tables of Contents For Document Reports 2706
- Team Review Category 350
- Team Review Topic 351
- Template Fragment 2709
- Text 923
- Timing Diagram 1227
- Timing Message 1432
- Toolbox Profile For MDG Technology 1561
- UML Diagram 822
- UML Pattern 1464
- Views Folder, Model View 688
- Views Folder, Model View (Context Menu) 690
- Views, Model View 688

- Create
  - Views, Model View (Context Menu) 690
  - Web Report 2744
- Create A Breakpoint Or Recording Marker Option 2152
- Create Property Implementation Dialog 1006
- CreateBaselineFlag Enum
  - Automation Interface 2817
- CreateModelType Enum
  - Automation Interface 2818
- CreateObject Action
  - Create Object In Simulation 2480
- CreateObjectAction 2134
- CreateOperationAction 2134
- Creation Factory
  - OSLC Service 280
- Cross Package References
  - XMI, Report Deletion 496
- Cross Reference
  - Between EMX Files 480
  - Delete 916
  - Set For Element 916
  - Trace With Traceability Window 725
  - Use In Element 916
- Crossed
  - Connectors 1114
- CRUD Matrix
  - Introduction 732
  - Toggle To Relationship Matrix 732
- CSV
  - Export 497, 501
  - Export From Relationship Matrix 738
  - Export From Rule Composer 1842
  - Export State Machine Table To 1222
  - Import 497, 503
  - Import Requirement Hierarchies 1780
  - Import Requirements 1780
  - Imported As Source For Standard Charts 2773
  - Preserve Hierarchy 500
  - Specifications 498
- CTF
  - In SDK 1631
  - Overview 2281
- Current Connector
  - Toolbar 143
- Current Element
  - Toolbar 141
- Cursor History
  - Code Editor, Common 2157
- Custom
  - Constraint Status Types, Define 1161
  - Constraint Types, Define 1160
  - Diagram 1767, 1796, 1990, 2634
  - Diagram Types 1536, 1570
  - Difficulty Types, Define 1163
  - Elements and Connectors 813
  - Priority Types, Define 1164
  - Requirement Types, Define 1166
  - Scenario Types, Define 1167
  - Status Types, Define 1159
  - Stereotypes 1461
  - Test Status Types, Define 1165
  - Toolbox Pages 813
- Custom Background
  - Create For Diagram, Image Manager 863
- Custom Color
  - Define For Connectors 619
  - Define For Elements 619
- Custom Compartment
  - Add Linked Note To 1604
  - Add To Elements 1604
  - In-place Editor 1039
- Custom Diagram
  - Description 1796
  - Elements And Connectors 1796
  - Model 1796
- Custom Docked Window
  - Add-In 3096
  - API 3096
  - Example Code 3096
- Custom Language
  - Settings, Document Report Generation 2666
- Custom Programming Language 1581
- Custom Reference
  - Delete 916
  - Set For Element 916
  - Trace With Traceability Window 725
  - Use In Element 916
- Custom Tagged Values
  - Create 1626
- Custom Template
  - Create 1701
  - Create From Project 210
  - Document Report, File Control Options 1050
  - Linked Document, File Control Options 1050
- Custom Tools
  - External Applications 158
- Custom View
  - Add-In Model 3095
- Customization
  - User Interface Options 151
- Customize
  - Code Generation Templates 1702
  - Command Sets 165

## Customize

- Commands 152, 153
- Diagram Appearance 823
- Dialog 152
- Document Language 2666
- External Tools, Pass Parameters To 160
- General Type Values 1158
- Keyboard 152
- Keyboard Shortcuts 161
- Menu Display 162
- Menus 152
- Model Search 708
- Options 104, 152
- RTF Language (Legacy) 2725
- Toolbar Option Appearance 163
- Toolbars 152, 156
- Tools 152
- Tools In Tools Menu 157
- User Interface 167
- Visibility Of Elements 922
- Window 152

## CustomProperties Collection

- Automation Interface, ElementFeatures Package 2918

## Cut And Paste

- Elements Between Diagrams 57

## Cut/Paste Element Options 663

## Cut/Paste Package Options 654

## CVS

- Create Environment For Version Control 404
- Package Version Control History, In 440
- TortoiseCVS 407
- Version Control Options 419
- Version Control, Create Local Working Copy 405

## CVS Workspace

- Verify 406

## Cyrillic

- Support, Under Unicode 43

**- D -**

## Dashed Border

- On Element 911

## Data

- Breakpoint, Set 2229
- Compare 507
- Import 380
- Integrity 597
- Integrity Check 597
- Integrity, Run SQL Patches 601
- Model Template 755, 762

## Reference, Export 376

## Data Breakpoint

- Set 2229

## Data Distribution Service

- MDG Technology For, Enterprise Architect 1481

## Data Flow

- Concepts 1797
- Connector 1797
- Context Diagram 1797
- Diagram 1797
- MDG Technology 1797
- Relationship 1797
- Toolbox Page 1797

## Data Management

- .EAP File Management 111
- Firebird Data Options 110
- Project Compare Option 110
- Project Integrity Option 110
- Project Transfer Option 110
- Submenu, Tools Menu 110

## Data Model

- Physical 2335
- To ERD Transformation, MDA-Style Transform 2024
- Transformation From Entity Relationship Diagram 2032
- Transformation To Entity Relationship Diagram 2024

## Data Modeling

- Change Table Owner 2349
- Check Constraint, Create 2370
- Compare DDL With Database 2381
- Create Columns 2345
- Create Data Model Diagram 2336
- Create Table 2339
- Data Model Diagram 2337
- Data Type Conversion, Table 2350
- Data Types, Introduction 2348
- Database View 2371
- Database View, Create 2372
- DBMS Conversion Procedure, Package 2349
- DBMS Data Types 2350
- DDL, Generate 2380
- Elements And Connectors 818
- Foreign Keys 2358
- Generate DDL 2380
- Generate DDL For A Package 2381
- Index, Create 2365
- Introduction 1937
- Notation 2384
- Primary Key Extended Properties 2358

- Data Modeling
  - Primary Key, Create 2355
  - Profile (UML) 1937
  - Set Database Type 2341
  - Set MySQL Table Type 2342
  - Set Oracle Table Properties 2343
  - Set Schema Owner 2342
  - Set Table Owner 2342
  - Set Table Properties 2340
  - Stored Procedure 2364
  - Tables And Columns 2338
  - Toolbox Page 818
  - Trigger 2368
  - Trigger, Create 2369
  - Typical Tasks 1937
- Data Object
  - Create In BPEL 2.0 1909
  - Model In BPEL 2.0 1909
- Data Repository
  - Adaptive Server Anywhere, Connect To 239
  - Connect To 199
  - MySQL, Connect To 225
  - Oracle 10g, Connect To, Via ODBC 255
  - Oracle 10g, Connect To, Via OLE DB 257
  - Oracle 11g, Connect To, Via ODBC 255
  - Oracle 11g, Connect To, Via OLE DB 257
  - Oracle 9i, Connect To, Via ODBC 255
  - Oracle 9i, Connect To, Via OLE DB 257
  - PostgreSQL, Connect To 233
  - Progress OpenEdge, Connect To 248
  - SQL Server, Connect To 218
- Data Source
  - Select 2378
- Data Store
  - Element (Data Flow Diagram) 1797
- Data Transfer
  - Between Repositories 504
  - Compare Projects 507
  - Copy Packages Between Projects 506
  - Transfer Project Data 504
- Data Type
  - Add 1171, 2350
  - Code 1171
  - Conversion, Table 2350
  - Data Modeling, Introduction 2348
  - Definition 483
  - Delete 1171, 2350
  - Element 1370
  - Extend 1171
  - For MySQL 2352
  - Instance 1370
  - Map Between DBMSs 2350, 2353
  - Modify 1171
  - Programming Language 1171
  - Referenced 1370
  - Set For DBMS 2350
- Database
  - Compare Package DDL With 2381
  - Default 2250
  - Design 1937
  - Keys 1937
  - Model Template 755, 762
  - Modeling 1937
  - Schema, Import Of 1937
  - Set Default, Code Generation Toolbar 144
  - Supported Types 1937, 2375
  - View 2372
  - View, What Is A 2371
- Database Administrator
  - And Enterprise Architect 194
  - Project Role 194
- Database Engineering
  - Change Table Owner 2347
  - Index Foreign Keys 2347
  - Introduction 2334
  - Physical Data Model 2335
  - Submenu (Tools Menu) 107
- Database Keys
  - Association Notation 2354
  - Introduction 2354
- Database Modeling
  - Enterprise Architect 9
- Database Object
  - Select From ODBC Data Source 2378
- Database Operation Properties
  - Dialog 2358
- Database Repository
  - Access Permissions For 214
  - Connect To 214
  - Create 214
  - List Of 214
  - Set Up 214
- Database Schema
  - Description 1943
  - Diagram 1943
  - Elements And Connectors 1943
  - Example 1943
- Database Table
  - Select From ODBC Data Source 2378
- Database Type
  - Set For Table 2341
- Database View
  - Create 2372
  - Definition 2372

- Database View
  - Example 2372
  - Report On 2372
- Datastore
  - Element, Activity Diagram 1293
- Datatype
  - Automation Interface Repository 2831
- Datatypes
  - For Oracle 2352
- DBMS
  - Conversion, For A Table 2350
  - Data Type Conversion, Table 2350
  - Map Data Types Between 2353
  - Set As Default For Model 2350
  - Set Datatypes For 2350
  - Supported Types 2375
- DBMS Conversion
  - Mapper 2349
  - Procedure 2349
  - Table Conversion Between DBMS Types 2349
  - Table Owner, Change 2349
- DBMS Settings
  - Change Table Owner 2347
  - Index Foreign Keys 2347
  - Manage 2347
- DDL
  - Compare With Database 2381
  - Data Modeling 2380
  - Default Script Editor 2250
  - Generate 2380
  - Generate For Package 2381
  - Generate For Table 2380
  - Import Schema From ODBC 2376
  - Schema, Import From ODBC 2376
  - Scripts And Generated Tables 1937
  - Transformation 2025
- DDS
  - MDG Technology For, Enterprise Architect 1481
- Debug
  - .NET 2186, 2200
  - .NET CLR Versions 2201
  - .NET With COM Interop Process 2202
  - Another .NET Process 2243
  - Application, Analyzer Menu Option 112
  - ASP .NET 2203
  - Break On Variable Changing Value 2229
  - C, General Setup 2189
  - C++, General Setup 2189
  - Code 2172
  - Create Sequence Diagram, Call Stack 2241
  - Deploy Script, Create New 2218
  - Enterprise Architect 9
  - Execution Options, Toolbar 146
  - File Search, Use 2164
  - Inspect Process Memory 2242
  - Intermediary Language, MDA Transformation 2053
  - Java 2186, 2191
  - Java Applets In Internet Browsers 2194
  - Java Web Servers 2195, 2199
  - Java, Advanced Techniques 2194
  - Java, General Setup 2191
  - Microsoft Native Applications 2189
  - On Windows 7 And Windows Vista 2187
  - Platforms 2186
  - Process First Chance Exceptions 2244
  - Run Script, Create New 2217
  - Script Debugging 2801
  - Script Search 2164
  - Search Window 2164
  - Show Loaded Modules 2243
  - Show Output 2231
  - Start & Stop Options 114
  - Step Into Function Calls 2231
  - Step Options 114
  - Step Out Of Functions 2231
  - Step Over Lines Of Code 2231
  - Step Through Function Calls 2548
  - Submenu, Analyzer 114
  - Toolbar 146
  - Tooltips In Code Editor 2236
  - Trace Variable Changes 2230
  - Tracepoint Statement Output 2215
  - Under Windows Vista 2186
  - Unit Test Script, Create 2182
  - Unit Test Script, Introduction 2182
  - View Call Stack 2239
  - View Local Variables 2234
  - View Local Variables, Long Values 2235
  - View Variables In Other Scopes 2237
  - Visual Basic, General Setup 2189
  - WINE Applications 2188
- Debug Session
  - Debug C++ 2189
  - Java, Attach To VM 2194
  - Microsoft Native Setup 2189
  - Set Up 2185
  - Set Up For .NET 2200
  - Set Up For Microsoft Native 2189
- Debug Symbols
  - Debug C++ 2191
  - Microsoft Native 2191
- Debugger 2216

- Debugger 2216
  - Actions 2222
  - Android 2210
  - Before Starting 2222
  - Compare Variable Snapshots 2236
  - Debug Another Process 2243
  - Detach From Process 2243
  - Execution Options, Toolbar 146
  - Frameworks 2222
  - GDB 2209
  - GNU 2209
  - Introduction 2222
  - Java Debug Wire Protocol 2213
  - JDWP 2213
  - Modules Window 2243
  - On Windows 7 And Windows Vista 2187
  - Overview 2222
  - PHP, Checklist 2207
  - PHP, Scenarios 2204
  - PHP, System Requirements 2206
  - PHP, Timeouts 2207
  - Process 2222
  - Process First Chance Exceptions 2244
  - Start 2231
  - Stop 2231
  - Switch To From Profiler 2558
  - Switch To Profiler 2231
  - System Requirements 2186
  - Toolbar 146
  - View Elements Of Array 2238
- Debugging
  - Breakpoint Properties 2226
- Debugging Actions 2231
- Decision
  - Element 1294
- Decision Model
  - Code Generation From 1935
  - Example 1931
  - Introduction 1931
  - Manage Decision Table 1931
  - Overview 1931
- Decision Table
  - Allowable Values 1931
  - Business Rule Modeling 1833, 1836
  - Code Generation From 1935
  - Completeness 1931
  - Conclusions, Manage 1931
  - Conditions, Manage 1931
  - Editor 1931
  - Example 1931
  - Hit Policy 1931
  - Introduction 1931
  - Overview 1931
  - Rule Action Section 1833, 1837
  - Rule Binding Section 1833, 1839
  - Rule Condition Section 1833, 1836
  - Table Aggregation 1931
- Deep Copy
  - Of Diagram 842
- Deep History 1311
  - Change From Shallow 944
- Default
  - Code Language, Set 144
  - Database 2250
  - Database, Set 144
  - DDL Script Editor 2250
  - Hours 588
  - Model Diagram, Cancel (Menu Option) 95
  - Model Diagram, Set (Menu Option) 95
  - Project Browser Behavior 672
  - Templates 1704
  - User Diagram, Cancel (Menu Option) 95
  - User Diagram, Set (Menu Option) 95
- Default Appearance
  - Background Color 927
  - Border Color 927
  - Border Thickness 927
  - Font Color 927
  - Of An Element 948
  - Of Element 927
  - Set For Profile Stereotype Objects 1502
- Default Constraint 2371
- Default Diagram
  - Model, Cancel (Menu Option) 95
  - Model, Cancel For 844
  - Model, Set (Menu Option) 95
  - Model, Set For 844
  - User, Cancel (Menu Option) 95
  - User, Set (Menu Option) 95
- Default Fonts
  - Model 624
  - Set 624
  - User 624
- Default Hours
  - Estimation 588
  - Per Adjusted Use Case Point 588
  - Project Management 588
  - Rate 588
  - Settings 588
- Default Object Variable
  - In Simulation 2479
- Default Templates
  - Override in Code Template Editor 1704
- Default Toolbox

- Default Toolbox
  - Override In Profile 1567
- Default Tools Toolbar 135
- Defect
  - Create From Test Item 2615
  - Items, Element Maintenance 2623
- Defect (Issue)
  - Add 2631
  - Element 2631
  - Hide Stereotype Letter 2631
  - Show Stereotype Letter 2631
- Defects Report
  - On Elements, Generate 2740
- Define
  - Author 1153
  - Browser Behavior 605
  - Clients 1157
  - Clipboard Image File Format 605
  - Email Exchange Server 605
  - Export/Import Scripts For MDG Technology 1578
  - File Directory 605
  - Foreign Key Index Template 2363
  - Foreign Key Name Template 2362
  - Home Web Site 605
  - Internet Search Engine 605
  - Primary Key Name Template 2357
  - Resources 1156
  - Roles 1155
  - Run-Time Variable 1380
  - Stereotype As Metatype 1508
  - Stereotype Constraints 1500
  - Testpoint Constraint 2581
  - Validation Configuration For MDG Technology 1576
- Define Custom Colors 619
- Define Menu Items
  - Create Add-In 3013
- Defined Environment Types 586
- Delegate
  - Connector 1404
  - Relationship 1404
- Delete
  - Category In Team Review 343
  - Comment (Reply) In Team Review 343
  - Connector From Model 62
  - Connectors 1119
  - Connectors, Quick Start 64
  - Diagram (Single) From Project Browser 665
  - Diagrams (Multiple) From Project Browser 665
  - Diagrams From Project browser 839
  - Diagrams, Quick Start 64
  - Document In Team Review 343
  - Document Template 2681
  - Element Changes 2625
  - Element Defects 2625
  - Element From Diagram 921
  - Element from Model 921
  - Element From Project Browser 921
  - Element In Specification Manager 1739
  - Element Issues 2625
  - Element Tasks 2625
  - Elements, Impact of Auditing 456
  - Elements, Quick Start 64
  - Instance Variable 1381
  - Item In Team Review 343
  - Line Points 1114
  - Linked Document 1097
  - Linked Document Template 1098
  - Locks 332
  - Package Attributes From Model Document 2676
  - Package In Project Browser 777
  - Packages From Model Document 2676
  - Packages, Quick Start 64
  - Project Task 527
  - Relationship 964
  - Relationship Using Matrix 737
  - Team Review Category 346
  - Team Review Document 346
  - Team Review Resource 346
  - Team Review Topic 346
  - Topic In Team Review 343
  - Views 771
- Delete Selected Element(s)
  - Menu Option (Edit Menu) 80
- Deletion
  - And Auditing 456
- Delphi
  - Code Generation 2271
  - Import, Reverse Engineering 2139
  - Language Options 2271
  - Limitations 2272
  - Modeling Conventions 2095
  - Properties 2272
- Demonstration
  - Of Enterprise Architect 67
- Dependency
  - Connector 1404
  - Details Report, Generate 2739
  - Relationship 1404
  - Relationship, Apply Stereotype 1405
- Dependency Report
  - In Traceability 724



- Deploy
  - Add-In 3014
  - Application, Analyzer Menu Option 112
  - MDG Technology From Add-In 1580
  - MDG Technology From File 1580
- Deploy Command
  - Create 2218
- Deploy Script
  - Create 2218
  - Execute, Build Toolbar Option 147
- Deployment
  - Connector 1406
  - Diagram 1191
  - Elements and Connectors 809
  - Model Template 755, 764
  - Relationship 1406
  - Toolbox Pages 809
  - View 769
- Deployment and Rollout
  - And Enterprise Architect 192
  - Project Role 192
- Deployment Diagram
  - Description 1191
  - Elements And Connectors 1191
  - Example 1193
- Deployment Spec
  - Element 1371
- Derived
  - Attribute 1001
  - Symbol 1001
- Design
  - Patterns 1464
- Design Document Templates 2684
- Design Master 310, 600
  - Create, Replication 311
- Design Systems Using UML
  - Enterprise Architect 9
- Designate Driving Triggers
  - Model State Machine For HDL 2131
- Desktop Edition
  - Of Enterprise Architect 20
- Desktop Tools
  - Add 157
  - Customize 157
- Destination Role 1132
- DestroyObject Action
  - Destroy Object In Simulation 2483
  - Simulation Object Action 2483
- DestroyObjectAction 2134
- Developer
  - And Enterprise Architect 188
  - Forward Engineering 188
  - Project Role 188
  - Reverse Engineering 188
  - Round-Trip Engineering 188
  - Visualise Package Arrangement 188
- Device
  - Element 1372
- Diagram
  - Activity, Description 1199
  - Activity, Generate From Scenarios 978
  - Add And Manage, Automation Interface Code Example 2999
  - Add Elements Via Context Menu 782
  - Add Link To Team Review Post 359
  - Add Profile Feature 1472
  - Add Profile Object 1472
  - Add To Project 822
  - Add To UML Model, Quick Start 50
  - Advanced Context Menu 783
  - Advanced Menu 95
  - Alias 825
  - Alternative Image For Element 860
  - Analysis 1801
  - Anti-Aliased Rendering 622
  - Anti-Aliased Text 622
  - Appearance Options 622
  - Appearance Options, Connectors 831
  - Appearance Options, Diagram Tab 825
  - Appearance Options, Element 828
  - Appearance Options, Features 830
  - Appearance Options, General 824
  - Appearance Options, Set 823
  - Appearance Options, Visible Class Members 831
  - ArchiMate 1926
  - Attribute Details, Show 830
  - Auto Route Layout 890
  - Automatic Layout 891
  - Automatic Save 625
  - Automation Interface, Diagram Package 2948
  - Background Color Gradient 622
  - Background Color Gradient, Set 613
  - Background Color, Set 613
  - Background Tile Design 613, 622
  - Behavior Options 625
  - Behavioral, Overview 1198
  - Box Layout 879
  - BPEL 1870
  - BPMN 1845
  - Business Interaction 1805
  - Business Modeling 1805
  - Business Process 758
  - Cancel Model Default 844

## Diagram

- Center-Focussed Layout 875
- Chain Layouts 894
- Change Type 840
- Change Type (Menu Option) 95
- Circular Layout 875
- Class 761, 1184
- Class Features, Visibility 608
- Close 784
- Collaboration (Now Communication) 1259
- Communication 1259
- Compare With Baseline 466
- Component 763, 1194
- Composite Structure 1188
- Connector Appearance Options 831
- Connector Notation, Show 831
- Context 1797
- Context Filter 789
- Context Menu 778
- Context Menu, Project Browser 665
- Converge Layout 886
- Copy Elements Between Diagrams 912
- Copy Image 93
- Copy, Deep 842
- Copy, Shallow 842
- Create 822
- Create Automatically 772
- Create Custom Background, Image Manager 863
- Create From Linked Document 1096
- Create Using Image Library 864
- Create, MOF 2458
- Creator 824
- Custom 1767, 1796, 1990, 2634
- Customize Appearance 823
- Data Flow 1797
- Data Model, Create 2336
- Data Model, Example 2337
- Database Schema 762, 1943
- Define Child Type, Special Attributes 1512
- Delete (Multiple) From Project Browser 839
- Delete (Single) From Project Browser 839
- Delete Element From 921
- Delete From Project Browser 665
- Delete Multiple Elements From 921
- Deployment 1191
- Details Note 825, 848
- Diagram 1800, 1999, 2000
- Diagram Notes, Show/Hide 608
- Diagram To Package 50
- Digraph Layout 883
- Display Multiple 121
- Display Options 608
- Display, Configure 825
- Diverge Layout 886
- Divide Between Pages In Document Report 825
- Document Report Options 825, 2667
- Drag & Drop Elements From Project Browser 833
- Drag Existing Package Onto 776
- Duplicate 842
- Element Appearance Options 828
- Element Compartments, Show/Hide 828
- Element Icons 954
- Element Stereotypes, Show 828
- Element, Copy 836
- Element, Paste 836
- Elliptical Layout 875
- Entity Relationship 1939
- Eriksson-Penker 1929
- Exclusive Edit Lock 318
- Extended 1181
- Extended UML 1793
- Fan Relations Layout 888
- Feature Return Types, Show 830
- Feature Stereotypes, Show 830
- Features Appearance Options 830
- Filter Display Of Connectors 718, 720
- Filter Display Of Elements 718, 720
- Find In Project Browser 93
- Find Related Elements 838
- Float As Window 121
- Float Diagrams 790
- Frame 608
- Frame Element 1296
- Frame, Border 1296
- Gantt Chart, Always Display As 825
- Generate From Scenarios 976
- Grid Size, Adjust 625
- Hand Drawn Mode 825
- Hyperlink To 2005
- Hyperlink, From Frame 1296
- Increase Display Size 622, 778
- Interaction 1225, 1249, 1259, 1262
- Interaction Overview 1262
- Layout Options 891
- Layout Tools 874
- Layout, Move Sections 869
- Layouts, Chain 894
- Legend 849
- Lock 336
- Lock, General (Menu Option) 95
- Lock, Require User Lock 339
- Lock, Security Off 872

## Diagram

- Lock, User/Group Locking 336
- Logical 1184
- Maintenance 2634
- Make All Elements Selectable 778
- Make Model Default 844
- Manage Display 784
- Managing 820
- MDG Technology 1181
- Menu 93
- Mind Mapping 1794
- Model Default 95
- Modeling With 778
- Modify Z-Order, Of Elements 778
- MOF Example 2458
- Move Elements 911
- Move Elements Between 59
- Move Elements Within 59
- Move Sections 869
- Move, Impact On Element 57, 1294, 1305, 1306, 1313
- Navigation And Selection Hotkeys 841
- Neaten Layout 885
- Note 848
- Notes 824
- Object 1186
- Object Run State 2591
- Open From Shortcut 204, 207
- Open From Shortcut (Direct Definition) 206
- Open Package In 845
- Overview 1181
- OWL Definition 1979
- OWL Facts 1979
- Package 1182
- Page Setup 608
- Pan And Zoom 698
- Parametric, SysML 2320
- Paste 842
- Per Page Layout 881
- Place Related Elements On Current 838
- Populate 784
- Print From Project Browser 665
- Print Page Footer 825
- Print Page Header 825
- Profile Attributes, PData and StyleEx 1573
- Profiles 1536, 1570
- Properties (Diagram Menu) 93
- Properties Dialog - Connectors Tab 831
- Properties Dialog - Diagram Tab 825
- Properties Dialog - Elements Tab 828
- Properties Dialog - Features Tab 830
- Properties Dialog - General Tab 824
- Properties Dialog, Visible Class Members, 831
- Properties Note, Add 848
- Properties, Set 823
- Property Strings, Show 831
- Qualifiers, Show 830
- RDF Definition 1979
- Redo Last Action 873
- Reference, As Frame 1296
- Relationship Traceability For 725
- Relationships, Show 831
- Rename 839
- Report On 2741
- Report On Elements Linked From Other Packages 2667
- Requirements 758, 1767
- Robustness 1249, 1259, 1800, 1801, 1997, 1999, 2000
- Robustness, Generate From Scenarios 984
- Rotate Image In Document Report 825
- Rule Model 1821
- RuleFlow, Generate From Scenarios 980
- Save 93, 778
- Save As Pattern 1464
- Save As UML Pattern (Menu Option) 95
- Save Automatically 625
- Save Image Of 93
- Save Profile 1525
- Save, Default Tools Toolbar 135
- Scale View 622
- Schema Diagram 1943
- Scripts 2794
- Scroll Through, Diagram Toolbar 138
- Sequence 1249
- Sequence, Generate From Scenarios 982
- Set Page Size 870
- Show As Diagram List 778
- Show As Gantt Chart 778
- Show Diagram List As 679
- Show Realised Interfaces For Class 865
- Show/Hide Package Contents 777
- Show/Hide Page Borders 825
- Slideshow, Model View 686, 695
- SoaML 2449
- SOMF 2.1 2454
- SPEM 1920, 1924
- Spring Layout 883
- Stae Machine, Generate From Scenarios 980
- State 1203
- State Machine 1203
- Stereotype 824
- Structural, Overview 1182
- Swimlanes 852

## Diagram

- Swimlanes Matrix 854
- Synchronize With Scenario Steps 976
- SysML 2294
- SysML, Parametric 2320
- Tab Context Menu 790
- Table Owner, Show 828
- Tabs, Close 790
- Tabs, Switch Between 791
- Tasks, General 820
- Test Domain 2586
- Themes, Appearance Options 611
- Themes, Built-In 611
- Themes, User Defined 611
- Tile Design 613, 622
- Timing 1225
- Toolbar 138
- Traceability 743
- Tracking Handle Size 625
- Types 1181
- Types, Built In 1572
- Types, Custom 1536, 1570
- UML 1181
- Undo Last Action 873
- Use Case 759, 1201
- User Default 95
- User Interface 1990
- Version 824
- View 784
- View Next 869
- View Previous 869
- Views, Switch Between 791
- Visibility Indicators 830
- Visible Relations (Menu Option) 95
- Visual Check For Changes 466
- Water Mark 622
- What Is A? 1181
- Whiteboard Mode 825
- Working With 778
- WSDL Binding Element 2437
- WSDL Message Element 2430
- WSDL PortType 2433
- WSDL Service 2441
- WSDL Template 2426
- XML Schema Example 2389
- Zoom, Diagram Toolbar 138
- Z-Order Element 844

## Diagram Caption Bar

- Hide 84
- Show 84

## Diagram Filters

- Access 720

- Application 718

- Clear Effect Of 720

- Create 720

- Delete 720

- Disable 720

- Enable 720

- Introduction 718

- Set Effect Of 720

- Suggested Use 718

- Window 720

## Diagram Frame

- Appearance, Change 1296

- Element 1296

## Diagram Gate

- Element 1297

## Diagram Image

- Copy To Clipboard 842

- Copy To Disk File 841

- Save To Disk File 841

## Diagram List

- Context Menu Options 679

- Description 684

- Filter Bar 677

- Generate Report 2642

- Options 679

- Show As Diagram 679

- Show Diagram As 778

- Toolbar 679

- Value Grouping 677

- View Header 677

- Work On Elements 679

## Diagram Note

- Insert New From Toolbar 140

## Diagram Only Report

- Generate 2741

## Diagram Package

- Automation Interface 2948

- Diagram, Automation Interface 2948

- DiagramLinks, Automation Interface 2955

- DiagramObject, Automation Interface 2956

- Swimlane, Automation Interface 2961

- SwimlaneDef, Automation Interface 2958, 2960

## Diagram Toolbox 792

- Activity Pages 808

- Analysis Pages 812

- ArcGIS 1946

- Class Pages 803

- Common Page 800

- Communication Pages 805

- Component Pages 809

- Composite Pages 804

- Custom Pages 813

- Diagram Toolbox 792
  - Data Modeling Page 818
  - Deployment Pages 809
  - GML 1984
  - Interaction Pages 806
  - Maintenance Pages 815
  - Metamodel Pages 812
  - Object Pages 804
  - ODM Pages 1972
  - Profile Pages 810
  - Requirement Pages 814
  - Shortcut Menu 799
  - State (Machine) Pages 807
  - SystemC Group 2100
  - Test Domain Page 819
  - Timing Pages 806
  - Use Case Pages 801
  - User Interface Pages 816
  - Verilog Group 2105
  - VHDL Group 2106
  - WSDL Page 817
  - XML Schema Pages 818
- Diagram Type
  - Add To MDG Technology 1550
- DiagramLinks
  - Automation Interface, Diagram Package 2955
- DiagramObject
  - Automation Interface, Diagram Package 2956
- Dialog
  - All User Permissions 328
  - Application Look 168
  - Attach To Process 2243
  - Audit Settings 448
  - Baseline Diagram Compare 466
  - Class Markup Selection 2542
  - Create Property Implementation 1006
  - Database Operation Properties 2358
  - Difference 451
  - Find And Replace 2161
  - General Types 1158
  - Group Properties (Script) 2794
  - Import Package 431
  - Link Note To Element Feature 924
  - Maintenance 1169
  - Manage Uses Context List 974
  - Model Message 570
  - New Action 1268
  - Options 604
  - Package Control Options 488
  - Paste Element 833
  - People 1152
  - Project Issues 528
  - Publish As HTML 2744
  - Qualifiers 1396
  - Recent Post Options 690
  - Select <Item> 994
  - Select Attribute Type 1001
  - Select Property 996
  - Set Attribute 998
  - Set Feature 998
  - Set Operation 998
  - Slideshow Properties 695
  - String Viewer 2235
  - Swimlane Details 853
  - Testpoint Editor 2584
  - UML Types 1146
  - Version Control Settings 415
- DIB Data Access Violation 2188
- Dictionary
  - User (Spell Checker) 554
- Diff Utility 457, 463
- Differencing
  - Facility 463
  - Output 468
  - With Baselines 457, 463
- Difficulty Field
  - Value Type, Define 1163
- Digraph
  - Diagram Layout 883
- Direct Substitution
  - Field Substitution Macros, Code Template Syntax 1649
- Direct Substitution Macros 1643
- Direction Indicator
  - Connector Label 866
- Directory Structure
  - Import, Reverse Engineering 2142
- Disable
  - Add-Ins 3018
  - ArchiMate 1926
  - BPEL 1870
  - BPMN 1845
  - Data Flow Diagrams 1797
  - Entity Relationship Diagram 1939
  - Eriksson-Penker MDG Technology 1929
  - Gang Of Four Pattern Technology 2285
  - GoF Pattern Technology 2285
  - ICONIX 2282
  - MDG Technologies 1477
  - Mind Mapping 1794
  - Security 318
  - SoaML 2449
  - SOMF 2454
  - SPEM 1919

- Disable
  - SysML 2294
- Disable Recording Markers 2224
- Disconnect
  - Controlled Package 490
- Discussion
  - Element 365
- Discussion Forum
  - Now Team Review 343
- Display
  - Attributes, Inherited 845
  - Connector Properties, Shape Scripts 1599
  - Constraints, Inherited 845
  - Element Properties, Shape Scripts 1599
  - Full Screen 121
  - Inherited Attributes 1007
  - Inherited Operation 1024
  - Operations, Inherited 845
  - Requirements, Inherited 845
  - Tagged Values, Inherited 845
- Display Options
  - Connector 635
  - Diagram 608
  - Element 631
  - Link 635
  - Relationship 635
- Display Size
  - Diagram, Increase 622
- Distributed Development
  - Replication 309
  - XMI Import/Export 309
- Diverge
  - Diagram Layout 886
- Dock Windows
  - Autohide 132
  - Navigation Compass 128
- Dockable
  - Windows 125
- Docked Window
  - Custom, API 3096
- Docked Windows
  - For Requirements 1782
- Document
  - Add To Team Review 353
  - Artifact 1358
  - Comment On In Team Review 355
  - Create In Team Review 353
  - Enterprise Architect Content 2638
  - Exclude Packages 2638
  - Generation 2640
  - Linked 1044
  - Projects 2638
  - Reports 2638, 2640
  - Resource, Document Generator (Enhanced) 2668
  - Single Element, RTF (Legacy) 2721
  - Team Review, Create Comment 355
  - Team Review, Delete 346
  - Team Review, Password Protect 357
  - Web 2638
  - WSDL Element 2443
- Document Artifact
  - Containing Quick Linker Definition 1523
  - Content In Document Report 2694
  - Create For Linked Document 1046
  - Creating Learning Center Pages 74
  - Element 1373
  - Element, For Linked Document 1044
  - Generate Document Report To 2680
  - Render In Document Report 1044
- Document Editor
  - Auto Spell 1089
  - Bookmarks 1083
  - Character Formatting 1062
  - Character Styles 1062
  - Copy Text 1059
  - Create Columns 1069
  - Create Sections 1069
  - Delete Line 1059
  - Display Options 1052
  - Drawing Objects 1085
  - Edit Picture 1085
  - Endnotes 1071
  - Export Report As Document 1050
  - File Control Options 1050
  - Fonts 1062
  - Footers 1071
  - Force Column Breaks 1069
  - Force Page Breaks 1069
  - Frames 1085
  - Headers 1071
  - Highlight Text 1059
  - Hyperlinks 1083
  - IME Option 1054
  - Import Document 1050
  - Insert File 1085
  - Insert Images 1085
  - Insert Objects 1085
  - Insert Project Constants 1054
  - Insert Report Constants 1054
  - Insert Table 1075
  - Insert Text Boxes 1085
  - Insert Watermark Picture (Templates) 1085
  - Level Numbering 1054

- Document Editor
  - List Formats 1064
  - List Symbols 1064
  - Lists And Overrides 1054
  - Lock Against Protection 1093
  - Move Text 1059
  - Numbered Lists 1064
  - Paragraph Formatting 1064
  - Paragraph Numbering 1064
  - Paragraph Styles 1064
  - Paste External Objects 1059
  - Picture Embed 1085
  - Picture Frame 1085
  - Picture Link 1085
  - Print Options 1094
  - Protect Document 1093
  - Protect Text 1093
  - Repagination 1069
  - Revert To Previous Copy 1050
  - Save File 1050
  - Search and Replace Options 1059
  - Section Numbering, User Defined 1081
  - Select Text 1059
  - Show Headers & Footers 1052
  - Show Hidden Text 1052
  - Special Text 1054
  - Spell Check Text 1089
  - Style Sheets 1054
  - Tab Setting 1067
  - Table Commands 1075
  - Table of Contents 1054
  - Tabular Sections 1075
  - Text Formatting 1062
  - Text Frame 1085
  - Text Scrolling 1059
  - Thesaurus 1089
  - Track Changes 1091
  - Update Styles 1050
  - Zoom 1052
- Document Generation
  - Enterprise Architect 9
- Document Generator
  - Cover Sheet Templates 2654
  - Document Options, From Diagram (Enhanced) 2667
  - Document Options, From Diagram (Legacy) 2667
  - Enhanced 2644
  - Style Sheet Templates 2653
  - Table Of Contents Templates 2652
- Document Options
  - Document Report 2654
  - Generator, From Diagram (Enhanced) 2667
  - Generator, From Diagram (Legacy) 2667
- Document Report
  - Add Table Of Contents In MS Word 2733
  - Add Table Of Figures In MS Word 2733
  - Advanced Options 2654
  - Bookmarks 2730
  - Bookmarks, In Master Document Elements 2669
  - Copy Bookmark To Clipboard 656, 665
  - Cover Sheet Templates 2654
  - Create Links In MS Word 2730
  - Custom Language Settings 2666
  - Define Structure And Content 2688
  - Document Options (Enhanced Generator) 2654
  - Element Filters 2660
  - Element-Level 2642
  - Embedded Elements, Include 2691
  - Exclude Filters 2660
  - Exclude Package 2761
  - Footers, Add In MS Word 2734
  - Formats 2644
  - Generate 2644
  - Generate (Enhanced) 2642
  - Generate From Diagram List 684
  - Generate From Package Browser 673
  - Generate To Document Artifact 2680
  - Generate To Linked Document 2680
  - Generate, Quick Start 2642
  - Headers, Add In MS Word 2734
  - Identify External Elements 2695
  - In MS Open Office 2654
  - Include Constraints 2693
  - Include Package 2761
  - Include Relationship Matrix Profile 2692
  - Include Scenarios 2693
  - Include Tagged Values 2698
  - Linked Document, Include 2694
  - Open In Microsoft Word 2729
  - Open Office Display 2654
  - Other 2736
  - Other Filters 2663
  - Project Constants, User Defined 2664
  - Save As Defined Document (Enhanced) 2668
  - Style Sheet Templates 2653
  - Switch Generator 2654
  - Table Of Contents Templates 2652
  - Tables, Apply Styles In MS Word (Legacy) 2734
  - Tables, Manipulate In MS Word (Legacy) 2734
  - Tables, Resize In MS Word (Legacy) 2734
  - Template Fragments 2708

## Document Report

- Template Fragments, Create 2709
- Template Fragments, Custom Query 2711
- Template, Design 2684
- Template, Insert Project Constants 1054
- Template, Insert Report Constants 1054
- Through Diagram List 2642
- Through Model Search 2642
- Through Package Browser 2642
- Update Links In MS Word (Legacy) 2735
- Word Substitution 2665

## Document Report Template

- Add To MDG Technology 1555
- Show Headers & Footers 1052
- Show Hidden Text 1052
- Template Design Tool Options 1052
- Zoom 1052

## Document Template

- Add Template Fragments To 2718
- Cover Pages, Create 2707
- Create 2681
- Delete 2681
- Design 2684
- Edit 2681
- Edit Fragment In 2718
- Export To Reference File 2681
- Import 2707
- Import From Reference File 2681
- List Formats, Toolbar Option 2684
- Normal.rtf 1058
- Override Fragment In 2718
- Stylesheets, Create 2705
- Switch Fragment In 2718
- System, Description 2648
- Tables of Contents, Create 2706
- View Fragment In 2718

## Document Template Design

- Print Options 1094

## Document Template Designer

- Add Content 2702
- Add Fields 2702
- Child Sections 2690
- Commands 2704
- Content Panel 2684
- Context Menu 2684
- Define Report Structure 2688
- Description 2684
- Embedded Elements Sections 2691
- Extract Model Information 2702
- Identify External Elements 2695
- Linked Document Sections 2694
- Protection of Template 1093

## Redo Edit 2704

- Relationship Matrix Profile Section 2692
- Section Markers 2684
- Sections Panel 2684
- Select Model Components 2688
- Set Sections In Report 2688
- Subsections Defined By Parent 2690
- Subsections Independent Of Parent 2690
- Tagged Value Section 2698
- Toolbar Options 2684
- Tools Menu Option 104
- Undo Edit 2704
- ValueOf Field 2698

## Document Template Editor

- Constraints Sections 2693
- Scenarios Sections 2693
- Tabular Sections 2700

## Documentation

- As Web Page 2744
- Elements 2669
- Generate On Web As Virtual Document 2759
- Generate, Project Browser Option 669
- Group, Toolbox 2669

## Documentation (Reports)

- Submenu (Project Menu) 89

## Documentation Submenu

- Package Context Menu, Project Browser 656
- Reports 656

## DocumentBreak Enum

- Automation Interface 2818

## DocumentGenerator Class

- Automation Interface 2985

## DocumentGenerator Interface

- Automation Interface 2985

## DocumentGeneratorInterface

- DocumentGenerator Class, Automation Interface 2985

## DocumentPageOrientation Enum

- Automation Interface 2818

## DocumentType Enum

- Automation Interface 2819

## DoDAF-MODAF

- MDG Technology For, Enterprise Architect 1481

## Domain

- Model Template 755, 760
- Organizational Relationships 755, 760
- Physical Units 755, 760
- Structure 755, 760

## DOORS, Telelogic

- MDG Link For, Enterprise Architect 1481

## Dot



Dot  
     On Association 1130  
 Download  
     DBMS Repository Scripts 68  
     Gang Of Four Patterns 1466  
     GoF Patterns 1466  
     Latest Version 68  
 Drag  
     Elements From Project Browser Onto Diagram 833  
     Objects From Project Browser Onto Diagram 833  
     Text Into Diagram To Create Element 1773  
 Drawing Methods  
     Shape Scripts 1591  
 Drill Down  
     Composite Element 936  
 Drop  
     Classifiers As Attributes 833  
     Classifiers As Links 833  
     Classifiers As New Instances 833  
     Elements From Project Browser Onto Diagram 833  
     Objects From Project Browser Onto Diagram 833  
 DTD  
     Validate XML Import/Export 483  
 Duplicate  
     Diagram 842  
     Element 915  
     Package 775  
     UML Diagram 842  
 Duration  
     Constraint 1424  
     Constraint Between Messages 1424  
     Observation 1424  
 Dynamic View 769  
 Dynamic Visual Filters 718

## - E -

EA\_AddinLicenseGetDescription  
     Add-In License Management Event, Add-In Model 3032  
 EA\_AddinLicenseValidate  
     Add-In License Management Event, Add-In Model 3031  
 EA\_Connect  
     Add-In Event 3022  
 EA\_Disconnect  
     Add-In Event 3023  
 EA\_FileClose  
     Broadcast Events, Add-In Model 3040  
 EA\_FileNew  
     Broadcast Events, Add-In Model 3041  
 EA\_FileOpen  
     Broadcast Events, Add-In Model 3042  
 EA\_GetCompartmentData  
     Compartment Events, Add-In Model 3035  
 EA\_GetMenuItems  
     Add-In Event 3023  
 EA\_GetMenuState  
     Add-In Event 3024  
 EA\_GetSharedAddinName  
     Add-In License Management Event, Add-In Model 3033  
 EA\_MenuClick  
     Add-In Event 3025  
 EA\_OnAttributeTagEdit  
     Tagged Value Broadcast Events, Add-In Model 3084  
 EA\_OnConnectorTagEdit  
     Tagged Value Broadcast Events, Add-In Model 3085  
 EA\_OnContextItemChanged  
     Context Item Events, Add-In Model 3037  
 EA\_OnContextItemDoubleClicked  
     Context Item Events, Add-In Model 3038  
 EA\_OnDeleteTechnology  
     Technology Events, Add-In Model 3092  
 EA\_OnElementTagEdit  
     Tagged Value Broadcast Events, Add-In Model 3086  
 EA\_OnEndValidation  
     Model Validation Broadcasts, Add-In Model 3050  
 EA\_OnImportTechnology  
     Technology Events, Add-In Model 3094  
 EA\_OnInitializeTechnologies  
     Technology Events, Add-In Model 3089  
 EA\_OnInitializeUserRules  
     Model Validation Broadcasts, Add-In Model 3048  
 EA\_OnNotifyContextItemModified  
     Context Item Events, Add-In Model 3039  
 EA\_OnOperationTagEdit  
     Tagged Value Broadcast Events, Add-In Model 3087  
 EA\_OnOutputItemClicked  
     Add-In Event 3027  
 EA\_OnOutputItemDoubleClicked  
     Add-In Event 3028  
 EA\_OnPostActivateTechnology  
     Technology Events, Add-In Model 3090  
 EA\_OnPostCloseDiagram

- |   |      |
|---|------|
| EA_OnPostCloseDiagram                     |      |
| Broadcast Events, Add-In Model            | 3042 |
| EA_OnPostInitialized                      |      |
| Broadcast Events, Add-In Model            | 3043 |
| EA_OnPostNewAttribute                     |      |
| Post-New Events, Add-In Model             | 3064 |
| EA_OnPostNewConnector                     |      |
| Post-New Events, Add-In Model             | 3062 |
| EA_OnPostNewDiagram                       |      |
| Post-New Events, Add-In Model             | 3063 |
| EA_OnPostNewDiagramObject                 |      |
| Post-New Events, Add-In Model             | 3063 |
| EA_OnPostNewElement                       |      |
| Post-New Events, Add-In Model             | 3061 |
| EA_OnPostNewGlossaryTerm                  |      |
| Post-New Events, Add-In Model             | 3067 |
| EA_OnPostNewMethod                        |      |
| Post-New Events, Add-In Model             | 3065 |
| EA_OnPostNewPackage                       |      |
| Post-New Events, Add-In Model             | 3066 |
| EA_OnPostOpenDiagram                      |      |
| Broadcast Events, Add-In Model            | 3044 |
| EA_OnPostTransform                        |      |
| Broadcast Events, Add-In Model            | 3044 |
| EA_OnPreActivateTechnology                |      |
| Technology Events, Add-In Model           | 3089 |
| EA_OnPreDeleteAttribute                   |      |
| Pre-Deletion Events, Add-In Model         | 3069 |
| EA_OnPreDeleteConnector                   |      |
| Pre-Deletion Events, Add-In Model         | 3071 |
| EA_OnPreDeleteDiagram                     |      |
| Pre-Deletion Events, Add-In Model         | 3071 |
| EA_OnPreDeleteDiagramObject               |      |
| Pre-Deletion Events, Add-In Model         | 3072 |
| EA_OnPreDeleteElement                     |      |
| Pre-Deletion Events, Add-In Model         | 3068 |
| EA_OnPreDeleteGlossaryTerm                |      |
| Pre-Deletion Events, Add-In Model         | 3074 |
| EA_OnPreDeleteMethod                      |      |
| Pre-Deletion Events, Add-In Model         | 3070 |
| EA_OnPreDeletePackage                     |      |
| Pre-Deletion Events, Add-In Model         | 3073 |
| EA_OnPreDeleteTechnology                  |      |
| Technology Events, Add-In Model           | 3091 |
| EA_OnPreDropFromTree                      |      |
| Pre-New Events, Add-In Model              | 3079 |
| EA_OnPreExitInstance                      |      |
| Broadcast Events, Add-In Model            | 3045 |
| EA_OnPreNewAttribute                      |      |
| Pre-New Events, Add-In Model              | 3080 |
| EA_OnPreNewConnector                      |      |
| Pre-New Events, Add-In Model              | 3077 |
| EA_OnPreNewDiagram                        |      |
| Pre-New Events, Add-In Model              | 3078 |
| EA_OnPreNewDiagramObject                  |      |
| Pre-New Events, Add-In Model              | 3078 |
| EA_OnPreNewElement                        |      |
| Pre-New Events, Add-In Model              | 3076 |
| EA_OnPreNewGlossaryTerm                   |      |
| Pre-New Events, Add-In Model              | 3083 |
| EA_OnPreNewMethod                         |      |
| Pre-New Events, Add-In Model              | 3081 |
| EA_OnPreNewPackage                        |      |
| Pre-New Events, Add-In Model              | 3082 |
| EA_OnRetrieveModelTemplate                |      |
| Broadcast Events, Add-In Model            | 3046 |
| EA_OnRunAttributeRule                     |      |
| Model Validation Broadcasts, Add-In Model | 3053 |
| EA_OnRunConnectorRule                     |      |
| Model Validation Broadcasts, Add-In Model | 3053 |
| EA_OnRunDiagramRule                       |      |
| Model Validation Broadcasts, Add-In Model | 3052 |
| EA_OnRunElementRule                       |      |
| Model Validation Broadcasts, Add-In Model | 3050 |
| EA_OnRunMethodRule                        |      |
| Model Validation Broadcasts, Add-In Model | 3054 |
| EA_OnRunPackageRule                       |      |
| Model Validation Broadcasts, Add-In Model | 3051 |
| EA_OnRunParameterRule                     |      |
| Model Validation Broadcasts, Add-In Model | 3055 |
| EA_OnStartValidation                      |      |
| Model Validation Broadcasts, Add-In Model | 3049 |
| EA_OnTabChanged                           |      |
| Broadcast Events, Add-In Model            | 3047 |
| EA_QueryAvailableCompartments             |      |
| Compartment Events, Add-In Model          | 3034 |
| EA_ShowHelp                               |      |
| Add-In Event                              | 3029 |
| EAB File                                  |      |
| Export                                    | 437  |
| Import                                    | 438  |
| Manually Import                           | 439  |
| Manually Locate                           | 439  |
| Model Branch File                         | 437  |
| EABase                                    |      |
| As Source                                 | 211  |
| Project                                   | 43   |

- EABase
  - Project File 210
- EAEditionTypes Enum
  - Automation Interface 2819
- EAExample Model
  - Open, Menu Option 123
  - Open, Start Page 68
- EAP File
  - As Project Database 199
  - Corrupt 602
  - Iterate Through, Automation Interface Code Example 2996
- EASL
  - Behavioral Model Templates 1688
  - Code Generation Macros, Behavioral Model 1688
  - Enterprise Architect Simulation Library 1688
- EASL Collections
  - Action 1690
  - Behavior 1690
  - Classifier 1690
  - Construct 1690
  - Node 1690
  - State 1690
  - State Machine 1690
  - Transition 1690
  - Trigger 1690
  - Vertex 1690
- EASL Properties
  - Action 1693
  - Argument 1693
  - Behavior 1693
  - Call Event 1693
  - ChangeEvent 1693
  - Classifier 1693
  - Condition 1693
  - Construct 1693
  - Edge 1693
  - EventObject 1693
  - Instance 1693
  - Parameter 1693
  - Primitive 1693
  - PropertyObject 1693
  - SignalEvent 1693
  - State 1693
  - StateMachine 1693
  - TimeEvent 1693
  - Transition 1693
  - Trigger 1693
  - Vertex 1693
- EASL\_GET
  - Code Generation Macro, Behavioral Model 1688
- EASLList
  - Code Generation Macro, Behavioral Model 1688
- ECF
  - Value 586
  - Weighting 586
- Eclipse
  - MDG Integration For, Enterprise Architect 1481
  - MDG Link For, Enterprise Architect 1481
- Edit
  - Attribute Name, In-Place Editor 1034
  - Document Template 2681
  - Element Name, In-Place Editor 1034
  - Item In Team Review 358
  - Linked Document Template 1100
  - Linked Documents 1048
  - Menu 80
  - Operation Name, In-Place Editor 1034
  - Pattern Default 1467
  - Test Details 2607
- Editions
  - Business and Software Engineering 20
  - Corporate 20
  - Desktop 20
  - Floating Licence 20
  - Lite 23
  - Model Simulation 2466
  - Of Enterprise Architect, Available 20
  - Of Enterprise Architect, Introduction 18
  - Professional 20
  - Standalone 20
  - System Engineering 20
  - Ultimate 20
- Editor
  - Decision Table 1931
  - Team Review 358
- Editor Language Properties
  - Code Editor Properties 2251
  - Global Options 2251
  - Language-Specific Options 2251
  - Macro Key Assignment 2251
- Effects
  - Javascript, In Simulation 2492
- Effort
  - Attributes 2880
  - Automation Interface, Element Package 2880
  - Field Substitution Macros 1665
  - Methods 2880
- Effort Management 518
- Effort Types 522
  - Define 522
  - Global 522
  - Non-Global 518

## EJB

- Entity Bean Transformations 2029
- Session Bean Transformations 2029

## Element 964, 1328, 1332

- Abort Edit Changes 101
- Accept Edit Changes 101
- Action 1266
- Activity 1279
- Activity Final 1305
- Activity Partition 1325
- Activity Region 1328
- Actor 1284
- Add And Manage, Automation Interface Code Example 2997
- Add Attribute, In-place Editor 1040
- Add Directly To Package 903
- Add In Specification Manager 1738
- Add Link To Team Review Post 359
- Add New Item (Inline Features Menu Option) 101
- Add Operation, In-place Editor 1040
- Add Supporting Diagrams and Elements 939
- Add Tagged Value (Option) 100
- Add To Diagram 52
- Add To Diagram From Project Browser 833
- Add To Diagram Via Context Menu 782
- Add To Favorites 944
- Add To Profile 1488, 1491, 1531
- Add To UML Model, Quick Start 52
- Advanced Settings 961
- Align 953
- Align Multiple 918, 951
- Align, Diagram Toolbar 138
- Alternative Image 860
- Analysis Diagram 1801
- Appearance, Format From Toolbar 785
- Apply Image From Clipboard 99
- ArcGIS 1946
- ArchiMate 1926
- Artifact 1358
- Associated Files 988
- Author 958
- Auto Counters, Apply 906
- Auto Counters, Set Up 904
- Auto Naming Aliases, Apply 906
- Auto Naming, Apply 906
- Auto Naming, Set Up 904
- Auto Numbering, Apply 906
- Auto Numbering, Set Up 904
- Automation Interface, Element Package 2881
- Autosize Group 851
- Autosize Single 851

- Background Color 616
- Behavioral Diagram 1265
- Boundary 1997
- Boundary, Settings 1348
- BPMN, Change Appearance 1862
- Browser Window 989
- Bulk Update 590
- Business Rule 1821
- Cardinality (Multiplicity), Non-Displayable 959
- Central Buffer Node 1285
- Change 2631, 2633
- Change Appearance, BPMN 1862
- Change Connector 61
- Change Type 100, 918
- Changes To 2623
- Changes, Add/Modify/Delete 2625
- Chart 2763
- Child Validation 2600
- Choice 1286
- Class 1363
- Collaboration 1367
- Collaboration Use 1368
- Combined Fragment 1287
- Compartments 955
- Complexity 958
- Component 1370
- Composite 936
- Concurrency 959
- Confirm As Parent 911
- Connect 1109
- Connectors In Relationships Window 742
- Constraint Note 1324
- Constraint, Attach 101
- Constraints 963
- Context Menu 939
- Context Menu, Advanced 944
- Context Menu, Find Submenu 944
- Context Menu, Project Browser 660
- Continuation 1332
- Continutaion 1332
- Control 1999
- Coordinates On Status Bar 150
- Copy And Paste Between Diagrams 57
- Copy Between Packages 915
- Copy Composite Between Diagrams 912
- Copy From/To Floating Diagram 837
- Create Child Diagram 822
- Create From Linked Document 1096
- Create From Maintenance Item 2629
- Create From Text File 902
- Create From Text File Text 1773
- Create From Toolbox 792

- Element 964, 1328, 1332
  - Create In Diagram 902
  - Create In Relationship Matrix 727
  - Create In Specification Manager 1738
  - Create Link From Project Browser 1118
  - Create Linked Document On 1047
  - Create With Quick Linker 897
  - Create, Same Type As Previous 93
  - Cross References (Option) 100
  - Custom Background Color 619
  - Custom Fill Color 619
  - Custom Line Color 619
  - Custom Text Color 619
  - Customize Visibility 922
  - Cut And Paste Between Diagrams 57
  - Dashed Border On 911
  - Data Store, Data Flow Diagram 1797
  - Data Type 1370
  - Datastore, Activity Diagram 1293
  - Decision 1294
  - Default Appearance, Named Element 99
  - Default Element Template 929
  - Defect 2631
  - Defects In 2623
  - Defects, Add/Modify/Delete 2625
  - Delete 921
  - Delete In Specification Manager 1739
  - Delete Item From 101
  - Deployment Spec 1371
  - Details 959
  - Device 1372
  - Diagram Frame 1296
  - Diagram Gate 1297
  - Discussion Facility 365
  - Discussion History 365
  - Display Depth 844
  - Display Options 631
  - Document Artifact 1373
  - Documentation 2669
  - Drag From Project Browser 833
  - Drag Properties Onto 989
  - Duplicate 915
  - Edit Attribute Keyword 1037
  - Edit Attribute Scope 1036
  - Edit Attribute Stereotype, In-place Editor 1035
  - Edit Item, Tasks 1032
  - Edit Name, In-Place Editor 1034
  - Edit Operation Parameter Keyword 1038
  - Edit Operation Parameter Kind 1039
  - Edit Operation Scope 1036
  - Edit Operation Stereotype, In-place Editor 1035
  - Embedded 100
  - Endpoint 1298
  - Entity 2000
  - Entity, Entity Relationship Diagram 1939
  - Entry Point 1300
  - Enumeration 1374
  - Eriksson-Penker 1929
  - Event 2001
  - Exception 1300
  - Execution Environment 1374
  - Exit Point 1304
  - Expansion Node 1301
  - Expansion Region 1301
  - Export Data In CSV Format 501
  - Expose Interface 1375
  - Extended By Stereotype 1997
  - External Requirements 963
  - External, Data Flow Diagram 1797
  - Feature 1769
  - Feature, Connect To 1110
  - Feature, Disconnect From 1110
  - File Artifact 1362
  - Fill Color 616
  - Fill Color Gradient 622
  - Fill Color, Set 613
  - Fill Color, Set In Theme 611
  - Fill Gradient Style, Set 613
  - Fill Gradient, Set 613
  - Filter Display On Diagram 718, 720
  - Find In Diagram Menu Options 660
  - Find in Diagrams 97, 944
  - Find In Diagrams, Package Browser 679
  - Find In Project Browser 97, 141, 944
  - Find Related 838
  - Fine Movement 869
  - Flow Final 1306
  - Font, Set 99
  - Fork 1307, 1309
  - Fragment 1287
  - Gang Of Four Pattern 2285
  - Gap, Model 745
  - General Settings 958
  - Get Project Custom Colors 621
  - GML 1984
  - GoF Pattern 2285
  - Gradient Direction, Set In Theme 611
  - Hide Type On Diagram 718, 720
  - Highlight Context 930
  - Highlight Incoming Connectors 1109
  - Highlight Outgoing Connectors 1109
  - Highlight Related On Diagram 789
  - History 1311
  - Hyperlink 2002, 2004, 2005

- 
- Element 964, 1328, 1332
    - Icon, Project Browser, User-Defined 1503
    - Icons On Diagram 954
    - Image 2006
    - Import Data In CSV Format 503
    - In Favorites 1176
    - Include External In Document Report 2695
    - Include Linked In Document Report 2667
    - Information Item 1376
    - Initial 1313
    - In-place Editor Options 1032
    - In-Place Formatting 954
    - Insert Maintenance Feature 1042
    - Insert New Feature (Inline Features Menu Option) 101
    - Insert Operation Parameter 1041
    - Insert Related Elements 933
    - Insert Testing Features 1043
    - Instance 1379
    - Interaction 1314
    - Interaction Occurrence 1317
    - InteractionUse 1317
    - Interface 1377
    - Internal Requirements 961
    - Interruptible Activity Region 1316
    - Issue 2631
    - Issues With 2623
    - Issues, Add/Modify/Delete 2625
    - Join 1307, 1310
    - Junction 1319
    - Keywords 958
    - Labels 866
    - Layout 951
    - Legend 849
    - Lifeline 1321
    - Link To Attribute Via Object 1008
    - Linked, Convert To Local Copy 931
    - Lock 336
    - Lock Indicators 340
    - Lock, Require User Lock 339
    - Lock, Security Off 939
    - Lock, User/Group Locking 336
    - Locked, Add Connector To 388
    - Maintenance Items 2623
    - Make All Selectable/Unselectable 778
    - Make Non-Selectable, Multiple Elements 951
    - Make Non-Selectable, Single Element 939
    - Make Same Size 953
    - Make Selectable, Multiple Elements 951
    - Make Selectable, Single Element 939
    - Managing 900
    - Master Document 2669
    - Match Size 951
    - Menu 97
    - Merge 1294
    - Merge Node 1322
    - Message Endpoint 1322
    - Message Label 1323
    - Mind Mapping 1794
    - Model Document 2669
    - Modeling With 900
    - Move Between Diagrams 59
    - Move Between Packages 58, 913
    - Move By Increments 869
    - Move From/To Floating Diagram 837
    - Move In Diagrams 911
    - Move Within Diagram 59
    - Move Within Package 57
    - Move, Impact Of Diagram 57
    - Multiple Selection 951
    - Multiple Update 590
    - N-Ary Association 2007
    - N-Ary, Entity Relationship Diagram 1939
    - New 902
    - Node 1378
    - Non-Selectable, Multiple Elements 951
    - Non-Selectable, Single Element 939
    - Note (Constraint, Comment) 1324
    - Note, Attach 101
    - Notes 923, 1142
    - Nudge 869, 953
    - Object 1379
    - Object Node 1325
    - Occurrence 1317
    - Operations, In Specification Manager 1740
    - Overrides & Implementations 100
    - Package 1382
    - Packaging Component 2008
    - Parents And Interfaces 100
    - Part 1383
    - Partition 1325
    - Paste As Link 82
    - Paste As New 82
    - Paste Copy 915
    - Paste From Clipboard As Metafile 82
    - Paste From Project Browser 833
    - Phase 958
    - Phase, Update For Package 590
    - Place Related On Current Diagram 838
    - Port 1384
    - Position Options 97
    - Postconditions 963
    - Preconditions 963
    - Primitive 1386

- Element 964, 1328, 1332
  - Process 1810, 2008
  - Process, Data Flow Diagram 1797
  - Properties 989
  - Properties Dialog 956
  - Properties Dialog, General Settings 958
  - Properties Options 939
  - Properties Window 992
  - Properties, As Attributes 1006
  - Properties, Edit From Package 1182
  - Properties, Links 964
  - Pseudo-State 1208
  - Receive 1327
  - Receive Event 2001
  - Region 1328
  - Region, Expansion 1301
  - Region, Interruptible Activity 1316
  - Relationship Traceability For 725
  - Relationship, Delete 964
  - Relationship, Entity Relationship Diagram 1939
  - Relationship, Hide 964
  - Relationship, Show 964
  - Requirement 1763, 1765
  - Requirements 961
  - Resize 919
  - Responsibilities 961
  - Risk 2009
  - Round Corners Of Rectangles 622
  - Rule Task 1826
  - Scenario 965
  - Scenarios & Requirements 992
  - Screen 1991
  - Select By Property 718
  - Selectable, Multiple Elements 951
  - Selectable, Single Element 939
  - Send 1328
  - Send Event 2001
  - Sequence Diagram 1253
  - Sequence, Lifecycle 1251
  - Set Alternative Image 99
  - Set Cross References 916
  - Set Custom References 916
  - Set Default Appearance 927
  - Set Element Template Package 929
  - Set Font 949
  - Set Parent 908
  - Set Project Custom Colors 621
  - Shadow Color 616
  - Shape Script Properties 1599
  - Show Maintenance Compartment 2630
  - Show Usage 910
  - Signal 1387
  - Size 919
  - SoaML 2451
  - SOMF 2454
  - Space Evenly 918
  - SPEM 1920
  - State 1329
  - State Invariant 1332, 1334
  - State Lifeline 1335
  - State Machine 1338
  - State/Continuation 1332
  - Status 958
  - Status, Update For Package 590
  - Stereotype 958
  - Stored Procedure 2364
  - Structural Diagram 1357
  - Structural, Add 935
  - Structured Activity, Conditional Node 1338, 1345
  - Structured Activity, Loop Node 1338, 1341
  - Structured Activity, Sequential Node 1338, 1341
  - Structured Activity, Structured Node 1338, 1340
  - Sub-Activity 1279
  - Sub-Activity, Conditional Node 1338, 1345
  - Sub-Activity, Loop Node 1338, 1341
  - Sub-Activity, Sequential Node 1338, 1341
  - Sub-Activity, Structured Node 1338, 1340
  - Submachine State 1329, 1338
  - Superimposition 844
  - Synch 1346
  - Synchronize Stereotyped Tagged Values From Toolbox 792
  - Synchronize Stereotypes With Profile 1473
  - SysML Activity 2295
  - SysML Block Definition 2300
  - SysML Interaction 2306
  - SysML Internal Block 2308
  - SysML Model 2310
  - SysML Parametric 2312
  - SysML Requirement 2314
  - SysML State Machine 2316
  - SysML Use Case 2318
  - System Boundary 1347
  - Table 1942
  - Task 2010
  - Tasks 900
  - Tasks On 2623
  - Tasks, Add/Modify/Delete 2625
  - Template Package 929
  - Template Parameters 959
  - Template, Default Element 929

- Element 964, 1328, 1332
  - Templates And Profiles 929, 1472
  - Terminate 1350
  - Test Case 2010
  - Test Cut 2588
  - Test Scripts Compartment, Show 2616
  - Test Set 2588
  - Test Suite 2589
  - Text 923
  - Text Color 616
  - Toolbar 140
  - Transformation 2017
  - Trigger 1350
  - Type, Change 100
  - UI Control 1992
  - UML 1265
  - Usage 910
  - Use Case 1352
  - Use Circle Notation 1355
  - Use Cross References 916
  - Use Custom References 916
  - Use Extras, Automation Interface Code Example 3001
  - Use Rectangle Notation 1355
  - User Interface 1992
  - Value Lifeline 1355
  - Version 958
  - Version, Update For Package 590
  - View Properties (Inline Features Menu Option) 101
  - Visibility 959
  - Visibility Options 631
  - Visual Representation, Introduction 953
  - Work On From Diagram List 684
  - Work On From Package Browser 673
  - Work On From Toolbar 141
  - Working With 900
  - WSDL Binding 2437
  - WSDL Message 2430
  - WSDL Namespace 2429
  - WSDL Service 2441
  - WSDL Types 2423
  - WSDL, Document 2443
  - WSDL, PortType 2433
  - XML Schema Any 2405
  - XML Schema Attribute Group 2399
  - XML Schema Complex Type 2401
  - XML Schema Global Attribute 2396
  - XML Schema Global Element 2392
  - XML Schema Group 2404
  - XML Schema Local Element 2394
  - XML Schema Package 2390
  - XML Schema Simple Type 2402
  - XML Schema, Enumeration 2412
  - XML Schema, Model Group 2410
  - XML Schema, Union 2408
  - Z-Order 844, 953
- Element Context Menu
  - Add Submenu, Project Browser 662
  - Code Engineering Submenu, Project Browser 664
  - Copy/Paste Submenu, Project Browser 663
- Element Development
  - Monitor in Kanban Chart 857
- Element Feature
  - Attribute 999
  - Connect To 1110
  - Disconnect From 1110
  - Operation 1014
- Element Filter
  - For Standard Charts 2772
- Element Package, Automation Interface
  - Constraint 2879
  - Diagram 2877
  - Effort 2880
  - Element 2881
  - File 2895
  - Issue 2896
  - Metric 2898
  - Requirement 2899
  - Resource 2900
  - Risk 2902
  - Scenario 2903
  - ScenarioExtension 2904
  - ScenarioStep 2905
  - TaggedValue 2907
  - Test 2908
- Element Templates
  - And Profiles 1485
- Element View
  - Gantt Chart 542
  - Project Task Allocation 542
- ElementFeatures Package, Automation Interface 2929
  - Attribute Class 2911
  - AttributeConstraint 2915
  - AttributeTag 2916
  - CustomProperties Collection 2918
  - Diagram 2910
  - EmbeddedElements Collection 2918
  - Method 2919
  - MethodConstraint 2923
  - MethodTag 2924
  - Parameter 2925



- 
- ElementFeatures Package, Automation Interface 2929
    - ParamTag 2927
    - Partitions Collection 2928
    - Properties 2929
    - Property 2929
    - TemplateParameter 2931
    - Transitions Collection 2932
  - Ellipse
    - Diagram Layout 875
  - Email
    - Access Within Enterprise Architect 170
    - Exchange Server, Define Default 605
  - Embedded Elements
    - In Child Composite Diagrams 938
    - Include In Document Reports 2691
    - Submenu Option 100
  - Embedded Image
    - In Document Report, Change From Linked 2730
  - EmbeddedElements Collection
    - Automation Interface, ElementFeatures Package 2918
  - EMX
    - File Cross References 480
    - File Import 480
  - Enable
    - Add-Ins 3018
    - ArchiMate 1926
    - Data Flow Diagrams 1797
    - Entity Relationship Diagram 1939
    - Eriksson-Penker MDG Technology 1929
    - Exclusive Diagram Edit Lock 318
    - Gang Of Four Pattern Technology 2285
    - GoF Pattern Technology 2285
    - ICONIX 2282
    - MDG Technologies 1477
    - Mind Mapping 1794
    - Security 318
    - SoaML 2449
    - SOMF 2454
    - SPEM 1919
  - Encrypt Password
    - Prior To Release 7.1 Of Enterprise Architect 333
  - Encrypt Password (Repository)
    - At Release 7.1 Of Enterprise Architect 204, 209
  - End Event, BPEL 1.1
    - Create 1885
    - Model 1885
    - Types 1885
  - End Event, BPEL 2.0
    - Create 1907
    - Model 1907
    - Types 1907
  - End User License Agreement 26
  - Endnote
    - Insert In Document 1071
  - Endpoint
    - Element 1298
  - Enterprise Architect
    - Add-In Model 3010
    - Alignment With UML 1179
    - And Deployment 192
    - And IDEs 9
    - Automation 2790
    - Build Systems 9
    - CASE Tool 3
    - Change Control, Project 2619
    - Change Management 382
    - Cloud Connection, Access To 68
    - Code Engineering With 2076
    - Common Areas Of Work 45
    - Community Site 38
    - Community Site, Access To 68
    - Connectors 1389
    - Create Project, Tutorial 47
    - Database Modeling 9
    - Debug 9
    - Demonstration 67
    - Design Systems 9
    - Dock Windows 128
    - Dockable Windows 125
    - Download Latest Version 68
    - Editions, Differences Between 20
    - Editions, Introduction 18
    - Editor 2250
    - End User License Agreement 26
    - Example Model 68
    - Export Data In CSV Format 501
    - Extend Trial Period 18
    - Feedback Pages 3
    - For Business Analysts 185
    - For Database Administrators 194
    - For Developers 188
    - For Implementation Managers 192
    - For Project Managers 190
    - For Software Architects 186
    - For Software Engineers 187
    - For Technology Developers 193
    - For Testers 191
    - Formal Statements 25
    - Fundamental Processes 7
    - Generate Documentation 9
-

## Enterprise Architect

- Generate Source Code 9
- Getting Started 42
- Glossary 3115
- Help 38
- How To Use 3
- Import Data In CSV Format 503
- In Action 67
- Install 34
- Interface Customization 68
- Interfaces For Modeling 67
- Introduction 3
- Key Features 13
- Keyboard Shortcuts 171
- Keyboard/Mouse Shortcuts 180
- License Agreement 26
- Lite Edition 23
- Main Context Menu 77
- Main Menu 77
- Manage Model Structure 9
- Manage Requirements 9
- MDA Transformation 9
- Model Complexity 9
- Model Requirements 9
- Modeling Tool 3
- Navigation Compass 128
- Newsletters 68
- Object Model, Introduction 2804
- Online Resources 68
- Online User Guide 3
- Order 33
- Overview Guide 68
- Performance, DBMS Connections 218, 225, 233, 239, 248, 255, 257
- Performance, In Auditing 446
- Performance, WAN Optimizer 259
- Predefined Search Definitions 706
- Pricing And Purchasing 33
- Professional Roles 182
- Project Change Control 2619
- Project Files, Open A Project 202
- Project Maintenance 2619
- Project Roles 182
- Project, What Is A? 200
- Quickstart Tutorial 47
- Read-Only Edition 23
- Register Full License 36
- Remove Recent Project 73
- Replication Merge Rules 310
- Repository Scripts 68
- Resources Portal 68
- Reverse Engineer 9

- Scripting 2790
- SDK, Introduction 1483
- Select Edition To Trial 18
- Share Model Development 9
- Software Product License Agreement 26
- Spell Checking 549
- Standard Windows 125
- Start 43
- Start Page 68
- Support 38, 40
- Team Review 343
- Third-Party Extensions 68
- Tools And Features For Modeling 67
- Trial Version 18
- Tutorials 68
- UML 2.3 Support 9
- UML Modeling With 750
- User Interface 67
- Uses Of 9
- Video Demonstrations 68
- Visualize Systems 9
- Web Services Access 170
- What Can I Do With It? 9
- What is Enterprise Architect? 5
- Working With 7

## Enterprise Architect Lite

- Availability 23
- Description 23
- Edition 23

## Enterprise Architect Simulation Library

- Behavioral Model Templates 1688
- EASL Code Generation 1688
- EASL\_GET Macro 1688
- EASLList Macro 1688

## Enterprise Architect Toolbox

- Documentation Group 2669
- MDG Technology Groups 1476

## Enterprise Java Beans 2029

- MDG Technology For, Enterprise Architect 1481

## Entity

- Create 2001
- Element 2000
- Element (Entity Relationship Diagram) 1939

## Entity Bean

- Transformation 2029

## Entity Relationship Diagram

- Concepts 1939
- Connector 1939
- Elements 1939
- Example Diagram 1939
- MDG Technology 1939

- Entity Relationship Diagram
  - Relationship 1939
  - Tagged Values 1939
  - Toolbox Page 1939
  - Transformation From Data Model 2024
  - Transformation To Data Model 2032
- Entry Point
  - Create ConnectionPointReference To 1210
  - Element 1300
- Entry Points
  - Tab Of Structural Specification 967, 976
- Enumeration
  - Automation Interface 2815
  - ConstLayoutStyles 2816
  - CreateBaselineFlag 2817
  - CreateModelType 2818
  - DocumentBreak 2818
  - DocumentPageOrientation 2818
  - DocumentType 2819
  - EAEditionTypes 2819
  - Element, Data Type 1374
  - EnumRelationSetType 2820
  - ExportPackageXMIFlag 2820
  - Literal 1374
  - MDGMenus 2821
  - MessageFlag 2821
  - ObjectType 2822
  - PropType 2823
  - ReloadType 2823
  - ScenarioDiagramType 2824
  - ScenarioStepType 2825
  - ScenarioTestType 2825
  - Structural Diagram Element 1374
  - XMIType 2825
  - XML Element 2412
- Enumeration Element
  - XML, Toolbox Icon 2412
- Enumeration Elements
  - Add To Profiles 1493
- EnumRelationSetType Enum
  - Automation Interface 2820
- Environment Complexity Factor
  - Definition 586
  - Estimate Project Size 588
  - Estimation 586
  - Value 586
  - Weighting 586
- ERD To Data Model
  - Transformation, MDA-Style Transform 2032
- ERDs 1939
- Eriksson-Penker
  - Concept 1929
  - Diagram 1929
  - Disable 1929
  - Elements 1929
  - Enable 1929
  - Extensions 1929
  - MDG Technology 1929
  - Relationships 1929
  - Toolbox Page 1929
- Eriksson-Penker Business Extensions 1802
- Establish Port-Trigger Mappng
  - Model State Machine For HDL 2131
- Estimation 1169
  - Default Hours 588
  - Environment Complexity Factors 586
  - Of Project Factors 584
  - Of Project Size 588
  - Of Project Timescale 584
  - Technical Complexity Factors 585
  - Use Case 584
- Estimation Factors
  - Project Types Menu Option 120
- Event
  - Business Modeling 1812
  - Created In Project Calendar 574
  - Displayed In Project Calendar 574
  - Element 2001
  - Receive 2001
  - Recurring, In Project calendar 574
  - Send 2001
- Event Sequence
  - Simulate With Trigger Sets 2515
- EventProperties
  - Automation Interface Repository 2833
- EventProperty
  - Automation Interface Repository 2833
- Events
  - EA\_OnPreNew 3075
- Example
  - ArcGIS Topology 1954
  - Maintenance Diagram 2635
  - Topology, ArcGIS 1954
- Example Diagram
  - Entity Relationship Diagram 1939
- Example Grammars 1723
- Examples And Tips
  - Automation Interface 2809
- Exception
  - Element 1300
- Exception Path
  - Add To Scenario (Toolbar) 970
- Exception Path, Scenario 965
- Exclamation Mark

- Exclamation Mark
  - Blue 340
  - Red 340
- Exclude
  - Connectors From Document Report, By Type 2660
  - Diagrams From Document Report, By Type 2660
  - Elements From Document Report, By Type 2660
  - Package In Generated Report 2761
- Exclude Package
  - Custom Script 2658
  - SQL Query 2658
- Exclusive Checkout
  - Version Control, TFS 411
- Exclusive Edit Lock
  - Automatic 318
  - Disable 318
  - Enable 318
  - Toggle 318
- Exclusive Gateway, BPEL 1.1
  - Data-Based 1884
  - Event-Based 1884
- Exclusive Gateway, BPEL 2.0
  - Data-Based 1906
  - Event-Based 1906
- Executable Statemachine
  - Artifact 1358
- Executable Statemachine Artifact
  - Build From 2122
  - Compile From 2122
  - Create 2122
  - Execute Code From 2122
  - Generate Code From 2122
  - Simulate Behavior 2122
- Execute
  - Build Script, Build Toolbar Option 147
  - Deploy Script, Build Toolbar Option 147
  - Run Script, Build Toolbar Option 147
  - Test Script, Build Toolbar Option 147
- Execution Analysis
  - Breakpoints And Markers Window, Record Sequence Diagrams 2224, 2540
  - Control Recording, Record Sequence Diagrams 2544
  - Control Stack Depth 2536
  - Diagram Features, Generate Sequence Diagrams 2535
  - Difference Between Recording Marker And Breakpoint 2538
  - Function Line Report 2564
  - Generate Sequence Diagram 2549
  - Marker Types, Record Sequence Diagrams 2538
  - Object Workbench, Create Variables 2568
  - Object Workbench, Introduction 2567
  - Object Workbench, Invoke Methods 2570
  - Object Workbench, Overview 2567
  - Place Markers, Recording Sequence Diagrams 2536
  - Profiler Operation 2557
  - Profiler Report, Load 2560
  - Profiler Report, Save 2560
  - Profiler Report, Save As Resource In Team Review 2566
  - Profiler Toolbar 2558
  - Profiler, Attach To Process 2564
  - Profiler, Getting Started 2558
  - Profiler, Launch 2564
  - Profiler, Overview 2555
  - Profiler, Prerequisites 2556
  - Profiler, Set Options 2563
  - Profiler, Set Sample Intervals 2563
  - Profiler, Start 2564
  - Profiler, Stop 2564
  - Profiler, Supported Platforms 2556
  - Profiler, System Requirements 2556
  - Record Activity For Method 2538
  - Record Sequence Diagrams, Introduction 2531
  - Record Sequence Diagrams, Overview 2532
  - Record Sequence Diagrams, Set Up 2535
  - Record State Transitions 2550
  - Record Unit Test Results 2575
  - Recording Activity For Class 2542
  - Recording History 2533
  - Recording Markers, Activate, Record Sequence Diagrams 2224
  - Recording Markers, Disable, Record Sequence Diagrams 2224
  - Run Unit Test 2575
  - Save Profile Report - Document Artifact 2560
  - Save Profile Report - Team Review Resource 2560
  - Save Recording History 2549
  - Sequence Diagrams, Limit Auto Recording 2538
  - Set Up To Capture State Changes 2551
  - State Machine 2551
  - State Transitions 2551
  - Team Review, Save Profiler Report As Resource 2566
  - Unit Test Script, Set Up 2573
  - Unit Test, Record Results 2575
  - Unit Testing, Introduction 2573

- Execution Analysis
  - Work With Marker Sets, Record Sequence Diagrams 2541
- Execution Analyzer
  - Context Menu 2175
  - Options 2175
  - Record Sequence Diagrams, Recording Options 2219
  - Recording 2547
  - Toolbar 2175
  - Window 2175
- Execution Environment
  - Element 1374
- Execution Profiler
  - Attach To Process 2564
  - Function Line Report 2564
  - Getting Started 2558
  - Launch 2564
  - Operation 2557
  - Overview 2555
  - Prerequisites 2556
  - Report, Example 2555
  - Report, Load 2560
  - Report, Save 2560
  - Report, Save As Resource In Team Review 2566
  - Save Profile Report - Document Artifact 2560
  - Save Profile Report - Team Review Resource 2560
  - Set Options 2563
  - Set Sample Intervals 2563
  - Start 2564
  - Stop 2564
  - Supported Platforms 2556
  - System Requirements 2556
  - Team Review, Save Report As Resource 2566
  - Toolbar 2558
- Exit
  - Menu Option (File Menu) 79
- Exit Point
  - Create ConnectionPointReference To Element 1210
- Expansion Node
  - Action 1301
  - Activity 1301
- Expansion Region
  - Element 1301
- Export
  - .EAB File 437
  - ArcGIS Schema 1961
  - ArcGIS XML Workspace 1961
  - Code Generation Templates 1637
  - Command Set 165
  - Data 376
  - Document Template 2681
  - MDA-Style Transformations 1637
  - Model Elements To External Tools 476
  - Modular ArcGIS Schemas 1962
  - MOF Model To XMI 2460
  - OWL/RDF .XML Files 1980
  - Profile, To Disk 1523
  - Reference Data 376
  - Reference Data, Introduction 374
  - Resource Script, Win32 Dialog 2141
  - State Machine Table To CSV 1222
  - To Rational Rose 473, 475, 482
  - To XMI 475, 476
  - To XML 476
  - Version Controlled Model Branch 437
  - XMI, Batch 492
- Export Diagrams
  - To Document 2741
- Export Package
  - Define Script In MDG Technology 1578
- ExportPackageXMIFlag Enum
  - Automation Interface 2820
- Expose Interface
  - Element 1375
- Extend
  - Archimate 1491
  - BPMN 1491
  - Connector 1406
  - Element With Stereotype 1488
  - Non-UML Objects 1491
  - Relationship 1406
  - SysML 1491
  - Toolbox Elements For MDG Technology 1567
  - ToolboxConnectors For MDG Technology 1569
- Extended Elements 1997
- Extended UML Diagrams 1793
- Extending Modeling Languages
  - And MDG Technologies 1483
  - With Enterprise Architect 1483
- Extending UML
  - With UML Profiles 1471
- Extension
  - SysML Requirement 2314
- Extension Points
  - Use Case 1354
- Extension Stereotypes 1997
- External
  - Element (Data Flow Diagram) 1797
  - Requirements 963, 1765
- External File

External File  
     Add Link To Team Review Post 359  
 External Requirements  
     Color Coded 1776  
 External Tools  
     Open 158  
     Pass Parameters To 160

## - F -

Fan Relations  
     Diagram Layout 888  
 Favorite Elements  
     Add To Resources Window 1176  
     Delete From Resources Window 1176  
     Folder, Resources Window 1176  
     View Properties In Resources Window 1176  
 Favorites  
     Drag Objects Into 686, 693  
     Model Views Folder 686  
 FDD Methodology 1769  
 Feature  
     As Attribute 999  
     As Operation 1014  
     Element 1769  
 Feature Driven Design Methodology 1769  
 Feature Visibility  
     Attributes 845  
     Custom Compartments 845  
     Customize 845  
     Inherited 845  
     Operations 845  
     Set 845  
     Suppress 845  
 Features  
     Connect To 1110  
     Disconnect From 1110  
     Of Enterprise Architect 13  
 Feedback Pages 3  
 Field Chooser  
     In The Specification Manager 1738  
 Field Substitution Macros  
     Access Data From Attributes 1651  
     Access Data From Classes 1652  
     Access Data From Connectors 1660  
     Access Data From Constraints 1664  
     Access Data From Efforts 1665  
     Access Data From Files 1665  
     Access Data From Links 1667  
     Access Data From Metrics 1669  
     Access Data From Operations 1669

Access Data From Packages 1671  
 Access Data From Parameters 1672  
 Access Data From Problems 1673  
 Access Data From Requirements 1673  
 Access Data From Resources 1674  
 Access Data From Risks 1675  
 Access Data From Scenarios 1675  
 Access Data From Tests 1677  
 Attribute 1651  
 Class 1652  
 Conditional Substitution 1649  
 Connector 1660  
 Constraint 1664  
 Direct Substitution 1649  
 Effort 1665  
 Examples 1649  
 File 1665  
 File Import 1666  
 Import, File 1666  
 Link 1667  
 Linked File 1668  
 Metric 1669  
 Miscellaneous 1648  
 Operation 1669  
 Package 1671  
 Parameter 1672  
 Problem 1673  
 Project 1648  
 Requirement 1673  
 Resource 1674  
 Risk 1675  
 Scenario 1675  
 Source Code Generation Options 1655  
 Template Parameter 1677  
 Test 1677

Fields and Conditions  
     In Search 718

Figure  
     Table, Add To Document Report In MS Word 2733

File  
     .EAB 437  
     Element Package, Automation Interface 2895  
     Field Substitution Macros 1665  
     Hyperlink To 2004  
     Linked, Field Substitution Macros 1668  
     Menu 79

File Artifact  
     Create 1362  
     Element 1362  
     External 1362  
     Internal 1362

- File Artifact
  - Options 1362
- File Import
  - Field Substitution Macros 1666
- File Search
  - List View 2164
  - Search Window, Debugging 2164
  - Toolbar 2164
  - Tree View 2164
  - Use 2164
- Fill Factor
  - Index, SQL Server 2365
  - SQL Server Primary Key 2355
  - SQL Server, Index 2365
- Filter
  - Diagram By Context 789
  - Diagram, Display Of Connectors 720
  - Diagram, Display Of Elements 720
  - Elements In Standard Charts 2772
  - Hidden Or Visible Connectors 720
- Filter Bar
  - In Diagram List 677
  - In Model Search 708
  - In Package Browser 677
  - In The Specification Manager 1738
- Filters
  - Add To Search 716
  - Analyzer Script, Recording Options 2219
  - AND 709
  - Diagram, Display Of Connectors 718
  - Diagram, Display Of Elements 718
  - Dynamic Visual 718
  - Element, Document Report 2660
  - Exclude, Document Report 2660
  - OR 709
  - Other, Document Report 2663
  - Search 709
- Find
  - Diagram In Project Browser 93
  - Element In All Diagrams 944
  - Element In Diagrams 97
  - Element In Diagrams, Package Browser 679
  - Element In Project Browser 97, 944
  - Submenu 944
- Find And Replace
  - Access Facility 2161
  - Dialog 2161
  - In Code Editor 2161
  - Metasequences 2161
  - Regular Expressions 2161
  - Tagged Regions 2161
- Find In Diagram
  - Element Context Menu, Project Browser 660
- Find in Project
  - Menu Option (Edit Menu) 80
- Find In Project Browser
  - Search Option 651
- Firebird
  - Create Project File 210
  - Update Index Statistics On Project File 110
- First Chance Exception
  - Process 2244
- Float
  - Current View, Considerations 121
  - Diagram As Window 121
  - Diagrams And Views 790
  - View As Window 121
- Floating
  - Icons 954
  - Toolbar Buttons 954
- Floating Diagram
  - Copy Element From 837
  - Copy Element To 837
  - Move Element From 837
  - Move Element To 837
- Floating Licence
  - Version of Corporate Edition 20
- Floating Toolbar Buttons
  - Display 631
- Floating Window
  - Autohide (Dock First) 132
  - Dock 128
  - Set 128
- Flow Final
  - Element 1306
- Flow Property
  - Direction 2306
  - Flow Direction (SysML Port) 2306
- Font
  - Element, Color 949
  - Element, Script 949
  - Element, Set In Theme 611
  - Element, Size 949
  - Element, Style 949
  - Set For Element Text 949
- Fonts
  - Set Model Default 624
  - Set User Default 624
- Footer
  - Add To Document Report In MS Word 2734
  - Insert In Document 1071
- Foreign Key
  - Composite, Create 2361
  - Composite, Representation In Diagram 2361

- Foreign Key
  - Constraint 2359
  - Create 2359
  - Description 2358
  - Index Template, Define 2363
  - Indexing 2347
  - Name Template, Define 2362
  - Representation In Diagram 2359
  - Transformation 2065
- Foreign Language Translation
  - Document Report Generation 2666
- Fork
  - Element 1307, 1309
  - Node In Simulation 2516
  - Pseudo-State 1307, 1309
- Fork/Join
  - Element 1307
- Formal Reviews Category
  - In Team Review 343
- Formal Statements 25
- Format
  - Element Appearance From Toolbar 785
  - Toolbar 785
- Forum
  - Discussion, Now Team Review 343
- Forward Engineering
  - Initial Code In Operations 1019
  - Introduction 2073
- Forward Slash
  - Derived Symbol 1001
- Forward Synchronization
  - Delete Code From Features In Model 2253
  - Of Package 2117
- Fragment
  - Add To Document Template 2718
  - Edit In Document Template 2718
  - Element 1287
  - Override In Document Template 2718
  - Switch In Document Template 2718
  - Template Fragments 2708
  - View In Document template 2718
- Frame
  - Autohide Windows In 132
  - Combine Windows In 128
  - Remove Windows From 128
  - Reveal Autohidden Windows In 132
  - Tabbed 128
- Free Sorting
  - On Project Browser 672
- Full Lock
  - Check For 342
- Full Screen

- Close 121
- Display 121
- Fully Qualified Tagged Value
  - Show In Element Compartment 845
- Function
  - Macros, Code Template Syntax 1678
  - Step Out Of 2231
- Function Call
  - Step Into 2231
  - Step Through 2548
- Function Line
  - Report, Profiler 2564
- Functional
  - Index 2365

## - G -

- Gang Of Four Pattern
  - Concepts 2285
  - Download 1466
  - MDG Technology 2285
  - Toolbox Page 2285
- Gantt Chart
  - Always Display Diagram As 825
  - Diagram View 784
  - Diagram View (Context Option) 665
  - Element View 542
  - Gantt View 594
  - Package Browser 673
  - Personal Tasks View 554, 555
  - Project View 538
  - Report View 546
  - Resource View 538
  - Show Diagram As 778
- Gantt View
  - Diagram View 594
  - Package View 594
  - Personal Tasks 594
  - Project Management 594
- Gap Analysis Matrix
  - Edit Notes 748
  - Gap Elements 745
  - Notes, Edit 748
  - Overview 745
  - Profiles 745
- Gap Element
  - Create 745
- Gap Notes
  - Edit 748
- Garden Of Eden Style 2418
- Gateway, BPEL 1.1



- Gateway, BPEL 1.1
  - Create 1884
  - Exclusive (XOR) 1884
  - Inclusive (OR) 1884
  - Model In BPEL 1.1 1884
  - Parallel (AND) 1884
  - Types 1884
- Gateway, BPEL 2.0
  - Create 1906
  - Exclusive (XOR) 1906
  - Inclusive (OR) 1906
  - Model In BPEL 2.0 1906
  - Parallel (AND) 1906
  - Types 1906
- GDB
  - Local 2209
  - Remote 2209
  - Script Set Up 2209
- General
  - Types, Dialog 1158
- General Options
  - Set For Project 605
- General Ordering
  - Sequence Diagram Messages 1426
- General Types
  - Project Types Menu Option 120
- Generalization
  - Sets 1120
- Generalization Link
  - Implement Parent Operations 1023
  - Override Parent Operations 1023
- Generalize
  - Connector 1409
  - Relationship 1409
- Generate
  - BPEL 1.1 1893
  - BPEL 2.0 Code 1916
  - DDL 2380
  - Document from Resources (Enhanced) 2668
  - Document Report From Diagram List 684
  - Document Report From Package Browser 673
  - Document Report From Virtual Document 2679
  - Global Element For Global ComplexTypes 2417
  - Global Element In XSD 2418
  - Package Source Code, (Tools Menu Option) 106
  - Report On Elements 673, 684
  - Report On Project 673
  - Rich Text Format Report (Legacy) 2726
  - RTF Document From Resources (Legacy) 2728
  - RTF Report (Legacy) 2726
  - WSDL File 2445
  - XML Schema For Referenced Packages 2417
  - XML Schema, For Child Packages 2417
  - XSD 2417
- Generate Code
  - For Business Rule 1843
  - From Activity Diagrams 2121, 2134
  - From Behavioral Models 2121
  - From Decision Model 1935
  - From Interaction Diagrams 2121, 2133
  - From Legacy State Machine Diagrams 2126
  - From Sequence Diagrams 2121, 2133
  - From State Machine Diagrams 2121
  - From State Machines 2122
  - Hardware Description Languages Supported 2121
  - Supported Hardware Description Languages 2121
  - Via Toolbar 144
- Generate Document
  - Create Links In Word 2730
- Generate Document Report
  - Element Filters Tab 2660
  - Exclude Filters Tab 2660
  - Other Filters Tab 2663
  - Project Constants, User Defined 2664
  - Update Links In Word 2735
- Generate Documentation Dialog
  - Apply Stylesheet 2644
  - Apply Theme 2644
  - Cover Sheet Templates 2654
  - General Tab 2644
  - Insert Cover Page 2644
  - Insert Table Of Contents 2644
  - Options 2644
  - Output Formats 2644
  - Select Template 2644
  - Style Sheet Templates 2653
  - Table Of Contents Templates 2652
  - Template Editor 2704
  - Templates Tab 2681
  - Word Substitution Tab 2665
- Generate Report
  - Quick Start Example 2642
- Generate Sequence Diagram
  - Diagram Features 2535
  - Execution Analysis 2549
- Generate Source Code
  - Enterprise Architect 9
  - Overview 2111
- Geodatabase
  - Export ArcGIS XML Workspace 1961
  - Import ArcGIS XML Workspace 1968

- Geodatabase
  - Modeling 1944
- Geography Markup Language
  - MDG Technology For 1983
- Get All Latest 427
- Get Function
  - Create As Attribute Property 1006
- Get Latest 427
- Get Project Custom Colors
  - For Elements And Connectors 621
- Getting Started
  - Add License Key 3165
  - In Enterprise Architect 42
  - Installing Enterprise Architect 34
  - License Information 3164
  - License Management 3162
  - Register Enterprise Architect 36
  - Registration Key 3164
  - Start Enterprise Architect 43
  - Upgrade Existing License 3169
- Global Attribute
  - XML Element, Toolbox Icon 2396
- Global Element
  - Generate For Global ComplexTypes 2417
  - Generate In XSD 2418
  - Import XSD 2421
  - XML, Toolbox Icon 2392
- Global Elements
  - Import 2420
- Global Risks 524
- Glossary
  - A 3116
  - Add Item, Glossary Dialog 535
  - B 3119
  - C 3121
  - Create Item From Notes Text 1142
  - D 3125
  - Delete Item, Glossary Dialog 535
  - Dialog 535
  - E 3128
  - F 3130
  - Filter List, Glossary Dialog 535
  - G 3132
  - H 3133
  - Hyperlink Term From Notes 1142
  - I 3134
  - Insert Item In Text 1142
  - J 3136
  - L 3137
  - M 3138
  - Model 533
  - Modify Item, Glossary Dialog 535
  - N 3141
  - O 3142
  - Of Terms 3115
  - P 3143
  - Project 533
  - Q 3147
  - R 3148
  - Report 536
  - Report Output Sample 537
  - S 3151
  - T 3156
  - U 3158
  - V 3160
- Glossary Detail Dialog
  - Add Item 534
  - Modify Item 534
- Glossary Entry
  - Add In Specification Manager 1758
- GML
  - Application Schema Files 1983
  - Application Schema, Generate 1987
  - Association Role 1984
  - Attribute 1984
  - Connectors 1984
  - Disable 1983
  - Elements 1984
  - Features 1983
  - ISO/TC 211 1983
  - MDG Technology For 1983
  - OGC 1983
  - Package 1984
  - Profile 1983
  - Relationships 1984
  - Toolbox Pages 1984
- GNU Debugger
  - Local 2209
  - Remote 2209
  - Script Set Up 2209
- Goal
  - Business Modeling 1813
- GoF Pattern
  - Concepts 2285
  - Download 1466
  - MDG Technology 2285
  - Toolbox Page 2285
- Gradient
  - Diagram, Set In Theme 611
  - Element, Set Direction In Theme 611
- Grammar
  - Commands 1709
  - Editing 1721
  - Examples 1723

- Grammar
  - For Imported Code 1705
  - Instructions 1707
  - Parsing AST 1722
  - Profiling Parsing Process 1722
  - Rules 1708
  - Syntax 1706
  - Terms 1709
  - Writing 1705
- Grammar Framework
  - AST Attributes 1711
  - AST Nodes 1711
- Green Circle
  - Breakpoint Marker On Element During Simulation 2477
- Grid
  - Diagram, Adjust 625
  - Snap To Default 625
- Group Element
  - XML, Toolbox Icon 2404
- Group Lock
  - Identify Owner 342
- Group Login 320
- Group Properties Dialog
  - Script 2794
- Guards
  - Javascript, In Simulation 2492
- Guillemets 1457
- H -**
- Hand Drawn
  - Diagram Mode 825
  - Diagram Mode, API 1573
- Hand-Drawn Mode
  - Set 622
- Hardware Description Languages
  - Model State Machine For 2131
  - Supported For Code Generation 2121, 2131
- HDL
  - Model State Machine For 2131
- Header
  - Add To Document Report In MS Word 2734
  - Insert In Document 1071
- Help
  - Add-In 117
  - Display On Add-Ins 117
  - File Formats 39
  - For Tagged Values 1138
  - Index Tab 38
  - Learning Center Topics 74
  - Menu 123
  - Release Date 39
  - Search Tab 38
  - Systems 38
  - Topic, Hyperlink To 2002
  - Version Details 39
- Helpers
  - Profile 1528
  - Profile, Diagram 1536
  - Profile, For Stereotype Profiles 1529
  - Profile, Toolbox 1538
  - Profile, Toolbox Hidden Sub-Menu 1543
- Hidden
  - Connectors, Filter For On Diagram 720
- Hidden Submenu
  - Create In Toolbox Profile 1564
- Hide
  - Connector On Diagram 62
  - Connectors 1119
  - Connectors, All Diagrams 1121
  - Connectors, Requirements Element 1121
  - Connectors, Single Diagram 1121
  - Diagram Caption Bar 84
  - Labels 1122
  - Package Contents On Diagram 777
  - Project Browser 646
  - Relationship 964
  - Toolbox 792
- Hide Name Under Image 860
- Hide/Show
  - Connector Labels 1107
  - Connectors 1107
- Hierarchy
  - Of Requirements 1765
  - Requirement, Import Via CSV 1780
- Highlight
  - Context Element 930
  - Incoming Connectors 1109
  - Outgoing Connectors 1109
- History
  - Deep 1311
  - Element 1311
  - Shallow 1311
- Hit Policy
  - Decision Table 1931
- Hotkeys
  - Diagram Navigation And Selection 841
- Hourly Rate 588
- HTML
  - And CSS Style Editor 2747
  - Report Templates 2747
  - Style Template Fragments 2749
  - Style Templates, Reporting 2747

## HTML

Template Fragments 2749

## HTML Report

Exclude Package 2761

Include Package 2761

## HTTP Binding

WSDL Model 2437

## Hyperlink

Action Element 2004

As Sub Activities 2005

Create In Notes Text 1143

Diagrams 2005

Element 2002

In Linked Document 1095

Insert In Document 1083

Insert New From Toolbar 140

To Element From Linked Document 1095

To External Files 2004

To Help Topics 2002

To Internet Facilities 2002

To Learning Center 2002

To Matrix Profiles 2002

To Model Search 2002

To Other Documents From Linked 1095

To Script 2004

To Team Review 2002

## - | -

## IBM Rational Rhapsody

Import Model From 482

## Icon

Attribute Private 670

Attribute Protected 670

Checked In Package 670

Checked Out Package 670

Composite Element 936

Hidden Decomposition Indicator 936

Namespace Root Package 670

Operation Private 670

Operation Protected 670

Project Browser, User Defined 1503

Special Attribute In Stereotype 1503

Spectacle, Inside Element 936

Version Controlled Package 670

## ICONIX

Disable 2282

Elements 2282

Enable 2282

Layout 2282

MDG Technology 2282

Relationships 2282

Roadmap 2282

UML Toolbox Pages 2282

## Icons

Floating 954

For Custom Elements In Project Browser 1565

For Custom Toolbox Items, Assign 1565

For Custom Toolbox Submenu Items, Assign 1564

For Element On Diagram 954

## IDE

And Enterprise Architect 9

## IDEF1X

Data Modeling Notation 2384

## Image

Add To MDG Technology 1556

Change Linked To Embedded, Document Report 2730

Element, Add To MDG Technology 1574

Handling, Document Report 2730

Insert In Diagram As Boundary 1362

Manager 860

Print Scaled 871

Refresh 860

Scale To Page Size 871

Select Alternative 860

Store In Enterprise Architect 860

Use From Image Library 864

Image Element 2006

## Image Library

Import 864

Use 864

## Image Manager

Create Custom Diagram Background 863

Select Alternative Image For Element 860

Use 860

## IME

Code Editor 2152

## Implementation Details

Report, Generate 2737

## Implementation Manager

And Enterprise Architect 192

Project Role 192

## Implementation Report

In Traceability 724

Target Types, Change 2739

## Implemented Interfaces

Generate/Disable Methods For 2253

## Implements

Connector 1440

Relationship 1440

## Import

**Import**

- .EAB File 438
- .EAB File, Manually 439
- ActionScript, Reverse Engineering 2139
- ArcGIS XML Workspace 1968
- Binary Module (Tools Menu Option) 106
- Binary Module, Languages Supported 2143
- Binary Module, Reverse Engineering 2143
- C#, Reverse Engineering 2139
- C, Reverse Engineering 2139
- C++, Reverse Engineering 2139
- Code Files (Tools Menu Option) 106
- Code Generation Templates 1637
- Code, Select Language 144
- Command Set 165
- Component Types 2249
- Data 380
- Database Schema from ODBC 2376
- DDL Schema from ODBC 2376
- Delphi, Reverse Engineering 2139
- Directory Structure, Reverse Engineering 2142
- Document Template 2681, 2707
- EMX files 480
- File, Field Substitution Macros 1666
- From XMI 478
- Global Elements 2420
- Handle Classes Not Found 2144
- Image Library 864
- Java, Reverse Engineering 2139
- MDA-Style Transformation Templates 1637
- MDG Technologies To APPDATA 1480
- Model From IBM Rational Rhapsody 482
- Normal.rtf Template 1058
- OWL/RDF .XML Files 1980
- Pattern 1466
- PHP, Reverse Engineering 2139
- Python, Reverse Engineering 2139
- Reference Data, Automatically 380
- Reference Data, Introduction 374
- Reference Data, Manually 380
- Referenced XML Schema 2420
- Requirements Via CSV 1780
- Resource Script, Win32 Dialog 2141
- Scenario as Test 2611
- Scenarios From Package 2611
- Source Code, Reverse Engineering 2138, 2144
- Source Directory (Tools Menu Option) 106
- Split WSDL Files 2447
- Test From Other Element 2613
- UML Pattern 1466
- UML Profiles To Resources Window 1526
- UML2 files 480
- User ID From Active Directory 324
- Version Controlled Model Branch 438, 439
- Visual Basic, Reverse Engineering 2139
- Visual Basic.Net, Reverse Engineering 2139
- WSDL Files 2447
- XMI 493
- XSD 2420
- Import Package
  - Define Script In MDG Technology 1578
- Import Package Dialog 431
- Import/Export
  - Submenu (Project Menu) 90
- Import/Export Submenu
  - Package Context Menu, Project Browser 659
- Imported Class Elements 2379
- Inbuilt Stereotypes 1997
- Include
  - Connector 1410
  - Package In Generated Report 2761
  - Relationship 1410
- Indentation
  - Automatic, Common Code Editor 2157
  - Manual, Common Code Editor 2157
- Index
  - Clustered 2365
  - Column-Based 2365
  - Data Modeling, Create 2365
  - Foreign Keys, Database Engineering 2347
  - Functional 2365
  - Oracle, Function Based 2365
  - SQL Server 2365
  - Stereotype 2365
  - Table 2365
  - Unique 2365
- Index Template
  - Foreign Key, Define 2363
- Indicator
  - Columns, Specification Manager 1743
  - Icons 1743
  - Locked Element 340
  - Visibility 830
- Information Engineering
  - Data Modeling Notation 2384
- Information Flow
  - And Patterns 1411
  - Connector 1410, 1411
  - In Combination 1411
  - Locate Items Conveyed 1410
  - Realized 1414
  - Relationship 1410, 1411
- Information Item
  - Conveyed 1413

- Information Item
  - Element 1376
  - Find In Project Browser 1103, 1411
- Inheritance
  - Connector 1409
  - Relationship 1409
- Inherited Attributes
  - Display 1007
- Inherited Feature
  - Show 845
- Inherited Operation
  - Display 1024
- Inherited Port
  - Manage 1385
- Initial
  - Element 1313
- Initial Code
  - Operations Properties 1019
- Initializer
  - For Attribute, Override 1007
- Inline Features
  - Submenu (Element Menu) 101
- Inline IME
  - Option In Document Editor 1054
- Inline Sequence Elements
  - Part And Port 1258
- InnoDB
  - BaseModel Script 220
- In-place Editor
  - Automatic Scroll Into View 1032
  - Connector Options 1123
  - Custom Compartment 1039
  - Edit Attribute Keyword 1037
  - Edit Attribute Name 1034
  - Edit Attribute Scope 1036
  - Edit Attribute Stereotype 1035
  - Edit Connector Labels 1123
  - Edit Element Name 1034
  - Edit Operation Name 1034
  - Edit Operation Parameter Keyword 1038
  - Edit Operation Parameter Kind 1039
  - Edit Operation Scope 1036
  - Edit Operation Stereotype 1035
  - Element Item, Tasks 1032
  - Element Options 1032
  - Insert Attribute To Element 1040
  - Insert Maintenance Feature 1042
  - Insert Operation Parameter 1041
  - Insert Operation To Element 1040
  - Insert Testing Feature 1043
  - Parts 1039
  - Ports 1039
- In-Place Formatting
  - Elements 954
- Input Method Editor
  - Code Editor 2152
- Insert
  - Attribute To Element, In-place Editor 1040
  - Bookmark Into Word 2730
  - Boundary Element 1348
  - Diagram Properties Note 848
  - Linked Elements 933
  - Maintenance Feature In Element 1042
  - Operation To Element, In-place Editor 1040
  - Related Elements 933
  - Testing Features In Element 1043
- Install
  - Enterprise Architect 34
- Instance
  - Classifier 1009
  - Define Creation, Special Attributes 1510
  - Examples 1009
  - In Simulation 2479
  - Paste Object As 833
  - Select Property On Classifier 1011
- Instantiated Template 1365
- Integrated Development Environment 2172
- Integration Testing
  - Display Details 2607
- Integrity
  - Of Model Data 597
  - Of Project Data 597
- Integrity Check 597
- Intelli-sense
  - Code Editor, Common 2160
  - In Script Editor 2798
  - Keystrokes, Script Editor 2798
  - Libraries 2166
  - Search Control 2166
- Interaction
  - Behavioral Aspects 1026
  - Diagram 1225, 1259, 1262
  - Edit Parameters 1026
  - Element 1314
  - Elements and Connectors 806
  - Parameter 1026
  - Parameters, Define 1028
  - Toolbox Pages 806
- Interaction Diagram
  - Description 1249
  - Diagram 1249
  - Elements And Connectors 1249
  - Example 1251
  - Generate Code From 2121, 2133

- Interaction Occurrence
  - Behavior Call 1026
  - Element 1317
- Interaction Operand Codition
  - In Simulation 2490
- Interaction Operator
  - Combined Fragment 1290
- Interaction Overview Diagram
  - Description 1262
  - Elements And Connectors 1262
  - Example 1263
- InteractionUse
  - Element 1317
- Interface
  - Customization Wizard 167
  - Element 1377
  - Expose Element 1375
  - Provided 1375
  - Required 1375
  - Set For Element 908
  - Source Code Generation 2111
- Interface Customization Wizard 68
- Intermediary Language
  - Debug, MDA Transformations 2053
  - MDA-Style Transforms 2053
- Intermediate Event, BPEL 1.1
  - Create 1878
  - Model 1878
  - Types 1878
- Intermediate Event, BPEL 2.0
  - Create 1901
  - Model 1901
  - Types 1901
- Internal Binding
  - PIM to PSM 2013
- Internal Block (SysML)
  - Render Part From Block Association 2304
  - Render Property From Block Association 2304
- Internal Editor
  - Language Properties 2250
- Internal Mail
  - Context Menu 565
  - Create Message 570
  - Delete Messages 565
  - Forward Message 570
  - Hyperlinks 570
  - Internal Mail 565
  - Model Mail Tab 565
  - Options 565
  - Personal Information 570
  - Personal Information Window 565
  - Reply To Message 570
  - Sent Mail Tab 565
  - Toolbar 565
- Internal Requirement
  - Define 1770
  - Properties 1770
- Internal Transition
  - Create 1449
  - Multiple 1449
- Internet
  - Connect To Project Via 266
  - Connecting To Model Via 262
- Internet Browser Applets
  - Java, Debug 2194
- Internet Facilities
  - Hyperlink To 2002
- Internet Search Engine
  - Access Within Enterprise Architect 170
  - Define Default 605
- Interrupt Flow
  - Connector 1415
  - Relationship 1415
- Interruptible Activity Region
  - Element 1316
- Interval Bar
  - Context Menu 1241
  - Timing Diagram Time Interval 1241
- Introduction
  - Copyright Notice 25
  - Enterprise Architect Key Features 13
  - Help File Formats 39
  - License Agreement 26
  - Order Enterprise Architect 33
  - Support 40
  - To Code Engineering 2073
  - To Enterprise Architect 3, 5
  - To Enterprise Architect SDK 1483
  - To Forward Engineering 2073
  - To Quick Linker 896
  - To Quick Linker Definitions 1515
  - To Reverse Engineering 2073
  - To Round-trip Engineering 2073
  - To Shape Scripts 1582
  - To Synchronization 2073
  - To Tagged Value Types 1621
  - To UML Objects 1265
  - To Version Control 383
  - Trademarks 29
- Invocation
  - Action, Simulation 1268
  - Associate With Behaviors 1027
  - Re-associate With Behaviors 1027

- Invocation
    - Synchronize Arguments With Behavior Parameters 1027
  - Invocation Actions 2134
  - Invocations
    - Modeling 1026
  - Invoke
    - Connector 801
    - Method, Object Workbench 2570
  - ISO/TC 211 1983
  - Issue
    - Add 2631
    - Create From Test Item 2615
    - Element 2631
    - Elements And Requirements 1785
    - Hide Stereotype Letter 2631
    - Items, Element Maintenance 2623
    - Show Stereotype Letter 2631
  - Issue (Defect)
    - Automation Interface, Element Package 2896
    - Element 2631
  - Issue (Project)
    - Add 529
    - Delete 529
    - Model 528
    - Modify 529
    - Report, Generate 528
  - Issues Report
    - On Elements, Generate 2740
    - Sample Output 532
    - Via Issues Tab 531
  - isUnique
    - UML Property 1130
  - Iterate Through EAP File
    - Automation Interface Code Example 2996
- J -**
- Java
    - Advanced Debug Techniques 2194
    - Applets In Internet Browsers, Debug 2194
    - AspectJ Extensions 2139
    - Code Generated From State Machine Diagram 2128
    - Code Generation 2272
    - Debug 2191
    - Debug On Android Devices 2210
    - Debug Session, Attach To VM 2194
    - Debug Wire Protocol 2210
    - Debug, System Requirements 2186
    - General Debug Setup 2191
    - Import, Reverse Engineering 2139
    - JDWP 2210
    - Language Options 2272
    - Modeling Conventions 2096
    - Modeling Conventions, AspectJ Extensions 2098
    - Object Workbench, Set Up 2216
    - Sample Model, Visual Execution Analysis 2529
    - Transformation, MDA-Style Transform 2034
    - Web Servers, Debugging 2195
  - Java Debug Wire Protocol
    - Debugger 2213
    - JDWP 2213
  - Java Program
    - Application Pattern 2077
  - JavaScript 2791, 2792
    - Create Object In Simulation 2480
    - Destroy Object In Simulation 2483
    - Instantiate Object Member, In Simulation 2480
    - To Simulate Activity Behavior 2489
    - To Simulate Class Behavior 2489
  - JBOSS
    - Server Configuration 2198
    - Server, Debugging 2195
  - JDWP Debugger
    - Local 2213
    - Remote 2213
    - Script Set Up 2213
  - Jet 3.5
    - Switch To Jet 4.0 43, 605
  - Jet 4.0
    - Supporting Spell Checking Languages 552
    - Switch From Jet 3.5 43, 605
  - Join
    - Element 1307, 1310
    - Node In Simulation 2516
    - Pseudo-State 1307, 1310
  - JScript 2791, 2792
  - Jump
    - Connectors 1114
  - Junction
    - Element 1319
  - JUnit Transformation
    - MDA-Style Transform 2037
- K -**
- Kanban
    - Apply Element Status 857
    - Apply Tagged Values 857
    - Facilities 857
    - Lanes 857
    - Project Management Methodology 857



Kanban  
 Set Up 857  
 Key  
 Combinations 171  
 Database 2354  
 Foreign 2354  
 Missing Combination 152  
 Primary 2354  
 Primary, Create 2356  
 Shared, Add For Add-In 3172  
 Key Bindings  
 Code Editor, Common 2167  
 Key Features  
 Of Enterprise Architect 13  
 Key Store  
 File Based 3165  
 Network Based 3165  
 Keyboard  
 Accelerator Map 171  
 And Mouse Shortcuts 180  
 Missing Combination 152  
 Shortcuts 171  
 Shortcuts, Customize 161  
 Shortcuts, Reset 161  
 Key-Mouse  
 Combinations 180  
 Keys  
 Foreign, Definition 1937  
 Primary, Definition 1937  
 Keystore  
 Troubleshooting 3168  
 Keywords  
 For RTF Report (Legacy) 2725

## - L -

Label  
 Alignment 866  
 Bold 866  
 Connector 866  
 Connector, Hide/Show 1107  
 Direction Indicator 866  
 Display Under Toolbar Icons 156  
 Element 866  
 Hide 866, 1122  
 Menu 866  
 ODM Element, Define New 1980  
 Set Color 866  
 Show 1122  
 Show And Hide In Toolbox 797  
 Visibility 1122  
 Label Visibility  
 On Sequence Messages 1258  
 Lane  
 Orientation, BPMN 2.0 1845  
 Language  
 Adjust in Document Report 2666  
 Adjust in RTF (Legacy) 2725  
 Code Generation, Hardware Definition 2080  
 Code Generation, Software 2080  
 Create Properties As Attributes 1006  
 For Spell Check 552  
 Macros 2257  
 Other Than English, General Interface, Unicode 43  
 Other Than English, Spell Checking 552  
 Pack For Download 552  
 Programming, Develop 1581  
 Reverse Engineering, Supported 2136  
 Software, Import, Supported 2139  
 Software, Reverse Engineering, Supported 2139  
 Support, Natural, Under Unicode 43  
 Supported By Code Editor 2147  
 Supported Hardware Description Languages 2131  
 Supported, Debugging 2074  
 Supported, Execution Analysis 2074  
 Supported, Recording 2074  
 Language Options  
 ActionScript 2264  
 Ada 2005 2264  
 ArcGIS 2265  
 C 2266  
 C# 2268  
 C++ 2269  
 Code Editor 2251  
 Code Engineering 2262  
 Code Generation 2262  
 Delphi 2271  
 Java 2272  
 MDG Technology 2278  
 PHP 2274  
 Python 2273  
 Reverse Engineering 2262  
 SystemC 2275  
 VB.NET 2275  
 Verilog 2276  
 VHDL 2277  
 Visual Basic 2277  
 Layout  
 Diagram, Automatically 891  
 Diagram, Move Sections 869  
 ICONIX 2282

- Layout
  - Sequence Diagram 1252
  - Workspace, Manage 163
- Layout Chains
  - Procedure 894
  - Suggested Options And Sequence 894
  - Window 894
- Layout Diagram
  - Auto Route Layout 890
  - Box Layout 879
  - Center Focussed Layout 875
  - Chain Layouts 874, 894
  - Circular Layout 875
  - Converge Layout 886
  - Digraph Layout 883
  - Diverge Layout 886
  - Elliptical Layout 875
  - Fan Relations Layout 888
  - Main Procedure 874
  - Neaten Layout 885
  - Options 874
  - Per Page Layout 881
  - Spring Layout 883
  - Suggested Combinations Of Formats 894
  - Tools 874
  - Top-To-Bottom Circle 875
- Layout Profile
  - Package Browser, Manage 679
- Layout Tools
  - Window 874
- Lazy Loading
  - Enterprise Architect Performance 218, 225, 233, 239, 248, 255, 257
  - For ASA Data Repository 239
  - For MySQL Data Repository 225
  - For Oracle Data Repository 255, 257
  - For PostgreSQL Data Repository 233
  - For Progress OpenEdge Data Repository 248
  - For SQL Server Data Repository 218
- Learning Center
  - Create Pages For Technology 74
  - Getting Started 74
  - Introduction 38
  - Task Guides 74
  - Window 74
- Legacy State Machine Diagram
  - Generate Code From 2126
- Legend
  - Add To Diagram 849
  - Add To State Machine Table 1219
  - Edit 849
  - Element 849
  - Ignore 1219
  - Insert New From Toolbar 140
  - Never Happen 1219
  - Remove From State Machine Table 1219
  - Style 849
- Letter A
  - Red 1044, 1047
- Level Indenting
  - Apply In Specification Manager 1759
- Level Numbering
  - Apply In Specification Manager 1759
  - For Requirements 1765
  - Show In Project Browser 651
  - Turn On/Off, Package Context Menu 654
- Library
  - Intelli-sense, Search 2166
- License
  - Agreement 26
  - Information 3164
  - Management 3162
- License Management
  - Add Shared Key For Add-In 3172
- Lifecycle
  - Of A Sequence Element 1251
- Lifeline
  - Element 1321
  - Objects In Sequence Diagrams 1254
  - Sequence Element, Termination 1251
- Limit Auto Recording to Stack Frame Threshold
  - Marker Types, Record Sequence Diagrams 2538
- Limitations
  - Of XMI 482
- Line
  - Angles, Tidy 1114
  - Bend At Cursor 1114
  - Bezier 1114
  - Jumps 1114
  - Straighten At Cursor 1114
  - Style 1114
  - Suppress Segments 1114
  - Width, Set In Theme 611
- Line Points
  - Add 1114
  - Delete 1114
  - Toggle 1114
- Line Selection
  - Code Editor, Common 2157
- Link
  - Add Note 1111
  - Copy Between Instances Of Elements From Project Browser 1118

- Link
  - Create Between Elements 1118
  - Create From Project Browser 1118
  - Display Options 635
  - Element Feature To Note 924
  - Field Substitution Macros 1667
  - Note To Element Feature 924
  - Paste Object As 833
  - Relationship Matrix 727
  - To Element Feature 1110
  - To Shared Repository Information 375
  - Window 742
- Link Notes
  - To Element Feature 924
  - To Internal Documentation 924
- Linked Diagram
  - Child Diagram 936
  - Composite Element 936
  - Switch On Composite Element 936
- Linked Document
  - Add Search Items To 703
  - Add Search Objects As Huperlinks 703
  - Content In Document Report 2694
  - Create 1044
  - Create Diagram From 1096
  - Create Element From 1096
  - Create Glossary Definition In 1048
  - Create In Document Artifact 1046
  - Create On UML Element 1047
  - Delete 1097
  - Drag Items From Model Search 693
  - Edit 1048
  - Editor Context Menu 1048
  - Editor Display Options 1052
  - Generate Document Report To 2680
  - Hyperlink To Elements 1095
  - Hyperlink To Other Documents 1095
  - Introduction 1044
  - On UML Element 1047
  - Render In Document Report 1044
  - Render Items As RTF 703
  - Replace 1097
  - Show Headers & Footers 1052
  - Show Hidden Text 1052
  - Template Design 1052
  - Template, File Control Options 1050
  - Template, Print Options 1094
  - Template, Scroll And Search Options 1059
  - Zoom 1052
- Linked Document Template
  - Add To MDG Technology 1555
  - Assign To Group 1098
  - Create 1098
  - Delete 1098
  - Edit 1100
  - Modify 1098
- Linked Element
  - Convert To Local Copy 931
- Linked File
  - Field Substitution Macros 1668
- Linked Image
  - In Document Report, Change To Embedded 2730
- List Format
  - In Generated Documents, Toggle 1064
- List Macro
  - Code Template Syntax 1683
- List Overrides
  - Document Reports 1081
- List Symbols
  - In Generated Documents, Toggle 1064
- Literal Text
  - Code Template Syntax 1643
- Live Code Generation
  - Menu Option 657
- Load
  - Controlled Package 492
  - Report Templates (Legacy) 2726
- Loaded Modules
  - Show In Debugger 2243
- Local
  - Directories 2256
  - Path Dialog 2256
  - Paths 2255
  - Pre/Post Conditions 1274
- Local Attribute
  - XML, Toolbox Icon 2398
- Local Element
  - XML, Toolbox Icon 2394
- Local Variables
  - View 2234
  - View Long Values 2235
- Locals Window 2234
  - View Long Values 2235
- Locate Compiler Error
  - In Code, For Build 2221
- Lock
  - Apply User Lock 339
  - Connector 332
  - Delete 332
  - Delete, User level 342
  - Diagram 336
  - Diagram, Require User Lock 339
  - Diagram, Security Off 872

## Lock

- Diagram, User/Group Locking 336
- Element 332, 336
- Element, Require User Lock 339
- Element, Security Off 939
- Element, User/Group Locking 336
- Full, On Element/Diagram 336
- Full, On Package 337
- Identify Owner 342
- Manage 332
- Manage, User-Level 342
- Model Elements 336
- Model Elements, Require User Lock 339
- Package 336
- Package, Require User Lock 339
- Packages, User/Group Locking 337
- Release On Package, User/Group Locking 337
- Release User Lock 339
- View 332
- View, User Level 342

## Locked Element

- Add Connectors 388
- Indicators 340

## Log

- Audit, Clear 448
- Audit, Load 448
- Audit, Save 448

## Logical Diagram

- Class Diagram 1184

## Logical Model 761

- Template (Class) 755

## Login

- As Different User (Security) 92
- Group 320
- Multiple Under One ID 320

## Loop Node

- Simulate 2499
- Structured Activity 1338, 1341

**- M -**

## Macro

- Behavioral Model 1688
- Branching 1684
- Code Template Syntax 1646, 1683
- Control 1683
- CONVERT\_DB\_TYPE 2066
- CONVERT\_NAMES 2067
- CONVERT\_TYPE 2066
- Direct Substitution 1643
- EASL Code Generation 1688

EASL\_GET 1688

EASLList 1688

Field Substitution, Code Template Syntax 1648

Function, Code Template Syntax 1678

Language 2257

List 1683

PI 1683, 1687

Preprocessor 2257

Processing Instruction 1687

REMOVE\_PREFIX 2067

SQL Search 711

Synchronization 1686

Tagged Value, Code Template Syntax 1676

Template Substitution, Code Template Syntax 1647

TRANSFORM\_CLASSIFIER 2068

TRANSFORM\_CURRENT 2063, 2066

TRANSFORM\_REFERENCE 2063, 2068

TRANSFORM\_TAGS 2066

## Mail Interface Package

Automation Interface 2990

MailInterface, Automation Interface 2990

## MailInterface Class

Automation Interface, Mail Interface Package 2990

Main Context Menu 77

Main Menu 77

## Maintain

Groups 320

Security Users 323

## Maintenance

Asterisk On Maintenance Window Tabs 2623

Compartment, Element 955

Create Defect Item From Test 2615

Dialog 1169

Elements And Connectors 815

Feature, Insert In Element 1042

Items 2621

Items, Show In Compartments 2630

Model Template 755, 766

Of Element Management 2625

Problem Types 1169

Support 2625

Testing Types 1170

Toolbox Pages 815

Window 2623

Workspace 2623

## Maintenance Data

Project Types Menu Option 120

## Maintenance Diagram

Description 2634

Elements And Connectors 2634

- Maintenance Diagram
  - Example 2635
- Maintenance Item
  - Copy Between Categories 2628
  - Create Element From 2629
  - Drag Onto New Element 989
  - Move Between Categories 2628
- Maintenance Report
  - Generate 2740
- Make Same
  - Submenu 953
- Manage
  - Add-Ins 3018
  - Baselines 461
  - Bookmarks 591
  - Changes 382
  - Diagrams 820
  - Elements 900
  - Inherited Ports 1385
  - Locks 332
  - MDG Technologies 1477
  - Redefined Ports 1385
  - Requirements 1782
  - User-Level Locks 342
  - Views 769
  - Views, Add Views 770
  - Views, Delete 771
  - Views, Rename 771
- Manage .EAP File
  - Submenu, Data Management Menu 111
- Manage Projects 68
- Manage Uses Context List Dialog 974
- Managed C++
  - Modeling Conventions 2093
- Manifest
  - Connector 1416
  - Relationship 1416
- Manual Version Control
  - With XMI 494
- Map
  - Data Types Between DBMSs 2353
- Map State Changes
  - Visual Execution Analyzer 2553
- Mapper
  - Data Type Conversion, Table 2350
- Marker
  - Record, Set 2537
  - Sets 2542
  - Types 2538, 2542
- Marker Management
  - Debugger 2224
- Marker Sets
  - Create 2541
  - Delete 2541
  - Maintain 2541
  - Work With 2541
- Markers
  - Recording, Nested 2548
- Masked Tagged Values
  - Create 1626
  - Template For 1626
- Master Document
  - Bookmarks In 2669
  - Element 2669
  - Element, Create 2672
  - Generate Documentation 2679
  - Generate Report To Document Artifact 2680
  - Generate Report To Linked Document 2680
  - Overview, Virtual Document 2669
  - Tagged Values 2672
- Master Document (HTML) 2759
- Math Functions
  - In Script Editor 2798
- Matrix
  - Overlays, Apply 734
  - Overlays, Create 733
  - Overlays, Delete 733
  - Overlays, Edit 733, 734
  - Overlays, Remove 734
  - Swimlanes 854
- Matrix Profile
  - Create In Specification Manager 1744
  - Hyperlink To 2002
- Matrix Specification
  - Artifact 1358
- MatrixOverlay
  - Tagged Value Type 733
- MDA Transform
  - Add To MDG Technology 1554
- MDA Transformation
  - Built-In 2013
  - Debug Intermediary Language 2053
  - Enterprise Architect 9
  - Overview 2013
  - Tools Menu Option 109
- MDA-Style Transformation
  - Built In, List Of 2020
  - C# Transformation 2021
  - C++ Transformation 2023
  - Chaining Transformations 2019
  - Communication Diagram To Sequence Diagram 2041
  - Convert Datatypes 2066
  - Convert Names 2067

- MDA-Style Transformation
  - Convert Types 2066
  - Copy Information 2066
  - Copy Tagged Values 2066
  - Create Connectors 2060
  - Create Objects 2055
  - Cross References 2068
  - Data Model To ERD Transformation 2024
  - DDL Transformation 2025
  - Duplication Of Connectors 2063
  - EJB Transformations 2029
  - ERD To Data Model Transformation 2032
  - Export Transformations 1637
  - Foreign Keys 2065
  - Import Transformation Templates 1637
  - Intermediary Language 2053
  - Java Transformation 2034
  - JUnit Transformation 2037
  - NUnit Transformation 2038
  - PHP Transformation 2040
  - Remove Prefix 2067
  - Sequence Diagram To Communication Diagram 2041
  - Specify Classifiers 2068
  - Template Parameter Substitution 2069
  - Transform Connectors 2063
  - Transform Elements 2017
  - Transformation Templates 2048
  - VB.Net Transformation 2042
  - Write Transformations 2051
  - WSDL Transformation 2043
  - XSD Transformation 2044
- MDDE
  - Advanced Debug Techniques, Java 2194
  - Analyzer Scripts, Manage 2175
  - Available Tools 2174
  - Basic Setup 2175
  - Breakpoint Management 2224
  - Build Script, Create 2180
  - Build Script, Introduction 2180
  - Debug .NET 2200
  - Debug .NET CLR Versions 2201
  - Debug .NET With COM Interop Process 2202
  - Debug Apache Tomcat Server Configuration 2199
  - Debug Apache Tomcat Windows Service 2199
  - Debug ASP .NET 2203
  - Debug Java 2191
  - Debug Java Applets In Internet Browsers 2194
  - Debug Java Web Servers 2195
  - Debug JBOSS Server Configuration 2198
  - Debug Symbols, C++ And Native Applications 2191
  - Debugger Frameworks 2222
  - Debugger System Requirements 2186
  - Debugger, Overview 2222
  - External Tools 2175
  - For C++ Applications 2189
  - For Microsoft Native Applications 2189
  - For WINE Applications 2188
  - General Debug Setup, Java 2191
  - General Workflow 2173
  - Generate Code 2173
  - Getting Started 2173
  - Introduction 2172
  - Java Debug Session, Attach To VM 2194
  - Limitations 2074
  - Marker Management 2224
  - Overview 2074
  - Prerequisites 2173
  - Recursive Builds 2180
  - Script Actions, Define 2179
  - Set Up Debug Session 2185
  - Set Up Debug Session For .NET 2200
  - Supported Environments 2074
  - Synchronize Code 2173
  - UAC-Enabled Operating Systems 2187
  - Workspace Layout 2174
- MDG Add-In For
  - CORBA 1481
  - Data Distribution Service 1481
  - DDS 1481
  - Department Of Defense Architecture Framework - Ministry Of Defence Architecture Framework 1481
  - DoDAF-MODAF 1481
  - DOORS 1481
  - Eclipse 1481
- MDG Add-Ins
  - Add-In Model 3097
  - MDG Events 3098
  - MDG\_BuildProject 3099
  - MDG\_Connect 3100
  - MDG\_Disconnect 3101
  - MDG\_GetConnectedPackages 3102
  - MDG\_GetProperty 3102
  - MDG\_Merge 3103
  - MDG\_NewClass 3106
  - MDG\_PostGenerate 3107
  - MDG\_PostMerge 3108
  - MDG\_PreGenerate 3109
  - MDG\_PreMerge 3109
  - MDG\_PreReverse 3110

- MDG Add-Ins
  - MDG\_Run\_Exe 3111
  - MDG\_View 3112
- MDG Events
  - Add-In Model 3098
  - MDG\_BuildProject 3099
  - MDG\_Connect 3100
  - MDG\_Disconnect 3101
  - MDG\_GetConnectedPackages 3102
  - MDG\_GetProperty 3102
  - MDG\_Merge 3103
  - MDG\_NewClass 3106
  - MDG\_PostGenerate 3107
  - MDG\_PostMerge 3108
  - MDG\_PreGenerate 3109
  - MDG\_PreMerge 3109
  - MDG\_PreReverse 3110
  - MDG\_Run\_Exe 3111
  - MDG\_View 3112
- MDG Integration For
  - Eclipse 1481
  - Visual Studio 2005/2008 1481
- MDG Link For
  - DOORS, Telelogic 1481
  - Eclipse 1481
  - Visio 1481
  - Visual Studio.NET 1481
- MDG Technologies
  - Import To Local System From Registry 289
  - In Reusable Asset Service Registry 289
- MDG Technology
  - Access, Remote From Enterprise Architect 1479
  - Activate 1477
  - Active, Automatically Pinned Pages 797
  - And Resources Window 1476
  - ArcGIS 1944
  - Code Generation 2278
  - Connectors Used In Toolboxes 1569
  - Create 1545
  - Create Learning Center Pages To Support 74
  - Create Toolbox Profile For 1561
  - Custom, Define Export/Import Scripts For 1578
  - Default For Model 1477
  - Define Validation Configuration 1576
  - Deploy From Add-In 1580
  - Deploy From File 1580
  - Develop Code Module 1581
  - Disable 1477
  - Elements Used In Toolboxes 1567
  - Enable 1477
  - Example Of Development 1527
  - Example Of Element Image Setup 1574
  - Extend UML Elements 1567
  - Extend UMLConnectors 1569
  - Geography Markup Language 1983
  - GML 1983
  - Import To APPDATA 1480
  - Include Custom Diagram Types 1536, 1570
  - Include Custom Toolbox 1538
  - Incorporate Model Template 1576
  - Introduction 1527
  - Introduction, Technology Sources 1475
  - Language Options 2278
  - Link for Downloads 1475
  - Manage 1477
  - ODM 1971
  - Ontology Definition Metamodel 1971
  - Profile Helpers 1528
  - RAS, Import To Model 291
  - Remove From Model 1477
  - Select 167
  - Toolbox Groups 1476
  - Win32® User Interface 1996
- MDG Technology (Integrated)
  - ArchiMate 1926
  - BPEL 1870
  - BPMN 1845
  - Data Flow Diagram 1797
  - Entity Relationship Diagram 1939
  - Eriksson-Penker 1929
  - Gang Of Four Pattern 2285
  - GoF Pattern 2285
  - ICONIX 2282
  - Mind Mapping 1794
  - SoaML 2449
  - SOMF 2.1 2454
  - SPEM 1919
  - SysML 2294
- MDG Technology For
  - CORBA 1481
  - Data Distribution Service 1481
  - DDS 1481
  - Department Of Defense Architecture Framework - Ministry Of Defence Architecture Framework 1481
  - DoDAF-MODAF 1481
  - Enterprise Java Beans 1481
  - Python 1481
  - SysML 1481
  - Systems Modeling Languages 1481
  - Testing 1481
  - The Open Group Architecture Framework 1481
  - TOGAF 1481

- MDG Technology For
  - Zachman Framework 1481
- MDG Technology For GML
  - Features 1983
  - ISO/TC 211 1983
  - Profile 1983
- MDG Technology Selection (MTS) File 1559
- MDG Technology Wizard
  - Add Code Modules 1552
  - Add Diagram Type To Technology 1550
  - Add Document Report Template To Technology 1555
  - Add Images 1556
  - Add Linked Document Template To Technology 1555
  - Add MDA Transforms 1554
  - Add Model Searches 1559
  - Add Model Views 1558
  - Add Pattern To Technology 1549
  - Add Profile To Technology 1548
  - Add Scripts 1556
  - Add Tagged Value Types 1551
  - Add Toolbox To Technology 1551
  - Add Workspace Layouts 1557
  - Create Technologies 1545
- MDG\_BuildProject
  - Add-In Model 3099
- MDG\_Connect
  - Add-In Model 3100
- MDG\_Disconnect
  - Add-In Model 3101
- MDG\_GetConnectedPackages
  - Add-In Model 3102
- MDG\_GetProperty
  - Add-In Model 3102
- MDG\_Merge
  - Add-In Model 3103
- MDG\_NewClass
  - Add-In Model 3106
- MDG\_PostGenerate
  - Add-In Model 3107
- MDG\_PostMerge
  - Add-In Model 3108
- MDG\_PreGenerate
  - Add-In Model 3109
- MDG\_PreMerge
  - Add-In Model 3109
- MDG\_PreReverse
  - Add-In Model 3110
- MDG\_Run\_Exe
  - Add-In Model 3111
- MDG\_View
  - Add-In Model 3112
- MDGMenus Enum
  - Automation Interface 2821
- Memory Viewer
  - Analyzer Menu Option 112
  - Window 2242
- Menu
  - Analyzer 112
  - Context, Diagram Advanced 783
  - Context, Project Browser 648
  - Customize Display 162
  - Diagram 93
  - Diagram Context 778
  - Diagram, Advanced 95
  - Edit 80
  - Element 97
  - Element Context 939
  - Element Properties 939
  - Extensions 117
  - File 79
  - Help 123
  - Items, Define In Add-In 3013
  - Label 866
  - Main 77
  - Main Context 77
  - Missing Option 152
  - More Element Tools 83
  - More Project Tools 83
  - New Element Or Connector Context 782
  - Package Control 485
  - Project 87
  - Record & Analyze Context 2546
  - Set Animation 162
  - Set Shadowing 162
  - Settings (Project) 119
  - Sub, Hidden In Toolbox 1543
  - Toolbars (View Menu Option) 83
  - Tools 104
  - Version Control (Package) 427
  - View 83
  - Visual Styles (View Menu Option) 83
  - Window 121
- Menu Option (File Menu)
  - Close Project 79
  - Exit 79
  - New Project 79
  - Open Project 79
  - Page Setup 79
  - Print 79
  - Print Preview 79
  - Print Setup 79
  - Reload Current Project 79



- Menu Option (File Menu)
  - Save Project As 79
- Merge
  - Baseline With Current Model, Overview 457
  - Element 1294, 1322
  - Node 1322
  - Options, Baseline Comparison 470
  - Resource Data 375
- Merge Packages
  - Relationship 1439
- Message 1124
  - Asynchronous Signal 1427
  - Collaboration (Now Communication) 1430
  - Colors In Communication Diagrams 1261
  - Communication 1428
  - Communication, Coloring 638
  - Communication, Create 1429
  - Communication, Re-Order 1430
  - Connector 1417
  - Create In Communication Diagram 1418
  - Create In Sequence Diagram 1418
  - Create On Timing Diagram 1432
  - Endpoint 1322
  - Group, Start New 1430
  - Label 1323
  - Level 1430
  - Move 1417
  - Recursion 1441
  - Relationship 1417
  - Re-Order 1430
  - Scope 1124
  - Self Message 1421
  - Self Message Call 1422
  - Sequence 1418
  - Sequence Communication 1430
  - Sequence Diagram, Asynchronous Signal 1427
  - Sequence Diagram, Examples 1423
  - Sequence Diagram, General Ordering 1426
  - Sequence Diagram, Self Message 1421
  - Sequence, Label Visibility 1258
  - Sequencing 1430
  - Source and Target 1124
  - Timing Diagram 1431
  - WSDL Diagram 2430
  - WSDL Element 2430
- Message Angle
  - Adjust With Duration Constraint 1424
- Message Behavior
  - In Simulation 2490
- Message Flow
  - Rules, BPMN 2.0 1845
- Message Part
  - WSDL Attribute 2431
- MessageFlag Enum
  - Automation Interface 2821
- Meta Object Facility
  - Export Model To XMI 2460
  - Introduction 2456
- Metaclass
  - Add To Profile 1488, 1531
  - Edit In Stereotype Profile 1535
  - Edit Using Profile Helper 1535
- META-INF Package 2029
- Metamodel
  - Elements and Connectors 812
  - Toolbox Pages 812
- Metasequence
  - Code Editor Find And Replace 2161
  - Helper Menu 2161
- Method
  - Add And Delete, Automation Interface Code Example 3000
  - Automation Interface, ElementFeatures Package 2919
  - Context Menu, Project Browser 667
  - Create Use Case For 2155
  - Delete If Not In Code In Reverse Synchronization 2253
  - From Rule Flow Activity 1829
  - Implemented Interfaces 2253
  - Include Bodies In Model When Reverse Engineering 2253
  - Invoke, Object Workbench 2570
  - Link To Use Case 2155
  - Show Parameters On Diagram 831
  - Work With, Automation Interface Code Example 3007
- MethodConstraint
  - Automation Interface, ElementFeatures Package 2923
- Methodology
  - FDD 1769
  - Feature-Driven Design 1769
- MethodTag
  - Automation Interface, ElementFeatures Package 2924
- Metric
  - Automation Interface, Element Package 2898
  - Field Substitution Macros 1669
- Metric Types 523
  - Define 523
  - Global 523
  - Non-Global 520
- Metrics 1169
  - Project Management Window 510

- Metrics And Estimation
  - Default Hour Rate 588
  - ECF 586
  - Effort Types 522
  - Environment Complexity Factors 586
  - For An Element 520
  - Metric Types 523
  - Risk Types 524
  - TCF 585
  - Technical Complexity Factors 585
- MFC Windows
  - Application Pattern 2077
- Microsoft .NET 2200
  - Object Workbench, Set Up 2216
- Microsoft C++
  - Sample Model, Visual Execution Analysis 2529
- Microsoft Common Source Code Control (SCC)
  - Create Environment For Version Control 411
- Microsoft Native
  - Debug Symbols 2191
  - Set Up Debug Sessions 2189
- Microsoft Team Foundation Server
  - Version Control, Create Local Working Copy 409
  - Version Control, Exclusive Checkout 411
- Microsoft Team Foundation Server (TFS)
  - Create Environment For Version Control 408
- Microsoft Word
  - Features In Document Report 2732
  - Open Document Report In 2729
  - Use In Report Documentation 2729
- Microsoft.NET
  - Sample Model, Visual Execution Analysis 2529
- Migrate
  - ArchiMate1.0 Model To 2.0 1927
  - BPMN 1.0 Model To 1.1 1866
  - BPMN 1.1 Model To 2.0 1868
  - SysML 1.1 Model To 1.2 2331
  - SysML 1.2 Model To 1.3 2331
- Migrate()
  - Function, ArchiMate 1927
  - Function, BPMN 1866
  - Function, BPMN 2.0 1868
  - Function, SysML 2331
- MigrateToBPMN11()
  - Function 1866, 2962
- Migration
  - From UML 1.3 597
  - To UML 2.0 597
- Mind Mapping
  - Concept 1794
  - Diagram 1794
- Disable 1794
- Elements 1794
- Enable 1794
- MDG Technology 1794
- Relationship 1794
- Toolbox Page 1794
- MiscData 2881
- Model
  - Activity, BPEL 1.1 1881
  - Activity, BPEL 2.0 1904
  - Add To Project 199
  - Analysis 1794
  - Analysis, Business Processes 1809
  - Automation Interface 2813
  - BPEL 1.1 1871
  - BPEL 1.1 Process 1876
  - BPEL 1.1, Create 1873
  - BPEL 2.0 1894
  - BPEL 2.0 Assign Activity 1912
  - BPEL 2.0 Process 1898
  - BPEL 2.0, Create 1897
  - BPMN 2.0, Serialize In XML 1869
  - Business 1804
  - Business Domain, Create 1824
  - Business Processes, Analysis 1809
  - Business Rules 1814
  - Business Rules, Develop 1816
  - Connecting To Via Cloud Services 262
  - Contents 753
  - Context Menu, Project Browser 648
  - Create Using Model Wizard 753
  - Data Integrity 597
  - Data Object, BPEL 2.0 1909
  - Databases 1937
  - Default Diagram, Cancel 844
  - Default Diagram, Set 844
  - Default Fonts 624
  - Delete 648, 753
  - Delete Element From 921
  - Delete Multiple Elements From 921
  - EAExample, Open 68
  - Gateway, BPEL 1.1 1884
  - Gateway, BPEL 2.0 1906
  - Glossary 533
  - GML, Generate Application Schema 1987
  - Hierarchy, WSDL 2426
  - Integrity Check 597
  - Integrity, Run SQL Patches 601
  - Issues, Record 528
  - MOF 2458
  - Navigation Tools 644
  - Package, Create Using Model Wizard 753

## Model

- Partial ArcGIS Schemas 1962
- Pattern 753
- Pool, BPEL 1.1 1888
- Property, BPEL 2.0 1910
- Quality Control 2593
- Remove Recent 73
- Requirements 1726, 1765
- Requirements, SysML 2325
- Root Node 648
- Search Tools 644
- Sequence Flow, BPEL 1.1 1887
- Sequence Flow, BPEL 2.0 1910
- Sharing, Introduction 307
- Shortcut 204
- Shortcut (Capture Current Environment) 207
- Shortcut (Direct Definition) 206
- Simulation 2463
- Structure 750
- SysML Operational Domain 2326
- SysML Parametric, Create 2320
- SysML System Design 2328
- SysML, Simulate 2322
- Systems Engineering, Create 2288
- Templates, Incorporate In Technology 1576
- Testing 2593
- Trace Tools 644
- Transformation 2017
- Transformations 2013
- UML, Specialized 1789
- Upgrade 599
- What Is A? 753
- WSDL, Binding Element 2437
- WSDL, Create 2426
- WSDL, Development 2423
- WSDL, Document Element 2443
- WSDL, Message Element 2430
- WSDL, Message Part Attribute 2431
- WSDL, Namepace Element 2429
- WSDL, Port Type Operation 2434
- WSDL, PortType Element 2433
- WSDL, Service Element 2441
- WSDL, Structure 2423
- XSD 2387

## Model Branch

- .EAB File 437
- Apply Version Control 426
- Check In 434
- Check Out 433
- Export 437
- File 437
- File, Import 438

- File, Import Manually 439

## Model Changes

- Auditing 446
- Record 446

## Model Default Diagram

- Redisplay 784

## Model Document

- Add Packages As Attributes 2675
- Change Package Sequence 2677
- Delete Package Attributes 2676
- Delete Packages 2676
- Document Order 2677
- Element 2669
- Element, Create 2673
- Generate Documentation 2679
- Generate Report To Document Artifact 2680
- Generate Report To Linked Document 2680
- Identify Search For 2673
- Model Search Sequence 2677
- Move Package Attributes Between Elements 2677
- Overview, Virtual Document 2669
- Select Template For 2673
- Sequence of Model Document Elements 2677
- Sequence of Package Attributes 2677
- Tagged Values 2673

## Model Document (HTML) 2759

## Model Driven Achitecture

- Overview 2013

## Model Driven Development Environment 2172

- Advanced Debug Techniques, Java 2194
- Analyzer Scripts, Manage 2175
- Available Tools 2174
- Basic Setup 2175
- Breakpoint Management 2224
- Build Script, Create 2180
- Build Script, Introduction 2180
- Debug .NET 2200
- Debug .NET CLR Versions 2201
- Debug .NET With COM Interop Process 2202
- Debug Apache Tomcat Server Configuration 2199
- Debug Apache Tomcat Windows Service 2199
- Debug ASP .NET 2203
- Debug Java 2191
- Debug Java Applets In Internet Browsers 2194
- Debug Java Web Servers 2195
- Debug JBOSS Server Configuration 2198
- Debug Symbols, C++ And Native Applications 2191
- Debugger Frameworks 2222
- Debugger System Requirements 2186

- Model Driven Development Environment 2172
  - Debugger, Overview 2222
  - External Tools 2175
  - For C++ Applications 2189
  - For Microsoft Native Applications 2189
  - For WINE Applications 2188
  - General Debug Setup, Java 2191
  - General Workflow 2173
  - Generate Code 2173
  - Getting Started 2173
  - Java Debug Session, Attach To VM 2194
  - Limitations 2074
  - Marker Management 2224
  - Overview 2074
  - Prerequisites 2173
  - Recursive Builds 2180
  - Script Actions, Define 2179
  - Set Up Debug Session 2185
  - Set Up Debug Session For .NET 2200
  - Supported Environments 2074
  - Synchronize Code 2173
  - UAC-Enabled Operating Systems 2187
  - Workspace Layout 2174
- Model Driven Generation
  - Add-Ins 1481
- Model File
  - Connect To ASA Data Repository 239
  - Connect To MySQL Data Repository 225
  - Connect To Oracle 10g Data Repository, Via ODBC 255
  - Connect To Oracle 10g Data Repository, Via OLE DB 257
  - Connect To Oracle 11g Data Repository, Via ODBC 255
  - Connect To Oracle 11g Data Repository, Via OLE DB 257
  - Connect To Oracle 9i Data Repository, Via ODBC 255
  - Connect To Oracle 9i Data Repository, Via OLE DB 257
  - Connect To PostgreSQL Data Repository 233
  - Connect To Progress OpenEdge Data Repository 248
  - Connect To SQL Server Data Repository 218
  - Create Adaptive Server Anywhere Repository 237
  - Create Model 210
  - Create MySQL Repository 222
  - Create Oracle 10g Server Repository 253
  - Create Oracle 11g Server Repository 253
  - Create Oracle 9i Server Repository 253
  - Create PostgreSQL Repository 229
  - Create Progress OpenEdge Repository 245
  - Create SQL Server Repository 217
  - Enterprise Architect Project Files 199
  - Set Up Adaptive Server Anywhere ODBC Driver 237
  - Set Up MySQL ODBC Driver 223
  - Set Up Oracle ODBC Driver 254
  - Set Up PostgreSQL ODBC Driver 230
  - Set Up Progress OpenEdge ODBC Driver 247
  - Transfer To Database 214
- Model Glossary
  - Add Item, Glossary Detail Dialog 534
  - Add Item, Glossary Dialog 535
  - Delete Item, Glossary Dialog 535
  - Delete Item, Project Glossary Tab 534
  - Filter List, Glossary Dialog 535
  - Filter List, Project Glossary Tab 534
  - Glossary Dialog 535
  - Glossary Report 536
  - Modify Item, Glossary Detail Dialog 534
  - Modify Item, Glossary Dialog 535
  - Project Glossary Tab 534
  - Redefine Entry Type, Project Glossary Tab 534
- Model Group Element
  - XML, Toolbox Icon 2410
- Model Mail
  - Create Message 570
  - Forward Message 570
  - Personal Information Window 565
  - Reply To Message 570
- Model Maintenance (.EAP)
  - Compact Project 601
  - Introduction 596
  - Rename Project 601
  - Repair Project 602
- Model Message
  - Dialog 570
- Model Package
  - Publish, XMI 476
- Model Pattern
  - Introduction 755
  - Select In Enterprise Architect 47
- Model Profile
  - Apply 854
  - Save 854
  - Swimlanes Matrix 854
  - Zachman 854
- Model Script
  - Script Debugging 2801
- Model Search
  - Access From Add-In 3019
  - Access From Shortcut 204, 207
  - Access From Shortcut (Direct Definition) 206

## Model Search

- Act On Results 703
- Add Data Columns 700
- Add Filters 716
- Add Items To Linked Document 703
- Add To MDG Technology 1559
- Advanced Search Options 715
- Bookmark Result Items 703
- Conditions 718
- Context Menu 703
- Copy Results To Clipboard 703
- Create Linked Document 703
- Create Search Definition 711
- Customize Search View 708
- Delete Result Items 703
- Display Properties Of Result Item 703
- Export Search Definitions 709
- Fields 718
- Filter Bar 708
- Generate Report 2642
- Generate Report On 703
- Group Results 700
- Hyperlink To 2002
- Import Search Definitions 709
- Introduction 700
- Locate Result Items In Model 703
- Manage 709
- Notes Options 700
- Options 700
- Print Results 703
- Process Results Of Search 700
- Remove Data Columns 700
- Scripts 703, 2794
- Sort And Select 708
- To Populate Model Views 693
- Toolbar Options 700
- Use In Relationship Matrix 731
- Value Grouping 708
- View 700
- View Header 708

## Model Simulation

- Activate Simulation Script 2474
- Breakpoint Sets 2477, 2520
- Breakpoints, Using 2477, 2520
- Edition Support 2466
- How It Works 2466
- Multi-Threaded, Concurrent State Region 2517
- Multi-Threaded, Forks And Joins 2516
- Run Simulation Script 2475
- Set Up Simulation Script 2472
- Supported Platforms 2466
- Topics 2463

## Model State Machine

- Active State Logic 2131
- Designate Driving Triggers 2131
- Establish Port-Trigger Mapping 2131
- For Hardware Description Languages 2131

## Model Template

- Business Process 755, 758
- Class 755, 761
- Component 755, 763
- Database 755, 762
- Deployment 755, 764
- Domain 755, 760
- Introduction 755
- Logical 755, 761
- Maintenance 755, 766
- Physical 755, 764
- Project 767
- Project Management 755
- Relational Database 755, 762
- Requirements 755, 758
- Testing 755, 765
- Use Case 755, 759
- User Interface 755

## Model Transfer

- Introduction 473

## Model Transformations

- Submenu (Tools Menu) 109

## Model Validation

- BPEL 1917
- Cancel 2596
- Configure 2596
- Define Configuration For MDG Technology 1576
- Element Composition Rule 2600
- Execute 2596
- Object Constraint Language 2594
- OCL 2594
- OCL Conformance Rule 2601
- Property Validity Rule 2600
- Rule, Element Composition 2600
- Rule, OCL Conformance 2601
- Rule, Property Validity 2600
- Rule, Well Formedness 2599
- Rules 2597
- Run 2596
- Submenu (Project Menu) 89
- Well Formedness Rule 2599

## Model Validation Broadcasts

- Add-In Model 3047
- EA\_OnEndValidation 3050
- EA\_OnInitializeUserRules 3048
- EA\_OnRunAttributeRule 3053
- EA\_OnRunConnectorRule 3053

- Model Validation Broadcasts
  - EA\_OnRunDiagramRule 3052
  - EA\_OnRunElementRule 3050
  - EA\_OnRunMethodRule 3054
  - EA\_OnRunPackageRule 3051
  - EA\_OnRunParameterRule 3055
  - EA\_OnStartValidation 3049
  - Model Validation Example 3056
- Model View
  - Add To MDG Technology 1558
  - Artifact 1358
- Model View Chart
  - Define 2765
  - Example 2765
- Model Views
  - Automatic Notification 593
  - Context Menu Options 690
  - Define In MTS File 1559
  - Define Search 693
  - Delete (Context Menu) 690
  - Delete (Toolbar) 688
  - Display Recent Postings 693
  - Drag Items Into Linked Document 693
  - Enable Technology-Defined View 693
  - Export As XML 693
  - Favorites 686
  - Favorites Folder, Create (Context Menu) 690
  - Favorites Folder, Create (Toolbar) 688
  - For Requirement Change Management 1785
  - Import As XML 693
  - Monitor Work Flow 593
  - Move Objects Between Views 693
  - Move Objects Into Favorites 693
  - My Views 686
  - Operations 693
  - Postings From Team Review 686
  - Properties 693
  - Recent Discussions 686
  - Recent Post Options 690
  - Refresh (Toolbar) 688
  - Refresh Search 693
  - Root Node, Create (Context Menu) 690
  - Root Node, Create (Toolbar) 688
  - Slide Show, Run (Manual) 695
  - Slideshow 686
  - Slideshow Folder, Create (Toolbar) 688, 695
  - Slideshow, Automate 695
  - Technology-Defined 686
  - Toolbar Options 688
  - Use Objects In Views And Favorites 693
  - View, Create (Context Menu) 690
  - View, Create (Toolbar) 688
  - Views 686
  - Views Folder, Create (Context Menu) 690
  - Views Folder, Create (Toolbar) 688
  - Window 686
- Model Violations
  - Examples 2594
- Model Wizard
  - Add Model Package 753
  - Quick Start 47
  - Use To Create Model 210
- Modeling
  - Behavior Calls 1026
  - Behaviors (General) 1014
  - Enterprise Architect Processes 7
  - Enterprise Architect, Getting Started 42
  - Geodatabases 1944
  - Getting Started - Enterprise Architect 42
  - Invocations 1026
  - Systems Engineering 2288
  - Tasks With Enterprise Architect 7
  - The Business Process 1807
  - Tools and Features 67
  - With Diagrams 778
  - With Elements 900
  - With Enterprise Architect, Overview 752
- Modeling Conventions 2080
  - ActionScript 2 and 3 2081
  - Ada 2005 2082
  - ANSI C 2084
  - C 2084
  - C# 2088
  - C, Object Oriented Programming 2086
  - C++ 2090
  - C++, Managed 2093
  - C++/CLI Extensions 2094
  - Delphi 2095
  - Java 2096
  - Java, AspectJ Extensions 2098
  - Object Oriented Programming in C 2086
  - PHP 2099
  - Python 2100
  - SystemC 2100
  - VB.NET 2102
  - Verilog 2105
  - VHDL 2106
  - Visual Basic 2109
- Modeling Language
  - Extending 1483
- Modeling Tool
  - Enterprise Architect 3
- Models Collection
  - Automation Interface 2850

- Model-View-Controller Pattern 1999
- ModelWatcher
  - Automation Interface Repository 2834
- Modify
  - Element Changes 2625
  - Element Defects 2625
  - Element Issues 2625
  - Element Tasks 2625
  - Linked Document Template 1098
  - Project Task 527
  - Relationship Using Matrix 737
  - RTF Style Template (Legacy) 2726
  - Tagged Values 1137, 1138
  - Z-Order 778
- Modules
  - Analyzer Menu Option 112
  - Window 2243
- MOF
  - Create Diagram 2458
  - Example Diagram 2458
  - Export Model To XML 2460
  - Introduction 2456
- Monitor
  - Events 593
  - Progress 593
  - Work Flow 593
- Monitor Project Tasks
  - Personal Tasks Window 559
- Monitor Workflow
  - Personal Information Window 560
- Mouse/Keyboard
  - Shortcuts 180
- Mouseovers
  - Code Editor, Common 2160
- Move
  - Attributes Between Elements 933
  - Connectors 1112
  - Connectors, Quick Start 57
  - Diagram Sections 869
  - Diagrams, Quick Start 57
  - Element From/To Floating Diagram 837
  - Elements Between Diagrams 59
  - Elements Between Packages 58, 913
  - Elements By Increments 869, 953
  - Elements In Diagrams 911
  - Elements Within Diagram 59
  - Elements Within Packages 57
  - Elements, Quick Start 57
  - Internal Responsibility To External Requirement 1771
  - Maintenance Item Between Categories 2628
  - Nudge Elements 953
  - Objects Between Packages 913
  - Operations Between Elements 933
  - Package Contents Up Or Down 672
  - Package In Project Browser 913
  - Packages, Quick Start 57
  - Submenu 953
  - Test Between Categories 2610
- MS Word
  - Create Report Links 2730
  - Features in Document Report 2732
  - Open Document Report In 2729
  - Update Document Report Links 2735
  - Use In Report Documentation 2729
- MTS File
  - Advanced Options 1559
  - Create 1559
  - Incorporate Model View 1559
  - Working With 1559
- Multi-page Diagram
  - Print 871
- Multiple Element
  - Selection, Drag Onto Diagram 833
- Multiple Login
  - Under One User ID 320
- Multiple Select
  - Items From Project Browser 833
- Multiple Stereotype
  - Restrict Application Of 1509
- Multiplicity 755, 760
  - Connector, Source Role 1130
  - Connector, Target Role 1132
  - Of Connector 1151
  - Of Element 1151
  - Of Element, Displayable 944
  - Of Element, Non-Displayable 959
- MVC Pattern 1999
- MyISAM
  - BaseModel Script 220
- MySQL
  - Create Repository 222
  - Data Repository, Connect To 225
  - Datatypes For 2352
  - ODBC Driver, Set Up 223
  - Set Up Project On Repository 220
  - Table Type, Set 2342

## - N -

- Name Direction Indicator 866
- Name Template
  - Foreign Key, Define 2362

- Name Template
  - Primary Key 2355, 2357
- Namespace
  - And Whitespace Characters 2120
  - Clear 2120
  - Dialog 2120
  - Explanation 2120
  - List 2120
  - Locate In Project Browser 2120
  - ODM Package, Define New 1980
  - Root 2120
  - Root Package Icon 670
  - Set 2120
  - WSDL Element 2429
- Naming Format
  - Camel Case 2067
  - Pascal Case 2067
  - Spaced 2067
  - Underscored 2067
- N-Ary
  - Association Element 2007
  - Element (Entity Relationship Diagram) 1939
- Navigate
  - Diagram 698
- Navigation And Selection
  - Hotkeys, For Diagram 841
- Navigation Compass
  - For Docking Windows 128
- Navigation Tools 644
- Neaten
  - Diagram Layout 885
- Nested Package
  - Version Control Of 386
- Nested Version Control Packages 390
- Nesting
  - Connector 1434
  - Recording Markers 2548
  - Relationship 1434
- New Action Dialog 1268
- New Code Sections
  - Add To Existing Features 1640
- New Project
  - Menu Option (File Menu) 79
- New Search Query 711
- New Structured Activity Dialog 1338
- News Letters
  - Sparx System 68
- Node
  - Element 1378
- Nodes
  - AST, In Grammars 1711
- Non-Selectable
  - Element 939
- Normal.rtf
  - Template File 1058
- Notation
  - Co-Region 1427
  - Data Modeling 2384
  - IDEF1X 2384
  - Information Engineering 2384
  - Process Modeling 1810
  - UML 2.1 2384
- Note
  - Add To Link/Connector 1111
  - Attach To Connector 898
  - Compartment Text Color 616
  - Create Project Glossary Item 1142
  - Element 1324
  - Element Border Color 616
  - Element Fill Color 616
  - Element Text Color 616
  - Element, Create 923
  - For Attribute 1142
  - For Connector 1142
  - For Diagram 1142
  - For Element 1142
  - For Operation 1142
  - Formatting 1143
  - Hyperlink Glossary Term 1142
  - Insert Glossary Item 1142
  - Insert New From Toolbar 140
  - Keyboard Shortcuts 1143
  - Link To Connector 944
  - Link To Connector Tagged Value 944
  - Link To Diagram Note 944
  - Link To Element Feature 924
  - Link To Internal Documentation 924
  - Spell Check 1142
  - Tab 1142
  - Toolbar Options 1143
  - Window 1142
- Notelink
  - Connector 1435
  - Insert New From Toolbar 140
  - Relationship 1435
- Notes
  - Gap, Edit 748
- Nudge
  - Elements 869, 953
- Numbered Sections
  - In Document Reports 1081
- Numbering Levels
  - Apply 1081
  - Document Reports 1081



## Numbering Levels

User Defined 1081

## Numbering List

Document Reports 1081

## Numeric Range Generator

Timeline Element States 1237

## NUnit Transformation

MDA-Style Transform 2038

**- O -**

## Object

Appearance, Options 631

Attributes, In Transformation 2055

Classes, In Transformation 2055

Classifiers 1009

Classify According To Class Associations 1012

Columns, In Transformation 2055

Connect 1109

Connector 964

Create From Attribute 1008

Create In MDA-Style Transformation 2055

Create In Simulation Manually 2480

Create In Simulation With CreateObject Action  
2480

Create In Simulation With JavaScript 2480

Default Variables, In Simulation 2479

Definition, In Transformation 2055

Destroy In Simulation With DestroyObject Action  
2483

Destroy In Simulation With JavaScript 2483

Element 1379

Elements And Connectors 804

Features 1009

In Simulation 2479

Instance 1379

Instantiate Object Member, In Simulation 2480

Links 964

Methods Of Classifying 1010

Move Between Diagrams 59

Move Between Packages 58, 913

Move Within Diagram 59

Move Within Package 57

Multiplicity 1130

Operations, In Transformation 2055

Packages, In Transformation 2055

Parameters, In Transformation 2055

Properties 956

Properties In Transformation 2055

Relationships 964

Scenario 965

State, Set 1381

Tables, In Transformation 2055

Toolbox Pages 804

Transformation 2055

Type, Change For Element 918

Type, In Transformation 2055

Types In Transformation 2055

## Object Action

CallBehavior Action 2480

CreateObject Action 2480

## Object Constraint Language

Model Validation 2594

Model Validation Rules For Conformance 2601

## Object Diagram

Description 1186

Elements And Connectors 1186

Example 1187

## Object Flow

Connector 1435

In Activity Diagram 1436

In State Machine Diagram 1435

Multiple 1436

Relationship 1435

Selection Behavior 1436

Simple 1436

Transformation Behavior 1436

With Action Pins 1436

## Object Node

Element 1325

On Activity Element 1325

## Object Oriented Programming

C Code Generation For UML Model 2086

Limitations 2086

## Object Run State Diagram 2591

## Object Workbench

Introduction 2567

Invoke Method 2570

Java 2216

Microsoft .NET 2216

Modes 2567

Overview 2567

Setup 2216

Workbench Variables, Constructors 2568

Workbench Variables, Create 2568

## ObjectType Enum

Automation Interface 2822

## Occurrence

Connector 1438

Element 1317

Relationship 1438

## OCL

Model Validation 2594

## OCL Constraints

- Attribute 2601
    - Element 2601
    - Feature 2601
    - Model Validation Rules for Conformance 2601
    - Relationship 2601
- ODBC
  - Data Modeling 2380
- ODBC Data Source
  - Select 2378
- ODBC Source
  - Select Database Objects From 2378
  - Select Stored Procedures From 2378
  - Select Tables From 2378
- ODM
  - Commands 1980
  - Example Diagrams 1979
  - Export OWL/RDF 1980
  - Import OWL/RDF 1980
  - MDG Technology For 1971
  - OMG 1971
  - OWL Definition Diagram 1979
  - OWL Elements, Descriptions 1974
  - OWL Facts Diagram 1979
  - OWL Properties, Descriptions 1974
  - OWL Relationships, Descriptions 1974
  - OWL Semantics, Descriptions 1974
  - RDF Definition Diagram 1979
  - RDF Elements, Descriptions 1977
  - RDF Relationships, Descriptions 1977
  - Toolbox Pages 1972
- Offline
  - Checkout 392
  - Version Control 392
- OLE DB Provider
  - Microsoft For Oracle 257
  - Oracle Provider 257
- Online Resources
  - Submenu 123
- Ontology Definition Metamodel
  - MDG Technology For 1971
- Open
  - External Tools 158
  - Package From Project Browser 774
  - Package Within Diagram 845
- Open A Project 202
- Open Project
  - Menu Option (File Menu) 79
- Open Repository
  - Automation Interface Code Example 2995
- Open Services for Lifecycle Collaboration
  - Base Requirements 268
  - Create Requirement 280
  - Creation Factory Service 280
  - Introduction 268
  - PREFIX Parameter 279
  - PROPERTIES Parameter 278
  - Property Constraints 271
  - Provider 269
  - Query Capability 273
  - Query Capability, PREFIX 279
  - Query Capability, PROPERTIES 278
  - Query Capability, SELECT 276, 277
  - Query Capability, WHERE 275, 277
  - Query Parameters 273
  - Query Response 273
  - Requirement Creation 280
  - Requirement Properties 271
  - Requirements Management 268
  - Resource Elements 269
  - Resource Shape 271
  - SELECT Parameter 276
  - Service Provider 269
  - Service Provider Resource 269
  - Use Of Cloud Services 267
  - WHERE & SELECT, Combine 277
  - WHERE Parameter 275
- Open Source Directory
  - Element Code 103
- Open Source File
  - Menu Option 104
- Operation
  - Add To Element, In-place Editor 1040
  - Advanced Properties 1020
  - Appearance 1014
  - As Action 1266, 1275
  - Associate With Behavior 1018
  - Behavior Description, Show In Diagram 1018
  - Behavior, Initial Code 1019
  - Business Rule 1830
  - Connect To 1110
  - Constraints 1023
  - Context Menu, Project Browser 667
  - Copy Between Elements 932
  - Create Use Case For 2155
  - Definition 1014
  - Dialog, Behavior Tab 1018
  - Dialog, General Tab 1015
  - Dialog, Post Tab 1023
  - Dialog, Pre Tab 1023
  - Disconnect From 1110
  - Display Inherited Operation 1024
  - Drag Onto New Element 989
  - Edit Name, In-Place Editor 1034

## Operation

- Edit Parameter Kind 1039
- Edit Scope 1036
- Edit Stereotype, In-place Editor 1035
- Element Feature 1014
- Fast Create 1015
- Field Substitution Macros 1669
- Implement Interface Operations 1023
- Implement Parent Operations 1023
- In Project Browser 1014
- Inherited, Show 845
- Introduction 1014
- Link To Use Case 2155
- Modifiers 1020
- Move Between Elements 933
- Of State Element, Behavior 1018
- Override Interface Operations 1023
- Override Parent Operations 1023
- Parameter Keyword, Edit 1038
- Parameter, By Reference 1031
- Parameter, Insert In Element 1041
- Parameters, Define 1028
- Private, Icon 670
- Properties, Behavior 1018
- Properties, General 1015
- Properties, Initial Code 1019
- Properties, Postconditions 1023
- Properties, Preconditions 1023
- Protected, Icon 670
- Raised Exception 1020
- Redefined 1020
- RuleTask, Behavior Call Action 1830
- Show On Diagram 845
- Tagged Values, Add 1022
- WSDL Port Type Operation 2434

## Operational Domain Model

- SysML 2326

## Options

- Audit 448
- Compare Utility 464
- Element Visibility 631
- Read-Only, For Enterprise Architect 23
- Reset For A Class 2279
- Themes 168
- Visual Styles 168

## Options Dialog

- ActionScript 2264
- Ada 2005 2264
- Appearance, Diagram 622
- ArcGIS 2265
- Attribute/Operation Specifications 2253
- C 2266

- C# 2268
- C++ 2269
- Communication Message Coloring 638
- Connector Settings 635
- Custom Colors, Define 619
- Delphi 2271
- Diagram Behavior 625
- Diagram Settings 608
- General Settings 605
- Gradients and Backgrounds, Diagram 613
- Introduction 604
- Java 2272
- Links Settings 635
- MDG Technology 2278
- Object Appearance 631
- Object Appearance, Default Fonts 624
- Object Appearance, Element Visibility 631
- PHP 2274
- Python 2273
- Sequence Diagram Options 629
- Standard Colors 616
- SystemC 2275
- VB.NET 2275
- Verilog 2276
- VHDL 2277
- Visual Basic 2277
- XML Specifications 639

## Oracle

- ODBC Driver, Set Up 254
- Package, Create 2374
- Sequence 2345
- Sequence Options, DDL For Packages 2381
- Sequence Options, DDL For Table 2380
- Set Datatypes 2352
- Set Up Project On Repository 250
- Tables, Set Properties 2343
- Tables, Tagged Values 2343
- Temporary Table 2343

## Oracle 10g

- Data Repository, Connect To, Via ODBC 255
- Data Repository, Connect To, Via OLE DB 257
- Server Repository, Create 253

## Oracle 11g

- Data Repository, Connect To, Via ODBC 255
- Data Repository, Connect To, Via OLE DB 257
- Server Repository, Create 253

## Oracle 9i

- Data Repository, Connect To, Via ODBC 255
- Data Repository, Connect To, Via OLE DB 257
- Server Repository, Create 253

## Order

- Enterprise Architect 33

- Order
    - Package Contents 672
  - OSLC
    - Base Requirements 268
    - Create Requirement 280
    - Creation Factory Service 280
    - Introduction 268
    - PREFIX Parameter 279
    - PROPERTIES Parameter 278
    - Property Constraints 271
    - Provider 269
    - Query Capability 273
    - Query Capability, PROPERTIES 278
    - Query Capability, SELECT 276, 277
    - Query Capability, WHERE 275, 277
    - Query Capability, PREFIX 279
    - Query Parameters 273
    - Query Response 273
    - Requirement Creation 280
    - Requirement Properties 271
    - Requirements Management 268
    - Resource Elements 269
    - Resource Shape 271
    - SELECT Parameter 276
    - Service Provider 269
    - Service Provider Resource 269
    - Use Of Cloud Services 267
    - WHERE & SELECT, Combine 277
    - WHERE Parameter 275
  - Outline
    - Red 340
  - Output
    - Business Modeling 1812
    - Debugger, View 2231
    - Debugger, Window 2231
  - Output Window 169
    - Audit History Tab 455
  - Overdue Task
    - Highlight 594
  - Overdue Tasks
    - Highlight 538, 542
  - Overlay
    - BREAD Matrix 732
    - CRUD Matrix 732
    - Relationship Matrix, Apply 734
    - Relationship Matrix, Create 733
    - Relationship Matrix, Delete 733
    - Relationship Matrix, Edit 733, 734
    - Relationship Matrix, Introduction 732
    - Relationship Matrix, Remove 734
  - Override
    - Attribute Initializer 1007
    - Default Toolbox In Toolbox Profile 1567
    - Implement Parent Operations 1023
    - Parent Operations 1023
  - Overrides & Implementations
    - Of Classes And Interfaces 100
  - Overview
    - Visual Execution Analyzer 2527
  - OWL
    - Definition Diagram 1979
    - Elements 1972
    - Elements, Descriptions 1974
    - Example Diagrams 1979
    - Export To File 1980
    - Facts Diagram 1979
    - Import From File 1980
    - Profile 1971
    - Properties 1972
    - Properties, Descriptions 1974
    - Relationships 1972
    - Relationships, Descriptions 1974
    - Semantics 1972
    - Semantics, Descriptions 1974
    - Toolbox Pages 1972
- P -**
- Package
    - Add And Manage, Automation Interface Code Example 2996
    - Add Element Directly 903
    - Add To Diagram From Toolbox 772
    - Add To Project Browser 772
    - Add To UML Model, Quick Start 50
    - Apply Version Control To Model Branch 426
    - As Bookmark 2730
    - Automation Interface 2813
    - Automation Interface Repository 2835
    - Baselines 457
    - Batch Export To XMI 492
    - Batch Import From XMI 493
    - Body, For Oracle 2374
    - Check In Multiple 434
    - Check In Single 432
    - Check Out Multiple 433
    - Check Out Single 431
    - Comparison, Example 468
    - Configuration 488
    - Configure For Version Control 424
    - Context Menu, Project Browser 651
    - Control, Menu 485
    - Control, Remove 490
    - Controlled 424

## Package

- Controlled, Report Deletion Of Cross Package References in XML 496
- Copy 775
- Copy Between Projects 506
- Create Diagram 822
- Create Oracle Packages 2374
- CSV Import/Export Specification 498
- Delete In Project Browser 777
- Duplicate 775
- Element 1382
- Exclude From Report Filters, Custom Script 2658
- Exclude From Report Filters, SQL Query 2658
- Existing, Drag Onto Diagram 776
- Export 484
- Export To XML 475
- Export To XML Stubs 488
- Export Version Controlled Model Branch 437
- Field Substitution Macros 1671
- GML 1984
- Hide Contents On Diagram 777
- Icon Overlays 670
- Import 484
- Import From EMX/UML2 480
- Import From XML 478
- Import Version Controlled Model Branch 438, 439
- Load 492
- Lock 336
- Lock, Require User Lock 339
- Lock, User/Group Locking 337
- META-INF 2029
- Modeling With 772
- Move 57
- Move Element Between 913
- Move Elements Between 58
- Move Elements Within 57
- Move In Project Browser 913
- Move Object Between 913
- Nested in Version Control 390
- Open From Diagram 845
- Open From Project Browser 774
- Paste As Chart Object (Graph) 835
- Paste As Element 835
- Paste As Matrix Specification 835
- Paste As Model View List 835
- Paste As Report Specification Link 835
- Paste Copy 775
- Paste From Project Browser 835
- Phase, Update 590
- Profile 1487

- Publish, XML 476
- Release Lock, User/Group Locking 337
- Rename 774
- Resynchronize Package Version Control Status 444
- Review Version Control History (SCC Example) 442
- Review Version Control History (SCC) 441
- Review Version Control History (TFS, CVS, Subversion) 440
- Save 491
- Save Profile 1525
- Set As Source For Standard Chart 2771
- Show Contents On Diagram 777
- Specification, For Oracle 2374
- Status, Update 590
- Synchronize Contents 2117
- Tasks 772
- Topology, ArcGIS 1954
- Undo Check Out, Version Control 432
- Update All To Latest Revision 435
- Update Contents 2117
- Update To Latest Revision 435
- Validate Version Control Configuration 443
- Version Control 383
- Version Control Revision, Check Out 440
- Version Control, Update All To Latest Revision 435
- Version Control, Update To Latest Revision 435
- Version, Update 590
- Working With 772
- XML 484
- XML Schema 2390

## Package Baseline

- Comparison In Reusable Asset Service 291

## Package Browser

- Context Menu Options 679
- Description 673
- Filter Bar 677
- Gantt View 673
- Generate Report 2642
- Insert Tagged Value Column 679
- Layout Profiles 679
- Options 679
- Toolbar 679
- Value Grouping 677
- View Header 677
- Work On Elements 679

## Package Context Menu

- Advanced Submenu, Project Browser 654
- Code Engineering Submenu, Project Browser 657

- Package Context Menu
  - Contents Submenu, Project Browser 655
  - Copy/Paste Submenu, Project Browser 654
  - Documentation Submenu, Project Browser 656
  - Import/Export Submenu, Project Browser 659
- Package Control
  - Options, Dialog 488
- Package Diagram
  - Description 1182
  - Elements And Connectors 1182
  - Example 1183
- Package Import
  - Connector 1438
  - Relationship 1438
- Package Merge
  - Connector 1439
  - Relationship 1439
- Package Scenarios
  - Import As Test Scenarios 2611
- Package Status
  - Update, Package Context Menu 654
- Package Structure
  - BPEL 1.1 1873
  - BPEL 2.0 1897
- Package View
  - Of Specification Manager 1759
- Packaging Component
  - Element 2008
- Page
  - Border, Show/Hide On Diagram 825
  - Footer, Print On Diagram 825
  - Header, Print On Diagram 825
  - Setup, Menu Option (File Menu) 79
  - Size, Set For Diagram 870
- Pan
  - Diagram View 868
- Pan And Zoom
  - Diagram 698
  - Window 698
- Paragraph Numbering
  - In Document Editor 1064
- Parameter 1031
  - Activity 1028
  - Automation Interface, ElementFeatures Package 2925
  - Behavior 1028
  - Behavior Call Arguments 1027
  - Behavior, Tagged Values 1030
  - Behavioral, Edit 1028
  - Behavioral, Extend 1028
  - Behavioral, Reassign 1028
  - Behavioral, Set 1028
  - Behavioral, Synchronize Call Argument With 1027
  - Behavioral, Synchronize Invocation Argument With 1027
  - Dialog 1028
  - Field Substitution Macros 1672
  - For Rule Flow Activity 1829
  - Interaction 1028
  - Kind Inout, By Reference 1031
  - Operation 1028
  - Operation, By Reference 1031
  - Reference 1031
  - Show Details On Diagram 831
  - Substitution 1445
  - Variables In Business Rules 1829
- Parameter Information
  - Display In Source Code 2149
- Parameter Kind
  - Edit for Element Operation 1039
- Parameter Substitution
  - Binding Connector 1445
- Parameterized Class
  - And Binding Connector 1444
- Parameterized Classes (Templates) 1365
- Parametric
  - Diagram 2320
  - Diagram, Simulate SysML Model 2322
  - Model, SysML, Create 2320
- ParamTag
  - Automation Interface, ElementFeatures Package 2927
- Parent
  - Confirm Element As Parent 911
  - Set For Element 908
- Parent And Interface
  - Elements 100
- Parse
  - Source Code Files In Viewer 2146
- Part
  - Add Property Value 1383
  - Bind To Association (SysML Block) 2304
  - Compartments Of Block Element 2305
  - Compartments Of Constraint Blocks 2305
  - Delete When Association Deleted 2304
  - Element 1383
  - Generate From Association (SysML Block) 2304
  - Property Tab 1386
  - Qualifiers 1386, 1396
  - Redefined 1386
  - Represent On Sequence Diagram 1258
  - Subsetted 1386

- Partial Class
  - Generate 2115
- Partial Schema
  - ArcGIS, Exporting 1962
  - ArcGIS, Modeling 1962
- Partition
  - Activity 1325
  - Docking 1325
  - Element 1325
  - Horizontal 1325
  - Vertical 1325
- Partitions Collection
  - Automation Interface, ElementFeatures Package 2928
- Pascal Case
  - Naming Format 2067
- Password
  - Administrator Change 334
  - Administrator Set 334
  - Security, Change 323
  - Set On Team Review Item 357
  - User Change 334
- Password Encryption
  - Prior To Release 7.1 Of Enterprise Architect 333
- Password Encryption (Repository)
  - At Release 7.1 Of Enterprise Architect 204, 209
- Paste
  - Connectors Between Pasted Instances 833
  - Copy Of Element 915
  - Copy Of Package 775
  - Diagram Into Package 842
  - Element From Project Browser 833
  - Element On Diagram 836
  - Multiple Elements From Project Browser 833
  - Multiple Items As Children 833
  - Multiple Items As Instances 833
  - Multiple Items As Links 833
  - Multiple Items From Project Browser 833
  - Object As Child 833
  - Object As Link 833
  - Object As New Instance 833
  - Package As Chart Object (Graph) 835
  - Package As Element 835
  - Package As Matrix Specification 835
  - Package As Model Model View List 835
  - Package As Report Specification Link 835
  - Package From Project Browser 835
- Paste Element Dialog 833
- Paste Elements
  - As Link 82
  - As New 82
  - From Clipboard As Metafile 82
  - Menu Option (Edit Menu) 80
  - Submenu (Edit Menu) 82
- Patches
  - SQL, Run 601
- Pattern
  - Action, Modify 1467
  - Actions 1464
  - Add To MDG Technology 1549
  - Application, Generate 2077
  - Create From Diagram 1464
  - Default, Change 1467
  - Design 1464
  - GoF, Download 1466
  - Import Into Model 1466
  - In Resources View 1464
  - Model-View-Controller 1999
  - MVC 1999
  - Save 1464
  - Save From Diagram 1464
- PDATA
  - Diagram Profile Attribute Values 1573
  - Element Attribute In MiscData, Object Model 2881
- People
  - As Project Resources 1156
  - Assign To Changes 2631
  - Assign To Issues 2631
  - Dialog 1152
  - Project Types Menu Option 120
  - Settings 1153
- Per Page
  - Diagram Layout 881
- Performance
  - Of Enterprise Architect 218, 225, 233, 239, 248, 255, 257, 259, 446
- Permission List
  - User Security 329
- Personal Information Window
  - Execute Workflow Scripts 560
  - Workflow Tab 560
- Personal Tasks
  - Allocated Work 554
  - Gantt Chart 554
  - Project Tasks 554
  - View 554
  - Workflow 554
  - Working Sets 554
- Personal Tasks View
  - Allocated Work Tab 555
  - Gantt Chart 555

- Personal Tasks View
  - Record Work Progress 555
  - Resource Allocation 555
- Personal Tasks Window
  - Working Sets Tab 561
- PHP
  - Code Generation 2274
  - Import, Reverse Engineering 2139
  - Language Options 2274
  - Modeling Conventions 2099
  - Transformation 2040
- PHP Apache
  - Sample Model, Visual Execution Analysis 2529
- PHP Debugger
  - Checklist 2207
  - Local 2204
  - Remote 2204
  - Scenarios 2204
  - Script Set Up 2204
  - Supported Operating Systems 2206
  - System Requirements 2206
  - Timeouts 2207
- Physical Data Model 2335
- Physical Model 755, 764
- PI
  - Macro 1687
  - Processing Instruction 1687
- PI Macro
  - Code Template Syntax 1683
- Pie Chart
  - Change Appearance 2782
- PIM
  - Internal Bindings 2013
- Pin
  - Action 1277
  - Add To Action 1277
  - As Action Property 1278
  - As Argument For Call Action 1277
  - Assign To Action 1277, 1278
  - Connector End Point 143
  - Connector Start Point 143
  - Properties 1277
  - Toolbox Pages 797
- Pkg Import
  - Connection 1438
  - Relationship 1438
- Pkg Merge
  - Connector 1439
  - Relationship 1439
- Place Recording Markers
  - Execution Analysis, Recording Sequence Diagrams 2536
- Platform Naming Conventions 2067
- Platform Specific Model 2013
- Platform-Independent Model 2013
- Pool
  - BPMN, Use By BPEL 2.0 Model 1911
  - Orientation, BPMN 2.0 1845
- Pool (BPEL 1.1)
  - Create 1888
  - Model 1888
- Port
  - Add To Element 1385
  - Compartments Of Block Element 2305
  - Element 1384
  - Flow Direction (SysML) 2306
  - For Trigger Element 1350
  - Inherited By Component Instance, From Classifying Component 1385
  - Inherited By Component, From Realized Interface 1385
  - Property Tab 1386
  - Qualifiers 1386, 1396
  - Redefined 1385, 1386
  - Represent On Sequence Diagram 1258
  - Subsetted 1386
- Port Type Operation
  - WSDL 2434
- PortType
  - WSDL Diagram 2433
  - WSDL Element 2433
- Position
  - Options For Elements In Diagram 97
- Position Elements 953
- Post
  - Edit In Team Review 358
- PostgreSQL
  - Data Repository, Connect To 233
  - ODBC Driver, Set Up 230
  - Repository, Create 229
  - Set Up Project On Repository 227
- Post-New Events
  - Add-In Model 3060
  - EA\_OnPostNewAttribute 3064
  - EA\_OnPostNewConnector 3062
  - EA\_OnPostNewDiagram 3063
  - EA\_OnPostNewDiagramObject 3063
  - EA\_OnPostNewElement 3061
  - EA\_OnPostNewGlossaryTerm 3067
  - EA\_OnPostNewMethod 3065
  - EA\_OnPostNewPackage 3066
- Pre New-Object Events
  - Add-In Model 3075
- Pre/Post Conditions



- Pre/Post Conditions
  - Constraints 1274
  - Local 1274
  - Notes 1274
  - On Actions 1274
- Precede
  - Connector 801
- Predefined Tag Type
  - Assign To Stereotype 1499
  - Define 1499
- Predefined Tagged Value Type
  - Filters 1622
  - Reference Data 1629
  - Structured 1622
  - Syntax 1622, 1629
- Pre-Deletion Events
  - Add-In Model 3067
  - EA\_OnPreDeleteAttribute 3069
  - EA\_OnPreDeleteConnector 3071
  - EA\_OnPreDeleteDiagram 3071
  - EA\_OnPreDeleteDiagramObject 3072
  - EA\_OnPreDeleteElement 3068
  - EA\_OnPreDeleteGlossaryTerm 3074
  - EA\_OnPreDeleteMethod 3070
  - EA\_OnPreDeletePackage 3073
- PREFIX Parameter
  - OSLC Query Capability 279
- Pre-New Events
  - EA\_OnPreDropFromTree 3079
  - EA\_OnPreNewAttribute 3080
  - EA\_OnPreNewConnector 3077
  - EA\_OnPreNewDiagram 3078
  - EA\_OnPreNewDiagramObject 3078
  - EA\_OnPreNewElement 3076
  - EA\_OnPreNewMethod 3081
  - EA\_OnPreNewPackage 3082, 3083
- Preprocessor Macros 2257
- Preserve Hierarchy
  - CSV Specification 500
- Primary Key
  - Complex 2355
  - Create 2355, 2356
  - Description 2355
  - Extended Properties 2358
  - Name Template, Define 2355, 2357
  - Simple 2355
  - SQL Server Fill Factor 2355
  - SQL Server, Non-Clustered 2358
- Primitive
  - Element 1386
- Print
  - Diagram (Single) From Project Browser 665
  - Diagrams (Multiple) From Project Browser 665
  - Menu Option (File Menu) 79
  - Multi-page Diagrams 871
  - Page Footer On Diagram 825
  - Page Header On Diagram 825
  - Preview, Menu Option (File Menu) 79
  - Project Issues 528
  - Scaled Image 871
  - Setup, Menu Option (File Menu) 79
  - Task List 526
- Print Preview
  - Display 783
  - Multiple Pages 783
- Priority Field
  - Value Type, Define 1164
- Private Key
  - Add 3165
- Private Model
  - Version Control Usage 385
- Problem
  - Field Substitution Macros 1673
- Problem Type
  - Define 1169
- Process
  - BPEL 1.1, Properties 1876
  - BPEL 2.0, Model 1898
  - Element 1810, 2008
  - Element (Data Flow Diagram) 1797
  - Model Template 755, 758
  - Modeling 755, 758
- Process First Chance Exceptions 2244
- Process Memory
  - Inspect 2242
- Process Modeling Notation 1810
- Processing Instruction
  - PI Macro 1687
- Professional Edition
  - Of Enterprise Architect 20
- Profile
  - Add Attribute To Diagram 1472
  - Add Connector To Diagram 1472
  - Add Element To Diagram 1472
  - Add Elements 1488, 1491, 1531
  - Add Enumeration Elements 1493
  - Add Metaclasses 1488, 1531
  - Add Non-UML Objects 1491
  - Add Operation To Diagram 1472
  - Add Quick Linker Definition 1523
  - Add Shape Script 1501
  - Add Stereotypes 1488, 1491, 1531
  - Add Structured Tagged Value Class 1495
  - Add To MDG Technology 1548

## Profile

- And Element Templates 929, 1472, 1485
- Application, Analyzer Menu Option 112
- Create 1485, 1487
- Create, Using Helpers 1528
- Diagram, Create 1536, 1570
- Edit Metaclasses, Using Profile Helper 1535
- Edit Stereotypes, Using Profile Helper 1535
- Elements And Connectors 810
- Export To Disk 1523
- Extending UML 1471
- Grammar Parsing 1722
- Import From XML 1472, 1485
- Import To Resources Window 1526
- In Resources Window (Not Recommended) 1526
- Metaclass, Edit 1535
- Package 1487
- Relationship Matrix, Contents In Document Report 2692
- Relationship Matrix, Create In Specification Manager 1744
- Save 1523
- Save From Diagram 1525
- Save From Package 1525
- Save Package As UML Profile 654
- Set Default Appearance Of Stereotype Objects 1502
- Stereotype 1487
- Stereotype Constraints 1500
- Stereotype Tagged Values 1473
- Stereotype, Create With Profile Helper 1529
- Stereotype, Edit 1535
- Stereotypes 1472, 1485
- Swimlanes Matrix 854
- Tags 1492
- Toolbox 1560
- Toolbox Pages 810
- Toolbox, Create 1538
- Work With 1485, 1529
- Zachman 854

## Profile Connector

- Add To Diagram 1472

## Profile Diagram

- Description 1196
- Example 1197

## Profile Element

- Add To Diagram 1472

## Profile Helpers 1528

## Profiler

- Attach To Process 2564
- Function Line Report 2564

- Getting Started 2558

- Launch 2564

- Operation 2557

- Overview 2555

- Prerequisites 2556

- Report, Example 2555

- Report, Load 2560

- Report, Save 2560

- Report, Save As Resource In Team Review 2566

- Save Profile Report - Document Artifact 2560

- Save Profile Report - Team Review Resource 2560

- Set Options 2563

- Set Sample Intervals 2563

- Start 2564

- Stop 2564

- Supported Platforms 2556

- Switch To Debugger From 2558

- System Requirements 2556

- Team Review, Save Report As Resource 2566

- Toolbar 2558

## Programming Language

- Custom 1581

- Develop 1581

## Progress OpenEdge

- Data Repository, Connect To 248

- ODBC Driver, Set Up 247

- Server Repository, Create 245

- Set Up Project On Repository 243

## Project

- Administration, Security Permissions 318

- Author 1153

- Browser 646

- Change Control 2619

- Clean 597

- Clean Prior To Build 2181

- Clients 1157

- Compact .EAP File 601

- Comparison With Other Project 507

- Configure 210

- Connect To Via Cloud Services 266

- Constants 2664

- Copy Existing 212

- Create In Enterprise Architect, Tutorial 47

- Create, Quickstart Tutorial 47

- Data Integrity 597

- Data, Transfer 504

- Develop In Team Environment, Introduction 306

- Development, Introduction 197

- EABase 43

- Estimation 584

## Project

- Explorer 646
- File, EABase 210
- Glossary 533
- Integrity Check 597
- Integrity, Run SQL Patches 601
- Issues 525
- Issues, Record 528
- Items, Add Via Toolbar 136
- Maintenance 2619
- Manage 68
- Menu 87
- Metrics 584
- Model Template 767
- Open Existing 199
- Quality Control 2593
- Recover 597
- Remove Recent 73
- Rename .EAP File 601
- Repair Project .EAP File 602
- Resources 1156
- Roles 1155
- Set Up On Access 2007 Repository 241
- Set Up On DBMS Repository 214
- Set Up On MySQL Repository 220
- Set Up On Oracle Repository 250
- Set Up On PostgreSQL Repository 227
- Set Up On Progress OpenEdge Repository 243
- Set Up On SQL Server Repository 215
- Set Up On Sybase ASA Repository 234
- Share, DBMS Repository 308
- Share, Network Drive 308, 309
- Share, Replication 308
- Sharing, Introduction 307
- Spell Checking 549
- Statistics, View 87
- Status View 525
- Structure 750
- Tasks 183, 525
- Testing 2593
- Timescale Estimation 584
- Toolbar 136
- Upgrade 599
- View 646
- What Is A? 200

## Project Author

- Add In Specification Manager 1758

## Project Browser

- Attribute Context Menu 667
- Code Engineering Context Menu, Add Submenu 664
- Collapse Contents 655

- Context Menus 648
- Copy Element References 663
- Copy Package References 654
- Copy/Paste Element Options 663
- Copy/Paste Package Options 654
- Cut/Paste Element Options 663
- Cut/Paste Package Options 654
- Default Behaviour 672
- Diagram Context Menu 665
- Element Context Menu 660
- Element Context Menu, Add Submenu 662
- Exclamation Marks 340
- Expand Contents 655
- Free Sorting 672
- Hide And Show 646
- Icon Overlays 670
- Icon, User-Defined 1503
- Introduction 646
- Method Context Menu 667
- Model Context Menu 648
- Move Items Within 646
- Open Package 774
- Operation Context Menu 667
- Order Package Contents 646
- Package Context Menu 651
- Package Context Menu, Advanced Submenu 654
- Package Context, Code Engineering Submenu 657
- Package Context, Contents Submenu 655
- Package Context, Documentation Submenu 656
- Package Context, Import/Export Submenu 659
- Reload Current Package 655
- Reset Sort Order 655
- Scripts 2794
- Selective Collapse of Packages 646
- Show Level Numbering 651
- Show Stereotypes 672
- Toolbar 669
- Version Control Indicators 391
- Views 646

## Project Calendar

- Allocated Resources View 578
- Calendar Panel 571
- Calendar Panel, Configure Event Subtypes 577
- Diary Panel 571
- Events 574
- Options 571
- Overview 571
- Project Tasks View 581
- Recurring Events 574

- Project Calendar
  - Toolbar 571
  - View, Calendar 574
- Project Constant
  - Document Reports 2664
  - Insert In Document Report Template 1054
- Project Custom Colors
  - Get For Elements And Connectors 621
  - Set For Elements And Connectors 621
- Project Factor Calibration 584
- Project File
  - Create 199
  - Open 202
- Project Gantt View 538
- Project Glossary
  - Add Item, Glossary Detail Dialog 534
  - Add Item, Glossary Dialog 535
  - Create Item From Notes Text 1142
  - Delete Item, Glossary Dialog 535
  - Delete Item, Project Glossary Tab 534
  - Filter List, Glossary Dialog 535
  - Filter List, Project Glossary Tab 534
  - Glossary Dialog 535
  - Glossary Report 536
  - Hyperlink Term From Notes 1142
  - Insert Item In Text 1142
  - Modify Item, Glossary Detail Dialog 534
  - Modify Item, Glossary Dialog 535
  - Redefine Entry Type, Project Glossary Tab 534
  - Tab 534
- Project Indicators
  - Project Types Menu Option 120
  - Risk Types 524
- Project Interface
  - Automation Interface 2962
  - Project 2962
- Project Issue
  - Add 529
  - Delete 529
  - Dialog 528
  - Modify 529
  - Print 528
  - Record 528
  - Report, Via Issues Tab 531
  - Report, Via Project Issues Dialog 531
  - Tab 528
- Project Management
  - Asterisk On Window Tabs 512
  - Default Hours 588
  - Effort Management 518
  - Effort Types 522
  - Environment Complexity Factors 586
  - Gantt View 594
  - Introduction 509
  - Kanban Lanes 857
  - Maintenance 2621
  - Metric Types 523
  - Metrics 510, 520
  - Model Template 755
  - Resource Allocation 512
  - Resource Report 546
  - Resources 510
  - Risk Management 519
  - Risk Types 524
  - Risks 510
  - Technical Complexity Factors 585
  - Toolbar 510
  - Window 510, 512
  - With Enterprise Architect 509
- Project Manager
  - And Enterprise Architect 190
  - Project Estimation 190
  - Project Role 190
  - Resource Management 190
  - Risk Management 190
- Project Role
  - And Enterprise Architect 182
  - Business Analyst 185
  - Database Administrator 194
  - Deployment and Rollout 192
  - Developer 188
  - Implementation Manager 192
  - Project Manager 190
  - Software Architect 186
  - Software Engineer 187
  - Technology Developer 193
  - Tester 191
- Project Settings
  - Configure (Settings Menu) 119
- Project Status View 525
- Project Status Window
  - Monitor Tasks 559
  - Project Tasks Tab 559
- Project Task
  - Add 527
  - Delete 527
  - Description 526
  - List 526
  - Modify 527
  - Tab, Print List 526
- Project Task Allocation
  - Element View 542
  - Resource View 538
  - Resource Report 546

- Project Task Allocation
  - Window 538
- Project Tasks
  - Create, Project Calendar 581
  - Create, Personal Tasks Window 559
  - Delete, Project Calendar 581
  - Delete, Personal Tasks Window 559
  - Edit, Project Calendar 581
  - Edit, Personal Tasks Window 559
  - Monitor, Project Calendar 581
  - Monitor, Personal Tasks Window 559
  - Tab, Personal Tasks Window 559
  - View, Project Calendar 581
- Project Team Review
  - Add External File Link To Post 359
  - Add New Category 350
  - Add Object Link To Post 359
  - Add Team Review Link To Post 359
  - Comment On Document 355
  - Connections To Other Team Reviews 363
  - Copy Path To Clipboard 346
  - Create Category 350
  - Create Comment 355
  - Edit Item 358
  - Editor 358
  - Load Data When Required 362
  - Loading Behavior 362
  - Mark All Posts Unread 362
  - Options 362
  - Preload 362
  - Search 362
- Project Types
  - Estimation Factors 120
  - General Types 120
  - Maintenance Data 120
  - People 120
  - Project Indicators 120
  - Submenu 120
- ProjectIssues
  - Automation Interface Repository 2844
- ProjectResource
  - Automation Interface Repository 2846
- ProjectRole
  - Automation Interface Repository 2847
- Properties
  - As Association Roles 1189
  - As Parts 1189
  - Automation Interface, ElementFeatures Package 2929
  - Behavior Tab 1026
  - Call Tab 1026
  - Connector, Advanced 1133
  - Connector, Menu Section 1103
  - Constraints, Scenario 988
  - Dialog, Element 956
  - Editor Language 2251
  - Effect Tab 1266
  - Element 956
  - Element Context Menu 939
  - Element, Advanced 961
  - Element, Associated Files 988
  - Element, Connectors 964
  - Element, Constraints 963
  - Element, Details 959
  - Element, External Requirements 963
  - Element, General 958
  - Element, Internal Requirements 961
  - Element, Links 964
  - Element, Relationships 964
  - Element, Requirements 961
  - Element, Trigger Tab 1350
  - Extend For Requirements 1777
  - Object 956
  - Of Classifiers, Composite Structure Diagram 1189
  - Of Requirements, Extended 1777
  - Of Requirements, Standard 1774
  - Part, Property Tab 1386
  - Port, Property Tab 1386
  - Requirement, Display On Diagram 1777, 1778
  - Window (Element) 992
- Properties Note
  - Diagram 848
- PROPERTIES Parameter
  - OSLC Query Capability 278
- Properties, EASL
  - Action 1693
  - Argument 1693
  - Behavior 1693
  - Call Event 1693
  - ChangeEvent 1693
  - Classifier 1693
  - Condition 1693
  - Construct 1693
  - Edge 1693
  - EventObject 1693
  - Instance 1693
  - Parameter 1693
  - Primitive 1693
  - PropertyObject 1693
  - SignalEvent 1693
  - State 1693
  - StateMachine 1693
  - TimeEvent 1693

## Properties, EASL

Transition 1693

Trigger 1693

Vertex 1693

## Property

Automation Interface, ElementFeatures Package  
2929

Bind To Association (SysML Block) 2304

Create In BPEL 2.0 1910

Delete When Association Deleted 2304

Element, Drag Onto New Element 989

Generate From Association (SysML Block)  
2304

Model In BPEL 2.0 1910

Redefined 996

Select For Instance 1011

Select, Dialog 996

Subsetted 996

## Property Validation

Attribute 2600

Element 2600

Feature 2600

Relationship 2600

## Property Value

Part, Add To 1383

## PropertyType

Automation Interface Repository 2848

## PropType Enum

Automation Interface 2823

## Protect Text InDocument Editor

Lock Against Protection 1093

Text Block 1093

Whole Document 1093

## Provided Interface 1375

## Proxy

(Shortcut) File 204

## Pseudo-State

Elements 1208

Fork 1307, 1309

In State Machine Diagram 1208

Join 1307, 1310

## PSM 2013

## Publish

Packages, Define Scripts In MDG Technology  
1578

## Publish As HTML Dialog 2744

## Publish Model Package

For ArcGIS XML Workspace 1961

For XMI 476

## Purchase

Enterprise Architect 33

## Python

Code Generation 2273

Import, Reverse Engineering 2139

Language Options 2273

MDG Technology For, Enterprise Architect  
1481

Modeling Conventions 2100

## - Q -

## QA Reports

Dependency Details 2739

Implementation Details 2737

Maintenance 2740

Testing Details 2736

Use Case Metrics 588

## Qualified Association 1395

## Qualifier

Association End 1396

Association Property 1395

Attribute 1396

Attribute Property 1395

Dialog 1396

Of Attribute 1003

Of Port 1386

Part 1396

Part Property 1395

Port 1396

Port Property 1395

Set Properties 1396

## Quality Control

Model Validation 2593, 2594

Spell Checking 549

Testing 2593, 2604

## Query Builder

Search Definition 711

## Query Capability

OSLC 273

OSLC, Combine WHERE &amp; SELECT 277

OSLC, PREFIX Parameter 279

OSLC, PROPERTIES Parameter 278

OSLC, SELECT Parameter 276

OSLC, WHERE Parameter 275

Parameters 273

Response 273

## Query Methods

In Shape Scripts 1599

## Quick Add

Tagged Values 1136

## Quick Linker 1109

Add Definition To Profile 1523

Arrow 896

Create Connector 898

Quick Linker 1109

- Create Element And Connector 897
- CSV File Format 1519
- Default Definition, Hide 1521
- Definition, Columns 1519
- Definition, Connector Names 1521
- Definition, Element Names 1521
- Definition, Example 1519
- Definition, Fields 1516
- Definition, Format 1516
- Definition, Object Names 1521
- Definitions, Introduction 1515
- Introduction 896
- Options 897

Quick Start

- Add Connectors To UML Model 55
- Add Diagram To Package 50
- Add Diagram To UML Model 50
- Add Element To Diagram 52
- Add Element To UML Model 52
- Add Package to UML Model 50
- Auditing 447
- Common Areas Of Work 45
- Create A Project In Enterprise Architect 47
- Define Connector Properties 56
- Define Element Properties 56
- Delete UML Model Components 64
- Generate Report 2642
- Generate Web Report 2744
- Move Project Components 57
- Project Tasks 183
- Save Project Changes 51
- View Web Report 2744

## - R -

### RAS

- Asset Dependencies, Review 288
- Asset Diagrams, Review 286
- Asset Elements, Review 286
- Asset Properties (Package), Review 286
- Available MDG Technologies 289
- Browse Assets In Registry 284
- Check Dependencies 302
- Compare Asset To Model 291
- Connect To Asset Registry 283
- Delete Assets In 304
- Delete Storage Files In 304
- Import Asset Package to Model 291
- Import Storage Files to Local System 290
- Import Technologies to Local System 289

- Import Technology To Model With Asset Package 291
- Introduction 282
- Maintain Storages 293
- MDG Technologies, Review 289
- Overview 282
- Package Versions, Check For 284
- Register Asset Packages 298
- Register Files 298
- Register Packages Needed By Asset 302
- Register Technologies 298
- Registry Server, Identify For Setup 293
- Review Asset Contents 286
- Review Asset Properties 286
- Review Storage Files 290
- Set Up 293
- Storage Password Protection 293
- Unavailable MDG Technologies 289
- Update Assets In 304

Rational Rose

- And XMI 473
- Export To 482

Rational Software Architect

- Models, Import 480

Rational Software Modeler

- Import \*.emx Files 478
- Import \*.uml2 Files 478

RDF

- Definition Diagram (ODM) 1979
- Elements 1972
- Elements, Descriptions 1977
- Example Diagram 1979
- Export To File 1980
- Import From File 1980
- Profile 1971
- Relationships 1972
- Relationships, Descriptions 1977
- Toolbox Pages 1972

Read-Only Options

- For Enterprise Architect 23

ReadSelf

- Action 1268

ReadVariable Action

- Simulation Variable Action 1273

Realization

- Connector, Quick Generation Of 1779

Realization Link

- Implement Parent Operations 1023
- Override Parent Operations 1023

Realize

- An Information Flow 1414
- Connector 1440

- Realize
  - Relationship 1440
- Realized Interfaces
  - For Class, Show On Diagram 865
- Receive
  - Element 1327
  - Event 2001
- Recent Discussions
  - Concerning Model View Items 686
- Recent Post Options Dialog 690
- Recent Postings
  - Display 693
- Reception
  - Definition 1387
  - Of Signal 1387
- Record
  - Application, Analyzer Menu Option 112
  - Submenu (Analyzer) 115
  - Toolbar Options 148
- Record & Analyze
  - Toolbar Options 148
  - Window Context Menu 2546
  - Window, Toolbar 2544
- Record & Analyze Window 2533
- Record Activity For Class
  - Execution Analysis, Record Sequence Diagram 2542
- Record Activity For Method
  - Execution Analysis, Record Sequence Diagram 2538
- Record Macro
  - In Source Code Editor 2149
- Record Marker
  - Set 2537
- Record Sequence Diagrams
  - Control Recording 2544
- Record State Changes
  - Visual Execution Analyzer 2553
- Record Work Progress
  - Personal Tasks View 555
- Recorder
  - Toolbar 2544
- Recording Actions
  - Create Sequence Diagram, Call Stack 2241
  - Debugger, Step Through Function Calls 2548
- Recording History
  - Context Menu Options 2546
  - Save, Execution Analysis 2549
  - Work On 2546
- Recording Markers
  - Activate, Execution Analysis 2224
  - Breakpoints And Markers Window, Execution Analysis 2224, 2540
  - Difference From Breakpoint 2538
  - Disable, Execution Analysis 2224
  - Marker Types, Execution Analysis 2538
  - Nested 2548
  - Place, Execution Analysis 2536
  - Work With Marker Sets, Execution Analysis 2541
- Recover
  - Controlled Package 494
  - Diagram After Changes (Baseline) 466
  - Project 597
- Rectangle Notation
  - Element Menu Option 944
  - For Shaped Elements 1355
- Recursion
  - Connector 1441
  - Message 1441
  - Relationship 1441
- Recursive Builds
  - Visual Execution Analyzer 2180
- Red
  - Border 340
  - Exclamation Mark 340
  - Letter A 1044, 1047
  - Letter T 2709
  - Object Outline 340, 872, 939
  - Triangle 591
- Redefined
  - Property, Port 1386
- Redefined Port
  - Manage 1385
- Redefined Property
  - Of Attribute 1003
- Redo
  - Last Action, Diagram Edits 873
  - Option (Edit Menu) 80
- Re-entrancy
  - In Add-Ins 3016
- Reference
  - Automation Interface 2812
  - Automation Interface Repository 2848
- Reference Data 1146
  - And Version Control 387
  - Cardinality (Multiplicity) 1151
  - Clients 1157
  - Constraint Status Types 1161
  - Constraint Types 1160
  - Difficulty Types (Requirements) 1163
  - Estimation 1169
  - Export 376



- Reference Data 1146
  - Export, Introduction 374
  - General Types 1158
  - Import Automatically 380
  - Import Manually 380
  - Import, Introduction 374
  - Maintenance 1169
  - Metrics 1169
  - People 1152
  - Priority Types (Requirements) 1164
  - Problem Types 1169
  - Project Author 1153
  - Requirement Types 1166
  - Resources 1156
  - Roles 1155
  - Scenario Types 1167
  - Share 374
  - Shared Repository 375
  - Status Types 1159
  - Stereotypes 1147
  - Tagged Value Types 1150
  - Test Status Types (Requirements) 1165
  - Testing Types 1170
  - UML Types 1146
- Reference Data Tagged Value Type 1629
- Reference Data Tagged Values
  - Create 1628
- Referenced XML Schema
  - Import 2420
- Refresh
  - Chart 2763
  - Diagram 308
  - Image 860
  - Project 308
  - View Of Shared Model 308
- Region
  - Composite State 1209
  - Concurrent Substate 1209
  - Element 1328
  - Expansion, Element 1301
  - Interruptible Activity, Element 1316
  - On Composite State 1330
  - State Machine 1209
- Register
  - Add-Ins 3171
  - Enterprise Architect 36
- Registration Key
  - In License Information 3164
- Registry
  - Add Files To 298
  - Add Packages To 298
  - Add Storage To 293
  - Add Technologies To 298
  - Check Dependencies 302
  - Delete Assets In 304
  - Delete Storage Files In 304
  - Identify In Reusable Asset Service 293
  - Register Packages Needed By Asset 302
  - Update Assets In 304
- Registry Settings
  - Customized Options 151
  - User 604
- Regular Expressions
  - Code Editor Find And Replace 2161
  - Metasequences 2161
  - Tagged Regions 2161
- Related Elements
  - Find 838
  - Insert In Diagram 933
  - Place On Current Diagram 838
- RelatedElements
  - Shape Script 1604
- Relational Database
  - Model Template 755, 762
- Relationship 1939
  - Abstraction 1391
  - Activity Edge 1415
  - Aggregate 1392
  - ArcGIS 1946
  - ArchiMate 1926
  - Assembly 1393
  - Associate 1393
  - Association 1393
  - Association Class 1398
  - Asynchronous Signal 1427
  - Communication 1402
  - Communication Path 1400
  - Compose 1401
  - Composite Aggregation 1401
  - Connector 1402
  - Control Flow 1403
  - Create Using Relationship Matrix 737
  - Create With Overlay 737
  - Data Flow 1797
  - Delegate 1404
  - Delete 964
  - Delete Using Relationship Matrix 737
  - Dependency 1404
  - Dependency, Apply Stereotype 1405
  - Deployment 1406
  - Display Options 635
  - Element (Entity Relationship Diagram) 1939
  - Entity Relationship Diagram 1939
  - Eriksson-Penker 1929

- Relationship 1939
  - Extend 1406
  - Extension (Profile Toolbox) 810
  - Generalization 1409
  - Generalize 1409
  - Generalize (Profile Toolbox) 810
  - GML 1984
  - Hide 964
  - Implements 1440
  - Include 1410
  - Information Flow 1410, 1411
  - Inheritance 1409
  - Interrupt Flow 1415
  - List, On Context References Tab 987
  - Manifest 1416
  - Matrix 727
  - Matrix, In Traceability 724
  - Message 1417
  - Mind Mapping 1794
  - Modify Using Relationship Matrix 737
  - Nesting 1434
  - Notelink 1435
  - Object Flow 1435
  - Occurrence 1438
  - Package Import 1438
  - Package Merge 1439
  - Pkg Import 1438
  - Pkg Merge 1439
  - Realize 1440
  - Recursion 1441
  - Representation 1443
  - Represents 1442
  - Role Binding 1441
  - Self Message 1421
  - Show 964
  - SoaML 2451
  - SPEM 1920
  - Substitute 1443
  - Substitution 1443
  - SysML Activity 2295
  - SysML Block Definition 2300
  - SysML Interaction 2306
  - SysML Internal Block 2308
  - SysML Model 2310
  - SysML Parametric 2312
  - SysML Requirement 2314
  - SysML State Machine 2316
  - SysML Use Case 2318
  - Tagged Value (Profile Toolbox) 810
  - Trace 1446
  - Transition 1446
  - Usage 1450
  - Use (Class Diagram Usage) 1450
  - Use (Use Case Connector) 1450
  - Visibility 1119
  - Window, Context Menu 742
- Relationship Matrix
  - Access From Shortcut 204, 207
  - Access From Shortcut (Direct Definition) 206
  - Connector Direction, Set 730
  - Connector Type, Set 730
  - Create Elements In 727
  - Create Profile In Specification Manager 1744
  - Create Relationship 737
  - Create Relationship With Overlay 737
  - Create Relationships In 727
  - Delete Relationship 737
  - Element Types, Set 729
  - Elements, Model Search For 731
  - Export to CSV 738
  - Facilities 727
  - In The Specification Manager 1744
  - Incorporate In Document Report 738
  - Incorporate In Web Report 738
  - Introduction 727
  - Link Direction, Set 730
  - Link Type, Set 730
  - Locate Elements 741
  - Manage Display Content 738
  - Model Searches In 731
  - Modify Relationship 737
  - Open In Specification Manager 1746
  - Operation 727
  - Options 738
  - Overlays, Apply 734
  - Overlays, Create 733
  - Overlays, Delete 733
  - Overlays, Edit 733, 734
  - Overlays, Remove 734
  - Packages, Drag From Project Browser 731
  - Print Matrix 738
  - Print Preview 738
  - Profile, Content In Document Report 2692
  - Profile, Delete 735
  - Profile, Save 735
  - Profile, Update 735
  - Review Elements 741
  - Save As .EMF File 738
  - Save As .PNG File 738
  - Scale Printout Width 738
  - Source Package, Set 731
  - Target Package, Set 731
  - Toggle To CRUD Matrix 732
- Relationship Rule

- Relationship Rule
  - Connectors 1956
  - Examples 1956
- Relationships Window
  - Context Menu 838
- Release History 68
- Reload
  - Diagram 308
  - Model (Shared) 308
  - Project 308
  - View 308
- Reload Current Project
  - Menu Option (File Menu) 79
- ReloadType Enumeration
  - Automation Interface 2823
- Remove
  - MDG Technologies, From Model 1477
  - Package Control 490
  - Recent Project 73
  - Replication 313
- REMOVE\_PREFIX
  - Macro 2067
- RemoveVariable Action
  - Simulation Variable Action 1273
- Rename
  - Diagram 839
  - Package 774
  - Project .EAP File 601
  - Views 771
- Re-order
  - Columns 2347
  - Messages 1430
- Repair
  - Project .EAP File 602
- Replace
  - Linked Document 1097
- Replica
  - Create 312
  - Synchronize 313
  - Upgrade 315, 600
- Replication
  - And Version Control 422
  - Change Collisions 310
  - Create Design Masters 311
  - Create Replicas 312
  - Disable 310
  - Impact On Auto Increment Columns 598
  - Introduction 310
  - Menu Options 111
  - Merge Rules 310
  - Remove 313
  - Resolve Conflicts 315
  - Synchronize Replicas 313
  - Upgrade Replicas 315
  - Using 310
- Report
  - As Web Page 2744
  - Dependency Details 2739
  - Diagram Only 2741
  - Document 2638
  - Documentation 2640
  - Documents, Other 2736
  - Function Line, Profiler 2564
  - Generation 2640
  - Implementation Details 2737
  - List Overrides 1081
  - Maintenance 2740
  - Numbered Sections 1081
  - Numbering Levels 1081
  - Numbering List 1081
  - Overview 2640
  - Project Issues, Via Issues Tab 531
  - Project Issues, Via Project Issues Dialog 531
  - Section Numbering 1081
  - Testing Details 2736
  - Testing, Generate 2742
  - Web 2638
- Report Constant
  - Insert In Document Report Template 1054
- Report Document Options
  - Document Generator (Enhanced) 825
  - RTF Generator (Legacy) 825
- Report Generator
  - Document Options (Enhanced) 825
  - Document Options (Legacy) 825
- Report Specification
  - Artifact 1358
- Reporting
  - Dependency 1787
  - Implementation 1787
  - Requirements 1787
  - Select Template 2648
- Repository
  - Access 2007, Set Up Project On 241
  - Adaptive Server Anywhere, Create 237
  - Attributes, Automation Interface 2850
  - Author Collection 2827
  - Automation Interface 2850
  - Client Collection 2828
  - Collection Class 2829
  - Datatype 2831
  - Encrypt Password 204, 209
  - EventProperties 2833
  - EventProperty 2833

## Repository

- Methods, Automation Interface 2850
- ModelWatcher 2834
- MySQL, Set Up Project On 220
- Open, Automation Interface Code Example 2995
- Oracle, Set Up Project On 250
- Package 2835
- Package, Automation Interface 2826
- Postgre SQL, Set Up Project On 227
- Progress OpenEdge, Create 245
- Progress OpenEdge, Set Up Project On 243
- ProjectIssues 2844
- ProjectResource 2846
- PropertyRole 2847
- PropertyType 2848
- Reference 2848
- Server-Based, Set Up Project On 214
- Server-Based, Access Permissions For 214
- Server-Based, Connect To 214
- Server-Based, Create 214
- SQL Server, Set Up Project On 215
- Stereotype 2873
- Sybase ASA, Set Up Project On 234
- Task 2874
- Term 2876
- Transfer Data Between 504
- Use Extras, Automation Interface Code Example 3004

## Representation

- Connector 1443
- Relationship 1443

## Represents

- Connector 1442
- Relationship 1442

## Require User Lock

- Apply Lock 339
- Release Lock 339

## Require User Lock Policy 319

## Required Interface 1375

## Requirement

- Aggregation 1779
- Analysis 755, 758
- And Level Numbering 1765
- And Use Cases 1765
- Auditing 1785
- Automation Interface, Element Package 2899
- Autonumbering 1773
- Baselines 1785
- Change Management 1785
- Changes 1785
- Color Code Status 1776

- Connect On Diagram 1779
- Connect Through Relationship Matrix 1779
- Connectors 1779
- Convert From Responsibility 1771
- Create From Text File Text 1773
- Create In Diagram 1773
- Create In OSLC Requirements Management 280
- Create In Project Browser 1773
- Dependency Report 1787
- Difficulty, Customize 1163
- Docked Windows 1782
- Drag Onto New Element 989
- Element 1763, 1765
- Element Template 1777
- Element, Hide/Show Connectors 1121
- Elements And Connectors 814
- Extend Properties, Default Format 1777
- External 963, 1765
- Fast Generate Realization Connector 1779
- Field Substitution Macros 1673
- Functional 961
- Gather 1726
- Hide Stereotype Letter 1763, 1773
- Hierarchies 1765
- Implementation Report 1787
- Import Via CSV 1780
- Inherited, Show 845
- Internal 961, 1770
- Internal, Import As Test 2614
- Internal, In Scenarios & Requirements Window 992
- Issues 1785
- Manage 1782
- Model 1726, 1765
- Model Template 755, 758
- Model Views 1785
- Modeling 755, 758
- Non-Functional 961
- Priority Type, Customize 1164
- Profile 1777
- Properties, Display On Diagram 1778
- Properties, Extended 1777
- Properties, Standard 1774
- Realization 1779
- Report Template 1787
- Reporting 1787
- Requirements Management 755, 758
- Review 1782
- Show Stereotype Letter 1763, 1773
- Stability 961
- Status, Color Coded 1776

## Requirement

- SysML Model 2325
- Tagged Values 1777
- Tagged Values, Display 1778
- Template 755, 758
- Toolbox Pages 814
- Trace Through Connectors 1779
- Trace Use Of 1784
- User-Defined Attributes 1777
- View 1783
- What Is A? 1726
- Windows For Tracing Use 1784
- Windows For Viewing 1783

## Requirement Type

- Add In Specification Manager 1758
- Define 1166

## Requirements Diagram

- Description 1767
- Elements And Connectors 1767
- Example 1768

## Requirements Management

- And Enterprise Architect 1726
- Enterprise Architect 9
- In Example Model 1726
- Open Services for Lifecycle Collaboration 268
- OSLC 268
- Overview 1726

## Requirements Model

- SysML 2325

## Requirements Modeling

- Enterprise Architect 9

## Reserved Names

- In Shape Scripts, Connectors 1611
- In Shape Scripts, Elements 1611

## Reset

- Auto Increment Column Values 598

## Reset Options

- For A Class 2279
- For All Classes 2279
- Options dialog 2279
- Source Code Language 669, 2279

## Reset Sort Order

- In Project Browser 57

## Resize

- Element 919
- Element By Increments 919
- Multiple Elements 919

## Resize Combo Box

- In MDG Win32® User Interface Technology 1996

## Resolve Change Conflicts

- Between Replicas 315

## Resource

- Add To Team Review Item 346
- Allocation 512
- And Tasking Details Dialog 546
- Assign Multiple To Task 515
- Automation Interface, Element Package 2900
- Delete From Team Review Item 346
- Field Substitution Macros 1674
- On Task Element 2010
- Report Gantt Chart 546

## Resource Allocation

- Assign Multiple Resources To Task 515
- In Project Calendar 578
- Project Task Allocation Window, Element View 542
- Project Task Allocation Window, Resource View 538
- Review Allocated Work, Personal Tasks View 555

## Resource Description Framework 1971

- Definition Diagram, Example 1979

## Resource Document

- Batch Generate Reports 2668
- Document Generator (Enhanced) 2668

## Resource Management 512

- Effort Types 522
- Metric Types 523
- Risk Types 524

## Resource Script

- Export Win32 Dialog 2141
- Import Win32 Dialogs 2141

## Resource Shape

- OSLC 271
- Property Constraints 271
- Requirement Properties 271

## Resource View

- Gantt Chart 538
- Project Task Allocation 538

## Resources

- Define 1156
- Favorites Folder 1176
- Project Management Window 510
- Window, And MDG Technologies 1476
- Window, And Profiles 1526
- Window, Import UML Profile 1526

## Resources Portal 68

## Resources View

- Of Patterns 1464

## Resources Window

- Document Cover Pages 1173
- Document Generation 1173
- Document Report Style Sheets 1173

## Resources Window

- Document Report Fragments 1173
- Favorites Folder 1173
- HTML Templates 1173
- Legacy RTF Templates 1173
- Linked Document Templates 1173
- MDG Technologies 1173
- Relationship Matrix Profiles 1173
- Resource Documents 1173
- System Report Templates 1173
- UML Patterns 1173
- UML Profiles 1173
- User Report Templates 1173
- Web Templates 1173
- Window 1173
- XSL Stylesheets 1173

## Responsibility

- Compartment, Element 955
- Define 1770
- Import As Test 2614
- Internal 961
- Move To External Requirement 1771

## Result

- End Event, BPEL 1.1 1885
- End Event, BPEL 2.0 1907

## Resynchronize

- Package Version Control Status 444

## Reusable Asset Service

- Asset Dependencies, Review 288
- Asset Diagrams, Review 286
- Asset Elements, Review 286
- Asset Properties (Package), Review 286
- Available MDG Technologies 289
- Browse Assets In Registry 284
- Check Dependencies 302
- Compare Asset To Model 291
- Connect To Asset Registry 283
- Delete Assets In 304
- Delete Storage Files In 304
- Import Asset Package to Model 291
- Import Storage Files to Local System 290
- Import Technologies to Local System 289
- Import Technology To Model With Asset Package 291
- Introduction 282
- Maintain Storages 293
- MDG Technologies, Review 289
- Overview 282
- Package Versions, Check For 284
- Register Asset Packages 298
- Register Files 298
- Register Packages Needed By Asset 302

- Register Technologies 298
- Registry Server, Identify For Setup 293
- Review Asset Contents 286
- Review Asset Properties 286
- Review Storage Files 290
- Set Up 293
- Storage Password Protection 293
- Unavailable MDG Technologies 289
- Update Assets In 304
- Use Of Cloud Services 267

## Reusable Subsystems

- Systems Engineering Modeling 2330

## Reverse Connector 1123

## Reverse Engineer

- Enterprise Architect 9
- Source Code 2136
- Supported Languages 2136

## Reverse Engineering

- And Auditing 456
- And MDG Integration 2079
- Binary Module, Supported Languages 2143
- Directory Structure 2142
- Eclipse 2079
- Handling Classes Not Found During Import 2144
- Import ActionScript 2139
- Import Binary Module 2143
- Import C 2139
- Import C# 2139
- Import C++ 2139
- Import Delphi 2139
- Import Java 2139
- Import PHP 2139
- Import Python 2139
- Import Source Code 2138, 2144
- Import Visual Basic 2139
- Import Visual Basic.Net 2139
- Initial Code In Operations 1019
- Introduction 2073
- Languages Supported 2262
- ODBC Data Sources 2376
- Source Code, Import Directory Structure 2142
- Supported Software Languages 2139
- Synchronize Model And Code 2118
- Visual Studio 2079
- Writing Grammar 1705

## Reverse Synchronization

- Delete Attribute If Not In Code 2253
- Delete Method If Not In Code 2253
- Delete Model Aggregations For Attributes Not In Code 2253

- Reverse Synchronization
  - Delete Model Associations For Attributes Not In Code 2253
  - Include Method Bodies In Model 2253
  - Of Package 2117
- Revert
  - Changes In Audit Log 451
- Review
  - Package Version Control History (SCC Example) 442
  - Package Version Control History (SCC) 441
  - Package Version Control History (TFS, CVS, Subversion) 440
  - Requirements 1782
- Review Document
  - Auto-Generate In Specification Manager 1749
  - Bind Package To Team Review Folder (Topic) 1749
  - Generate In Specification Manager 1749
  - Reassign Default Folder In Specification Manager 1749
  - Specification Manager, Create In 1749
  - View, Specification Manager 1751
- Rhapsody
  - Import Model From 482
- Rich Text Format Report
  - Apply Filter (Legacy) 2722
  - Diagram Format (Legacy) 2723
  - Dialog (Legacy) 2720
  - Exclude Elements (Legacy) 2722
  - Exclude Objects (Legacy) 2722
  - Generate (Legacy) 2726
  - Generator (Legacy) 2720
  - Include Glossary (Legacy) 2723
  - Include Issues (Legacy) 2723
  - Include Tasks (Legacy) 2723
  - Object Selections (Legacy) 2724
  - Options (Legacy) 2723
  - Save As RTF Document (Legacy) 2728
  - Set Main Properties (Legacy) 2721
  - Single Element (Legacy) 2721
  - Templates, Load (Legacy) 2726
  - Wizard (Legacy) 2720
- Rich Text Notes
  - In Legacy RTF Generator Reports 2720
- Risk
  - Automation Interface, Element Package 2902
  - Element 2009
  - Field Substitution Macros 1675
  - Management 519, 2009
  - Project Management Window 510
- Risk Types
  - Define 524
  - Global 524
  - Non-Global 519
- Roadmap
  - ICONIX 2282
- Robustness Diagram 1997
  - Generate From Scenario 976, 984
- Role
  - Context Menu 1102
  - Define 1155
  - Tagged Values 1132
- Role Binding
  - Connector 1441
  - Relationship 1441
- RoleTag
  - Automation Interface, Connector Package 2944
- Roll Back
  - Changes In Audit Log 451
- Rollback Change
  - Baseline Comparison 470
- Round Corners
  - Set For Rectangular Elements 622
- Round-Trip Engineering
  - Introduction 2073
- RSA
  - Models, Import 480
  - XMI 478
- RSM
  - Import \*.emx Files 478
  - Import \*.uml2 Files 478
- RTF Report
  - Apply Filter (Legacy) 2722
  - Custom Language Settings (Legacy) 2725
  - Diagram Format (Legacy) 2723
  - Dialog (Legacy) 2720
  - Exclude Elements (Legacy) 2722
  - Exclude Objects (Legacy) 2722
  - Generate (Legacy) 2720, 2726
  - Include Glossary (Legacy) 2723
  - Include Issues (Legacy) 2723
  - Include Tasks (Legacy) 2723
  - Keywords (Legacy) 2725
  - Object Selections (Legacy) 2724
  - Options (Legacy) 2723
  - Save As RTF Document (Legacy) 2728
  - Set Main Properties (Legacy) 2721
  - Single Element (Legacy) 2721
  - Templates, Load (Legacy) 2726
  - Wizard (Legacy) 2720
- RTF Style Editor (Legacy) 2726
- Rule Action
  - Add To Business Rule Composer 1837

## Rule Composer

- Add Rule To Task 1834
- Business Rule Modeling 1833
- Computation Rule Table 1833, 1840
- Decision Table 1833
- Decision Table, Binding 1839
- Decision Table, Define Conditions 1836
- Decision Table, Rule Actions 1837
- Define Computation Rule 1840
- Define Conditions 1836
- Remove Rule From Task 1834
- Rule Table 1833

## Rule Condition

- Define 1836

## Rule Flow

- Activity 1826
- Activity Parameter 1829
- Behavior 1826
- Diagram 1826
- Generate Code From Behavior 1843
- Model 1826

## Rule Model

- Diagram 1821

## Rule Statements Table

- Add Rule 1834
- Business Rule Modeling 1834
- Remove Rule 1834

## Rule Task

- Element 1826

## RuleFlow Diagram

- Generate From Scenario 976, 980

## Run

- Application, Analyzer Menu Option 112
- SQL Patches 601

## Run Command

- Create 2217

## Run Script

- Create 2217
- Execute, Build Toolbar Option 147

## Run State

- Add Instance Variable 1380
- Define, Element Context Menu 944

## Run-Time

- Variable, Define 1380

## Runtime Object

- In Script Editor 2798
- Session, Script Editor 2800

## Run-Time State

- Add Instance Variable 1380
- Delete Instance Variable 1381
- Introduction 1380

**- S -**

## Sample Model

- Java 2529
- Microsoft C++ 2529
- Microsoft.NET 2529
- Visual Execution Analyzer 2529

## Save 491

- Changes 51
- Controlled Package 491
- Diagram As UML Pattern 1464
- Diagram Changes Automatically 625
- Diagram Image To Disk File 841
- Diagram, Context Menu Option 778
- Diagram, Default Tools Toolbar 135
- Diagram, Quick Start 51
- Package with XMI 491
- Profile From Diagram Context 1525
- Profile From Package Context 1525
- Report As Defined Document (Enhanced) 2668
- RTF Report As RTF Document (Legacy) 2728
- UML Pattern 1464

## Save As

- Shortcut 204

## Save Project As

- Copy, (File Menu Option) 206
- Copy, File Menu Option 207
- Menu Option (File Menu) 79
- Shortcut (Capture Current Environment) 207
- Shortcut (Direct Definition) 206

## Scale

- Image To Page Size 871

## Scale Diagram View 622

## Scan XMI and Reconcile

- Version Control 427

## SCC

- Create Environment For Version Control 411
- Package Version Control History, Example 442
- Package Version Control History, In 441
- Providers Dialog 417
- Version Control Options 417
- Version Control, Upgrade For Enterprise Architect 4.5 412

## Scenario

- Alternate Path 965, 967
- Alternate Path, Add 970, 972
- Automation Interface, Element Package 2903
- Basic Path 965
- Context Menu, Item 972
- Create 968, 970, 972
- Create Element 974



## Scenario

- Delete Path 976
- Delete Step 970, 972
- Delete Text 974
- Description Tab 965
- Edit Text 974
- Element 965
- Entry Points Tab 967
- Entry Points Tab, Context Menu 976
- Exception Path 965
- Exception Path, 967
- Exception Path, Add 970, 972
- External Test Cases 986
- Field Substitution Macros 1675
- Floating Toolbar 976
- Generate Activity Diagram From 978
- Generate Diagram From 976
- Generate From Activity Diagram 985
- Generate From Clipboard Text 972
- Generate Robustness Diagram From 984
- Generate RuleFlow Diagram From 980
- Generate Sequence Diagram From 982
- Generate State Machine Diagram From 980
- Generate Structure From Notes 965
- Glossary Reference, Add 974
- In Scenarios & Requirements Window 992
- Include In Document Report 2693
- Insert Context Reference 974
- Internal Test Cases 986
- Item Context Menu 972
- Join Scenarios 976
- Link Step To Use Case 972, 974
- Link To Element 974
- Merge Steps 972
- Move Step 970
- Move Steps 972
- Object 965
- Organization 976
- Set Responsible Entity 972
- Split Step 974
- Step Numbers On Sequence Messages 982
- Structured Specification Tab 965, 967
- Structured Specification, Create 968
- Test Cases 986
- Testing 2607
- Text Context Menu 974
- Toolbar 970
- Type, Define 1167
- Undo Changes 974
- Use Case 1253

## ScenarioDiagramType Enum

- Automation Interface 2824

## ScenarioExtension

- Automation Interface, Element Package 2904

## Scenarios &amp; Requirements

- Toggle Window/View 992
- Window 992

## ScenarioStep

- Automation Interface, Element Package 2905

## ScenarioStepType Enum

- Automation Interface 2825

## ScenarioTestType Enum

- Automation Interface 2825

## Scheduled Tasks

- Use Of Cloud Services 267

## Schema

- ArcGIS, Export 1961
- ArcGIS, Import 1968
- Database 1943
- Database, Import From ODBC 2376
- DDL, Import From ODBC 2376
- Modular, ArcGIS, Export 1962
- Owner Tagged Value 2342
- Partial, ArcGIS, Model 1962
- Set Owner 2342

## Scope

- Values, Programming Languages 2080

## Scope Guides

- Code Editor, Common 2157

## Screen

- Element 1991
- Element, Default Fill Color 616

## Script

- Add To MDG Technology 1556
- ArcGIS Workspace Validation 1970
- Commands 2796
- Console 2796
- Copy 2792
- Create 2792
- Debugging 2801
- Delete 2792
- Deploy, Create 2218
- Diagram 2794
- Engines 2791
- Exclude Package From Report Filters 2658
- Execute 2792
- Generate Learning Center Documentation 74
- Group 2791, 2792
- Group Type 2794
- Hyperlink To 2004
- JavaScript 2791, 2792
- JScript 2791, 2792
- Local 2791, 2792
- Model Search 703, 2794

## Script

- Move 2792
- Normal 2794
- Project Browser 2794
- Search 700, 2164, 2794
- Template 2791, 2792
- Template Fragments 2714
- Template Fragments, Example 2716
- Template Fragments, Output 2717
- Unit Test, Create 2182
- Unit Test, Introduction 2182
- User 2791, 2792
- VBScript 2791, 2792
- Workflow 2792, 2794
- Workflow Functions 368
- Workflow, Introduction 367

## Script Actions

- Define, Visual Execution Analyzer 2179

## Script Editor

- Context Menu 2152

## Scripting

- In Enterprise Architect 2790

## Scripting Objects 2798

- Session 2800

## Scripting Window

- Console Tab 2796
- Console Toolbar 2796
- Context Menu 2792
- Create Script 2791
- Editor 2798
- Group Properties Dialog 2794
- Intelli-sense 2798
- Math Functions 2798
- Runtime Objects 2798
- Script Editor 2798
- Scripting Objects 2798
- Scripts Tab 2792
- Toolbar 2792
- Type Libraries 2798

## Scripts

- Analyzer Menu Option 112
- Build 2180
- Write Shape Scripts 1585

## SDK

- Enterprise Architect 1483

## Search

- Add To MDG Technology 1559
- Add-In 3019
- Advanced Options 715
- Code Editor Facility 2152
- Conditions 718

## Configure Code Editor Context Menu Options 2152

- Debugger File Search 2164
- Definitions 709
- Definitions, New 711
- Definitions, Predefined 706
- Element Features 709
- Element Features, Document Reports 2663
- Element Filters, Document Reports 2660
- Fields 718
- Filter On Feature Values 715
- Filters 709
- Intelli-sense Libraries 2166
- List, Predefined Searches 706
- Macros, SQL 711
- Manage 709
- Model 709
- Model, Create Search Definition 711
- Other Filters, Document Reports 2663
- Project 709
- Query, New 711
- Results, Manipulate 700
- Scripts 2164, 2794
- Simple 709
- Specific Package 715
- Team Review 362
- User Defined, Storage 711
- Whole Model 715
- Within Project Browser 669

## Search Data Parameter

- Add-In Search 3020

## Search Filters

- Add To Search 716

## Search Project

- Add Filters 716

## Search Tools 644

## Section Numbering

- In Document Reports 1081

## Security

- Basics 316
- Change Password 323, 334
- Disable 318
- Enable 318
- Lock Elements & Diagrams, User/Group Locking 336
- Lock Packages, User/Group Locking 337
- Locked Element Indicators 340
- Maintain Groups 320
- Maintain Users 323
- Policy 319
- Re-enable 318

## Security

- Release Elements & Diagrams, User/Group Locking 336
- Release Package, User/Group Locking 337
- Require User Lock Mode 319
- Reset Password 334
- Rigorous Security 319
- Set Password 334
- Standard Security 319
- Submenu (Project Menu) 92
- Tasks 316
- User Permission List 329
- User/Group Lock Mode 319
- What Is User Security? 316
- Security Group Permissions 321
- See the Pass Parameters to Rule Flow Activity 1826
- Select
  - Alternative Image 860
  - ODBC Data Source 2378
  - Stereotypes 1455
- Select <Item> Dialog 994
- Select All
  - Menu Option (Edit Menu) 80
- Select Attribute Type
  - Data Type 1001
  - Dialog 1001
- Select By Type
  - Menu Option (Edit Menu) 80
- SELECT Parameter
  - OSLC Query Capability 276
  - OSLC, Combine With WHERE 277
- Select Property Dialog 996
- SELECT Statement
  - Enable Drag And Drop 711
  - SQL Search 711
- Selectable
  - Element 939
- Self Message
  - Calls 1422
- Self-Message
  - Connector 1421
  - Hierarchy, Sequence Diagram 1256
  - Relationship 1421
  - Return 1421
- Send
  - Element 1328
  - Event 2001
- SendSignal Action
  - Signal Tab 1268
- Sent Mail
  - Personal Information Window 565

## Sequence

- Communication Messages 1430
- Create Message 1418
- Elements and Connectors 806
- Message 1418
- Message, Change Timing Details 1424
- Message, Timing Details 1424
- Oracle, DDL Options 2380
- Oracle, DDL Options For Packages 2381
- Sequence Diagram
  - Activation Levels 1256
  - Add Space To Top 1258
  - And Version Control 1254
  - Damage To 1254
  - Description 1249
  - Diagram Features, Generate Sequence Diagrams 2535
  - Display Options 629
  - Element Activation 1255
  - Elements 1253
  - Elements And Connectors 1249
  - Example 1251
  - From Code Execution, Control Stack Depth 2536
  - Garbage Collect Option 629
  - Generate Code From 2121, 2133
  - Generate From Debugger Call Stack 2241
  - Generate From Recording, Execution Analysis 2549
  - Generate From Scenario 976, 982
  - Generate In Execution Analysis 2532
  - Generate, Execution Analysis 2549
  - Layout 1252
  - Lifeline Activation Level 1256
  - Messages, Asynchronous Signal 1427
  - Messages, Self Message 1421
  - Recording History, Execution Analysis 2533
  - Rendering Options, Default 629
  - Save Recording History, Execution Analysis 2549
  - Self-Message Hierarchy 1256
  - Step Numbers On Messages 982
  - Top Margin, Change 1258
  - Transformation To Communication Diagram 2041
- Sequence Diagram Message
  - Examples 1423
  - External To Sequence 1423
  - General Ordering 1426
- Sequence Diagrams
  - Filter calls 2219
- Sequence Element

- Sequence Element
  - Inline, Part And Port 1258
- Sequence Flow
  - Create In BPEL 1.1 1887
  - Create In BPEL 2.0 1910
  - Model In BPEL 1.1 1887
  - Model In BPEL 2.0 1910
  - Rules, BPMN 2.0 1845
- Sequence Message
  - Classifier Features 1009
  - Label Visibility 1258
  - Modify Height 1252
- Sequence Recording Option
  - Record Activity For Class 2542
  - Record Activity For Method 2538
- Sequential Node
  - Structured Activity 1338, 1341
- Server
  - Apache Tomcat, Debugging 2195
  - JBOSS, Debugging 2195
  - Tomcat, Debugging 2195
- Server Configuration
  - JBOSS 2198
  - Tomcat 2199
- Server Repository
  - Create for Oracle 10g 253
  - Create for Oracle 9i 253
- Service
  - WSDL Diagram 2441
  - WSDL Element 2441
- Service Configuration
  - Tomcat 2199
- Service Oriented Architecture
  - Development 2076
  - Implementing XML-Based, In Enterprise Architect 2076
  - Modeling Language (SoaML) 2386
- Service Oriented Architecture Modeling Language 2449
- Service Oriented Modeling Framework 2454
- Service Provider
  - OSLC 269
  - Resource, OSLC 269
- Service-Oriented Modeling Framework (SOMF) 2386
- Session
  - Object 2800
- Session Bean
  - Transformation 2029
- Set
  - Activities For Transitions 994
  - Association Specialization 1123
  - Classifiers For REFGUID Tagged Values 994
  - Collection Classes 2259
  - Connector Visibility 1119
  - Default Diagram, Model 844
  - Default Tree Behavior 672
  - Diagram Appearance Options 823
  - Diagram Page Size 870
  - Diagram Properties 823
  - Element Classifier 1467
  - Element Cross References 916
  - Element Custom References 916
  - Element Parent 908
  - Feature Visibility 845
  - Font For Element Text 949
  - Group Permissions 321
  - Instance Classifier 994
  - Interface For Element 908
  - Main RTF Report Properties (Legacy) 2721
  - Message Source and Target 1124
  - Object State 1381
  - Operation Parameter Return Type 994
  - Operation Parameter Type 994
  - Operation Return Type 994
  - Parent For Element 908
  - Pattern Element Defaults 994
  - Relationship Visibility 1119
- Set Attribute Dialog 998
- Set Feature Dialog 998
- Set Function
  - Create As Attribute Property 1006
- Set Operation Dialog 998
- Set Project Custom Colors
  - For Elements And Connectors 621
- Set Up
  - Adaptive Server Anywhere ODBC Driver 237
  - Cloud Server 262
  - Database Repository 214
  - Debug Session 2185
  - For .NET 2200
  - MySQL ODBC Driver 223
  - Oracle ODBC Driver 254
  - PostgreSQL ODBC Driver 230
  - Progress OpenEdge ODBC Driver 247
  - Project On DBMS Repository 214
  - Single Permissions 327
  - User Groups 326
- Settings
  - Author 1153
  - Cardinality (Multiplicity) 1151
  - Clients 1157
  - Constraint Status Types 1161
  - Constraint Types 1160

- Settings
  - Default Hours 588
  - Difficulty Types 1163
  - Effort Types 522
  - Environment Complexity Factors 586
  - Estimation 1169
  - General Types, Customize 1158
  - Maintenance 1169
  - Menu 1146
  - Menu, Configure 119
  - Metric Types 523
  - Metrics 1169
  - People 1152, 1156
  - Priority Types 1164
  - Problem Types 1169
  - Project Author 1153
  - Project Resources 1156
  - Requirement Types 1166
  - Risk Types 524
  - Roles 1155
  - Scenario Types 1167
  - Status Types 1159
  - Stereotypes 1147
  - Tagged Value Types 1150
  - Technical Complexity Factors 585
  - Template Package 929
  - Test Status Types 1165
  - Testing Types 1170
  - UML Types 1146
- Shadow
  - Set On Menu Borders 162
- Shallow Copy
  - Of Diagram 842
- Shallow History 1311
  - Change To Deep 944
- Shape
  - <LabelID> 1611
  - Attributes 1588
  - Decoration 1611
  - Editor 1584
  - Label 1611
  - Main 1611
  - Properties 1588
  - Source 1611
  - Target 1611
- Shape Attributes
  - Shape Scripts 1588
- Shape Editor 1149, 1584
- Shape Scripts
  - Add Composite Diagram To Element 1608
  - Add Custom Compartments 1604
  - Add To Profile 1501
- Addin Property, Use Of 1614
- Alter Embedded Element 1614
- Assign To Stereotype 1583
- Basic Shapes 1614
- Block Layout Examples 1585
- ChildElement 1604
- Cloud Path 1614
- Color Queries 1598
- Commenting - C Style 1585
- Conditional Branching 1598
- Conditional Statements 1599
- Connector 1614
- Create 1583
- Custom Shapes 1582
- Decoration 1585
- Display Element Properties 1599
- Double Line 1614
- Drawing Methods 1591
- Editable Field 1614
- Example Shape Scripts 1614
- Filled Arrow 1614
- Getting Started 1583
- Incoming Connector Point 1614
- Introduction 1582
- Multiple Condition 1614
- Order Of Declaration 1585
- Outgoing Connector Point 1614
- Override Element Appearance 1583
- Overview 1585
- Properties, Connector 1599
- Properties, Element 1599
- Query Methods 1599
- RelatedElements 1604
- Reserved Names, Connectors 1611
- Reserved Names, Elements 1611
- Return Command 1598
- Return Statement 1614
- Shape Attributes 1588
- Shape Editor 1149, 1584
- Single Condition 1614
- Stereotypes 1582
- Sub Shapes 1585
- Subshape 1614
- Subshape Layout 1603
- Syntax Grammar 1613
- Terminate Execution 1598
- Writing Scripts 1585
- Share
  - An Enterprise Architect Project 308
  - Project On Network Drive 309
  - Resource Data 375
- Shared

- Shared
  - Data 375
  - Database Information 375
  - Reference Data 375
  - Repository 375
- Shared Key
  - Add 3165
  - Add For Add-In 3172
  - Issues 3168
- Shared Model
  - Version Control Usage 385
- Shared Model Development
  - Enterprise Architect 9
- Shared Package
  - Version Control Usage 385
- Shortcut
  - Clear 207
  - Diagram (Capture Current Environment) 207
  - Keyboard 171
  - Keyboard/Mouse 180
  - Menu, Toolbox 799
  - Model (Capture Current Environment) 207
  - Model Search (Capture Current Environment) 207
  - Relationship Matrix (Capture Current Environment) 207
  - Team Review (Capture Current Environment) 207
- Shortcut To
  - Diagram 204
  - Diagram (Direct Definition) 206
  - Model 204
  - Model (Direct Definition) 206
  - Model Search 204
  - Model Search (Direct Definition) 206
  - Relationship Matrix 204
  - Relationship Matrix (Direct Definition) 206
  - Team Review 204
  - Team Review (Direct Definition) 206
- Show
  - Connectors, All Diagrams 1121
  - Connectors, Requirements Element 1121
  - Connectors, Single Diagram 1121
  - Diagram Caption Bar 84
  - Duplicate Tagged Values 1139
  - Element Stereotype 1456
  - Feature Stereotype 1456
  - Labels 1122
  - Package Contents On Diagram 777
  - Project Browser 646
  - Relationship 964
  - Toolbox 792
  - Usage Of Element 910
  - Use Case Arrowhead 1125
- Show Grid
  - Diagram Menu Option 93
- Show Name Under Image 860
- Show Status Colors on Diagrams
  - Menu Option 1776
- Show/Hide
  - Connectors 1107
  - Labels 1107
- Signal
  - Element 1387
  - Reception 1387
- Simple Type Element
  - XML, Toolbox Icon 2402
- Simple View 769
- Simulate
  - Conditional Node 2499
  - Loop Node 2499
  - Structured Activity 2499
  - SysML Parametric Model 2322
- Simulate the parametric models to verify their 2288
- Simulation
  - Accept Event Actions 1268
  - Activity Parameter 2501
  - Activity Return 2501
  - Activity Return In CallBehavior 2501
  - AddVariableValue Action 1273
  - BPMN 2521
  - BPMN Process, Compare With UML Activity 2524
  - BPMN, Control Via Tagged Values 2523
  - Classes In 2479
  - ClearVariable Action 1273
  - Composite Element In 2518
  - Create BPMN Model For 2521
  - Create Objects, By JavaScript 2480
  - Create Objects, Manually 2480
  - Create Objects, With CreateObject Action 2480
  - CreateObject Action 2480
  - Default Object Variables 2479
  - Destroy Global Object 2483
  - Destroy Objects, By JavaScript 2483
  - Destroy Objects, With DestroyObject Action 2483
  - DestroyObject Action 2483
  - Instances In 2479
  - Instantiate Inner Object 2480
  - Instantiate Object Member 2480
  - Interaction Operands 2490
  - Invocation Actions 1268
  - Message Behavior 2490

## Simulation

- Multi-Threaded, Concurrent State Region 2517
- Multi-Threaded, Forks And Joins 2516
- Object Actions 1268
- Objects In 2479
- ReadVariable Action 1273
- RemoveVariable Action 1273
- Simulate Activity Behavior 2489
- Simulate Class Behavior 2489
- State Machine Table 1224
- StructuralFeature 1268
- UML Activity, compare With BPMN Process 2524
- Windows 2469
- WriteVariable Action 1273

## Simulation Actions

- Accept Event 2497
- Invocation 2497
- Object 2497
- StructuralFeature 2497
- ValueSpecificationAction 2497
- Variable 2497

## Simulation Class

- Automation Interface, Simulation Package 2993

## Simulation Events Window

- Context Menu Options 2504
- Toolbar Option 2504
- Use In Simulation 2504

## Simulation Package

- Automation Interface 2993
- Simulation Class, Automation Interface 2993

## Simulation Script

- Activate 2474
- Evaluate Guards And Effects 2472
- Record Simulation 2472
- Run 2475
- Set Up 2472

## Single Permissions

- Set Up 327

## Single User 2256

## Size

- Elements By Increments 953
- Submenu 953

## Slideshow

- Diagram Presentations 686, 695
- Model View 686, 695
- Of Diagrams, Automate 695
- Properties Dialog, Model View 695
- Remove From Model View 695
- Run, Diagram View 695
- Run, Full Screen View 695
- Stop 695

## Slot

- Select On Instance 1011

## Snap To Grid

- Diagram Menu Option 93
- Display Grid 625
- Set As Global Option 625
- Set Grid Size 625

## Snapshot

- Of Variables In Code Execution 2236

## SOA

- Development 2076
- Implementing XML-Based, In Enterprise Architect 2076
- Modeling Language 2449

## SOA WSDL

- Elements 817

## SOA XSD

- Elements and Connectors 818

## SoaML

- Concept 2449
- Connectors 2451
- Diagrams 2449
- Disable 2449
- Elements 2451
- Enable 2449
- MDG Technology 2449
- Relationships 2451
- Toolbox Pages 2451

## SOAP Binding

- WSDL Model 2437

## Software and Systems Process Engineering Meta-model

- SPEM 1919

## Software Architect

- And Enterprise Architect 186
- Project Role 186

## Software Development

- Database Engineering 2334
- MDA Transformations 2013
- Model Transformations 2013
- SOA 2386
- Software Engineering 2073
- XML Engineering 2386

## Software Development Kit

- Enterprise Architect 1483

## Software Engineer

- And Enterprise Architect 187
- Project Role 187

## Software Product License Agreement 26

## SOMF

- Concept 2454
- Diagrams, 2.1 2454

## SOMF

- Disable 2454
- Elements 2454
- Enable 2454
- MDG Technology 2454
- Toolbox Pages, 2.1 2454

## Sort Order

- Project Browser, Reset 57

## Source

- Set for Message 1124

## Source Code

- Add New Features And Elements 1641
- Control 383
- Display In Source Code Viewer 2146
- Display Parameter Information 2149
- Editor Functions 2149
- Editors 2146
- Engineering, Project Browser Options 669
- File Parsing In Source Code Viewer 2146
- Generate For Attribute As Executable Statemachine 667
- Generate For Method In Project Browser 667
- Generate For Operation In Project Browser 667
- Import, Reverse Engineering 2138
- Internal Editor Options 2250
- Open Directory, Project Browser Submenu Option 664
- Record Macro 2149
- Reset Language 669, 2279
- Reverse Engineering, Supported Languages 2136
- Synchronize 2136
- Synchronize With Method In Project Browser 667
- Synchronize With Operation In Project Browser 667
- View For Method In Project Browser 667
- View For Operation In Project Browser 667
- View Options 2149
- View, Project Browser Submenu Option 664
- Viewer 2146
- Viewer Toolbar 2149
- XML Structure Tree 2146

## Source Code Editor

- Compare 2148
- External 2148
- Internal (External Code) 2148
- Variants 2148
- View 2148
- Window 2148

## Source Code Engineering

- Element Level 103

- Package Level 106

- Submenu (Element Menu) 103

- Submenu (Tools Menu) 106

## Source Code Generation

- Class 2111
- Interface 2111
- Options, Field Substitution Macros 1655
- Overview 2111
- Supported Software Languages 2111

## Source File

- Open, Option 104

## Source Object

- Multiplicity 1130

## Source Role

- Details 1130

## Space Evenly

- Multiple Elements 918
- Submenu 953

## Sparx Support

- Contact 394

## Sparx Systems

- Enterprise Architect Community Site 38, 68
- Website 38

## Spatial Reference

- ArcGIS 1958
- Class, Create 1958
- Element, Create 1958
- Package 1958
- Tagged Value 1958
- Vertical Coordinate System 1958

## Special Attribute

- Define Child Diagram Types 1512
- Define Composite Elements 1511
- Define Creation Of Instances 1510
- Define Tag Grouping 1514
- Metatype, In Profiles 1503
- Stereotype, In Profiles 1503

## Specialize Association 1123

## Specialized UML Models 1789

## Specification Manager

- Add Elements 1738
- Add Linked Document To Element 1740
- Add Tagged Values 1740
- Appearance, Change 1757
- Apply Autonaming/Aut numbering 1759
- Apply Level Numbering 1759
- Auditing 1752
- Auto-Generate Review Document 1749
- Autonaming System, Add 1758
- Aut numbering System, Add 1758
- Bind Package To Team Review Folder (Topic) 1749



- Specification Manager
  - Bold Text 1757
  - Collapse/Fix Element Hierarchies 1757
  - Configuration Options, Overview 1756
  - Configure 1731
  - Context Menu 1729
  - Create Elements 1738
  - Create New Specification Package 1735
  - Create Relationship Matrix Profile 1744
  - Customize Columns 1738
  - Delete Elements 1739
  - Develop Elements In 1731
  - Display Area 1729
  - Document Format Reports 1754
  - Edit Notes 1740
  - Elements, Review 1747
  - Empty Specification Package 1735
  - Export Data To CSV 1754
  - Facilities 1728
  - Field Chooser Dialog 1738
  - Filter Bar 1738
  - Font Size, Change 1757
  - Generate Report 2642
  - Generate Reports 1754
  - Glossary Entries, Add 1758
  - Hide/Display Notes 1757
  - Icons, Indicator 1743
  - Import Data From CSV 1754
  - Indent Element Descriptions 1759
  - Indicator Columns 1743
  - In-Place Editing 1729
  - Introduction 1728
  - Locate Package In Project Browser 1759
  - Maintenance 1752
  - Manage Changes 1752
  - Metrics Report 1754
  - Model Options 1758
  - Operations On Elements 1740
  - Overview 1729
  - Package Baselines 1752
  - Package Options 1759
  - Package View 1759
  - Print Screen Content 1754
  - Project Authors, Add 1758
  - Project Management 1752
  - Quality Assurance Reports 1754
  - Reassign Default Review Document Folder 1749
  - Relationship Matrix, Open 1746
  - Requirement Types, Add 1758
  - Review Document On Package, Create 1749
  - Review Document On Package, View 1751
  - Review Element Properties 1740
  - Review Elements 1747
  - Select Element Type 1734
  - Select Specification Package 1735
  - Select Specification Type 1734
  - Set Up 1731
  - Spell Check 1754
  - Tagged Value Types, Add 1758
  - Technology Element Types 1734
  - Testing 1752
  - Toolbar 1729
  - Tools 1728
  - Traceability 1744
  - Track Changes 1752
  - Uses 1728
  - Using 1731
  - Using The Relationship Matrix 1744
  - Using The Traceability Window 1744
  - View 1729
  - View As Gantt Chart 1752
  - Web Documents 1754
- Spectacle Icon
  - Inside Element 936
- Spell Check
  - Auto Spell 1089
  - Correct Words 554
  - Dictionary 554
  - In Document Editor 1089
  - Languages Other Than English 552
  - Model 553
  - Options, Configure 550
  - Perform 553
  - Project 553
  - Single Package 553
- Spell Checking
  - Automatic, Disable 549
  - Automatic, Enable 549
  - Introduction 549
- Spelling Options
  - Select 550
- SPEM
  - Base Plug-In 1919
  - Concept 1919
  - Connectors 1920
  - Diagram 1920
  - Diagram, Example 1924
  - Disable 1919
  - Elements 1920
  - Enable 1919
  - MDG Technology 1919
  - Method Content 1919
  - Package 1919

## SPEM

- Process 1919
- Relationships 1920
- Software and Systems Process Engineering Meta-model 1919
- Stereotype Presentation 1920, 1924
- Toolbox Page 1920

## Spring

- Diagram Layout 883

## SQL

- Custom Searches 711
- Editor 711
- Enable Drag And Drop 711
- Patches, Run 601
- Query As Source For Model View 2765
- Query As Source For Standard Charts 2773
- Query As Source For Time Series Charts 2767
- Search Macros 711
- Search, SELECT Statements 711
- Server Data Repository, Connect To 218
- Server Repository, Create 217
- Template Fragments 2712

## SQL Query

- Exclude Package From Report Filters 2658

## SQL Server

- Default Constraint 2371
- Fill Factor, Index 2365
- Fill Factor, Primary Key 2355
- Index 2365
- Non-Clustered Primary Key 2358
- Primary Key, Fill Factor 2355
- Set Up Project On Repository 215

## Stack

- Depth, Control 2536

## Standard Chart

- Artifact 1358

## Standard Colors

- Attribute 616
- Behavior Text 616
- Connector Line 616
- Constraint Element Border 616
- Constraint Element Fill 616
- Constraint Element Text 616
- Element Border 616
- Element Fill 616
- Element Text 616
- Method 616
- Note Compartment Text 616
- Note Element Border 616
- Note Element Fill 616
- Note Element Text 616
- Operation 616

## Options 616

- Screen Element Fill 616
- Screen Fill 616
- Shadow 616

## Standard Element Stereotypes 1457

## Standard Package

- Version Control Usage 385

## Start

- Application 43
- Enterprise Architect 43

## Start Event, BPEL 1.1

- Create 1877
- Model 1877
- Types 1877

## Start Event, BPEL 2.0

- Create 1900
- Model 1900
- Types 1900

## Start Page 68

- Hide 121, 790
- Quick Start 47
- Show 121
- Show When Hidden 784

## Start-Up

- Apply Working Sets 561
- Create Working Sets 561
- Open Diagrams & Views Automatically 561

## State

- Chart 1203
- Composite 1329, 1330
- Diagram 1203
- Element 1329
- Entry And Exit Actions 1015, 1329
- In Timeline Element 1237
- Internal Transition 1449
- Locate In State Machine Diagram 1220
- Locate In State Machine Table 1220
- Operation Behavior 1018
- Reposition In State Machine Table 1219
- Simple 1329
- State Machine Table Conventions 1221
- Transition 1446

## State (Machine)

- Elements and Connectors 807
- Toolbox Pages 807

## State Changes

- Capture, Execution Analysis 2551
- Map, Visual Execution Analyzer 2553
- Record, Visual Execution Analyzer 2553
- Set Up To Capture, Execution Analysis 2551

## State Invariant

- Element 1332, 1334

- State Lifeline
  - Element 1335
- State Lifeline Element
  - Add State 1230
  - Add To Timing Diagram 1229
  - Add Transition 1233
  - Change Transition Time 1232
  - Define Name 1229
  - Delete State 1231
  - Delete Transition 1232
  - Edit State 1231
  - Edit Transition 1232
  - Merge Transitions 1232
  - Move Transition 1233
  - Set Timeline Start Position 1229
  - Sizing and Scale 1229
  - Synchronize Transition 1232
- State Machine
  - Code Engineering Options, C 2266
  - Code Engineering Options, C# 2268
  - Code Engineering Options, C++ 2269
  - Code Engineering Options, Java 2272
  - Create ConnectionPointReference 1210
  - Element 1338
  - Entry And Exit Actions 1015
  - Execution Analysis 2551
  - In Visual Execution Analyzer 2551
  - Include in Execution Analysis Recording 2547
  - Model For Hardware Description Languages 2131
  - Regions 1209
- State Machine Diagram
  - Code Generated From 2128
  - Description 1203
  - Display Format 1203
  - Elements And Connectors 1203
  - Example 1205
  - Generate Code From 2121
  - Generate From Scenario 976, 980
  - Legacy, Generate Code From 2126
  - Locate State In State Machine Table 1220
  - Locate Transition In State Machine Table 1220
  - Locate Trigger In State Machine Table 1220
- State Machine Table
  - Add States 1217
  - Add Substates 1217
  - Add Triggers 1218
  - Cell Color 1213
  - Cell Enumeration 1213
  - Cell Highlights 1213
  - Cell Size 1213
  - Change Position In Diagram View 1216
  - Change Size 1216
  - ChangeTransitions 1218
  - Conventions 1221
  - Description 1212
  - Export To CSV 1222
  - Format 1212
  - Insert Transitions 1218
  - Legend, Add 1219
  - Legend, Remove 1219
  - Locate State In State Machine Diagram 1220
  - Locate Transition In State Machine Diagram 1220
  - Locate Trigger In State Machine Diagram 1220
  - Operations, Overview 1215
  - Options 1213
  - Remove Substate Parent Relation 1217
  - Reposition States 1219
  - Reposition Sub-States 1219
  - Reposition Triggers 1219
  - Simulation 1224
  - State-Next State 1212, 1223
  - State-Trigger 1212, 1222
  - Table Format 1213
  - Trigger-State 1212
- State Region
  - Composite 1209
- State Transition
  - Record In Execution Analysis Recording 2547
- State Transitions
  - Record, Visual Execution Analyzer 2550
- State/Continuation
  - Element 1332
- Statement
  - Trace 2227
- Status Bar
  - Element Coordinates 150
  - Hide 150
  - Show 150
  - Workspace 150
  - Zoom Control 150
- Status Type
  - Color 1159
  - Define 1159
  - For Different Elements 1159
- Step Into
  - Function Calls 2231
- Step Out Of
  - Functions 2231
- Step Over
  - Lines Of Code 2231
- Step Through
  - Function Calls 2548

## Stereotype

- Add Shape Script In Profile 1501
- Add To Profile 1488, 1491, 1531
- Add, Automation Interface Code Example 3006
- Analysis 1800
- And Element Templates 929
- And Metafiles 1452
- Apply To Dependency Relationship 1405
- Apply To Element 1453
- Apply To UML Object 1453
- Associated With Tagged Values 1473
- Automation Interface Repository 2873
- Custom 1461
- Define As Metatype 1508
- Definition 1452
- Dialog 1461
- Edit In Stereotype Profile 1535
- Edit Using Profile Helper 1535
- Extension 1997
- Inbuilt 1997
- Index 2365
- Multiple, Restrict Application Of 1509
- Predefined Tag Types 1499
- Profile, Create With Profile Helper 1529
- Selector 1455
- Set Default Appearance Of Objects In Profile 1502
- Settings 1147
- Show On Project Browser 672
- SPEM Presentation 1920, 1924
- Standard Element 1457
- Synchronize Element With Profile 1473
- Tagged Values In Profile 1492
- Tags, Define 1492
- UML Description 1452
- Visibility 1456
- With Alternative Images 1459

## Stereotype Element

- Define Creation Of Instances 1510

## Stereotyped Element

- Table 1942

## Storage

- Change 293
- Check Dependencies 302
- Copy 293
- Create 293
- Delete 293
- Delete Assets In 304
- Delete Storage Files In 304
- Manage 293
- Password Protection On 293
- Register Asset Packages 298

- Register Files 298
- Register Packages Needed By Asset 302
- Register Technologies 298
- Update 293
- Update Assets In 304

## Storage Files

- Import To Local System 290
- Reusable Asset Service 290

## Store

- Image In Enterprise Architect 860

## Stored Procedure

- As Individual Class 2364
- Definition 2364
- Element 2364
- Select From ODBC Data Source 2378
- Supported Databases 2364

## Straighten

- Line At Cursor 1114

## Strategic Modeling

- Integrated Technology 1481

## String Viewer Dialog 2235

## Structural Diagram

- Elements 1357
- Overview Of Types 1182

## Structural Element

- Add 935
- Dialog 935
- Incorporate Inherited Properties 935
- Modify 935

## Structural Specification

- Generate From Description 965

## StructuralFeature Actions

- Set Structural Feature 1268
- Types 1268

## Structured Activity

- Conditional Node 1338, 1345
- Element 1338, 1340, 1341, 1345
- Loop Node 1338, 1341
- Nested 1338
- Node 1338, 1340
- Sequential Node 1338, 1341
- Simulate 2499

## Structured Activity Node 1340

## Structured Scenario

- Constraints Tab 988

## Structured Specification

- Alternate Path 967
- Alternate Path, Add 970, 972
- Context Menu, Item 972
- Create 968, 970, 972
- Create Element 974
- Delete Path 976

- Structured Specification
  - Delete Step 970, 972
  - Delete Text 974
  - Edit Text 974
  - Entry Points Tab 967
  - Entry Points Tab, Context Menu 976
  - Exception Path 967
  - Exception Path, Add 970, 972
  - External Test Cases 986
  - Floating Toolbar 976
  - Generate From Activity Diagram 985
  - Generate From Clipboard Text 972
  - Glossary Reference, Add 974
  - Insert Context Reference 974
  - Internal Test Cases 986
  - Item Context Menu 972
  - Join Scenarios 976
  - Link Step To Use Case 972, 974
  - Link To Element 974
  - Merge Steps 972
  - Move Steps 972
  - Of Scenario Steps 967, 968
  - Organization 976
  - Set Responsible Entity 972
  - Split Step 974
  - Test Cases 986
  - Text Context Menu 974
  - Toolbar 970
  - Undo Changes 974
- Structured Tagged Value
  - Add Class To Profile 1495
  - Create 1622
- Structured Tagged Value Type 1622
- Style
  - Defining In Document Reports 2653
  - For Connectors 1106
  - Hierarchy 2653
  - Templates 2653
- Style Sheet
  - For Document Reports 2653
  - Hierarchy 2653
  - Selecting 2653
  - Templates 2653
- Style Template
  - Fragments, HTML 2749
- StyleEx
  - Diagram Profile Attribute Values 1573
- Stylesheet
  - Create For Document Reporting 2705
- Sub Activity
  - As Hyperlink 2005
- Sub-Activity
  - Conditional Node 1338, 1345
  - Element 1279, 1338, 1340
  - Loop Node 1338, 1341
- Submachine State
  - Element 1329, 1338
- Submenu
  - Add-In Options 117
  - Advanced, Element 100
  - Alignment 953
  - Appearance (Element) 99
  - Data Management, Tools Menu 110
  - Database Engineering (Tools Menu) 107
  - Debug, Analyzer 114
  - Documentation (Project Menu) 89
  - Hidden, Create In Toolbox Profile 1564
  - Import/Export (Project Menu) 90
  - Inline Features (Element Menu) 101
  - Make Same 953
  - Manage .EAP File, Data Management Menu 111
  - Model Transformations (Tools Menu) 109
  - Model Validation (Project Menu) 89
  - Move 953
  - Online Resources 123
  - Paste Elements (Edit Menu) 82
  - Project Types 120
  - Record (Analyzer) 115
  - Security (Project Menu) 92
  - Size 953
  - Source Code Engineering (Element Menu) 103
  - Source Code Engineering (Tools Menu) 106
  - Space Evenly 953
  - Toolbars 84
  - Version Control (Project Menu) 91
  - Visual Styles 84
  - Web Services (Tools Menu) 108
  - XML Schema (Tools Menu) 108
  - Zoom 868
  - Z-Order 953
- Sub-Menu
  - Hidden In Toolbox 1543
- Subsetted
  - Property, Port 1386
- Subsetted Property
  - Of Attribute 1003
- Subshape
  - Example 1603
  - In Shape Scripts 1603
- Sub-State 1330
  - Reposition In State Machine Table 1219
- Substitute
  - Connector 1443

- Substitute
  - Relationship 1443
- Substitute Words
  - In Extended Report Generator 2665
  - In RTF Report (Legacy) 2725
- Substitution
  - Conditional 1649
  - Connector 1443
  - Direct 1649
  - Macro 1599
  - Relationship 1443
- Subtype
  - Relationship 1120
- Subversion
  - Create Environment For Version Control 397
  - Package Version Control History, In 440
  - Prepare An Environment Under Wine 402
  - TortoiseSVN 403
  - Using With Enterprise Architect Under WINE Crossover 401
  - Verify SVN Workspace 400
  - Version Control Options 420
  - Version Control, Create Local Working Copy 399
  - Version Control, Create Repository Subtree 398
  - Windows-Based Client 401
- Support
  - For Registered Users 40
  - For Trial Users 40
- Supported DBMSs 2375
- Supported Languages
  - Code Engineering 2262
  - Code Generation 2262
  - Code Generation From Behavioral Models, Software Languages 2121
  - Code Generation, Hardware Definition 2080
  - Code Generation, Hardware Description Languages 2121
  - Code Generation, Software 2080
  - Debugging 2074
  - Execution Analysis 2074
  - Hardware Description Languages 2131
  - Hardware Description Languages, Code Generation 2121
  - Recording 2074
  - Reverse Engineering 2136, 2262
  - Software Code Generation 2111
- Suppress
  - Line Segments 1114
- SVN
  - Version Control Options 420
- SVN Workspace
  - Verify 400
- Swimlane Details
  - Dialog 853
- SwimlaneDef
  - Automation Interface, Diagram Package 2958
- Swimlanes
  - Automation Interface, Diagram Package 2960, 2961
  - Manage 852
  - On Diagram 852
  - Orientation 852
  - Set Up 852
- Swimlanes Matrix
  - Activate 854
  - And Matrix Dialog, Matrix Tab 854
  - Create Columns And Rows 854
  - Define Heading 854
  - Delete Items 854
  - Edit Items 854
  - Lock 854
  - Model Profile 854
  - Size 854
- Swimlans
  - Define Single 853
- Switch
  - Between Diagram Tabs 791
  - Document Generator 2654
- Switch To
  - Debugger From Profiler 2558
  - Profiler From Debugger 2231
- Sybase Adaptive Server Anywhere
  - ODBC Driver, Set Up 237
- Sybase ASA
  - Set Up Project On Repository 234
- Synch
  - Element 1346
- Synchronization 310
  - Intial Code In Operations 1019
  - Introduction 2073
  - Macros, Code Template Syntax 1686
  - Of Source Code And Model 2136
- Synchronize
  - Batch With Code 103
  - By Dragging Element From Toolbox 1473
  - Call Argument With Behavior Parameter 1027
  - Class With Code 103
  - Classes, Import Database Schema 2376
  - Code 1638
  - Elements With Profile 1473
  - Existing Code Sections 1640
  - Invocation Argument With Behavior Parameter 1027

- Synchronize
  - Package With Source, (Tools Menu Option) 106
  - Replicas 313
  - Stereotypes From Profile 1473
  - Tagged Values And Constraints 1473
  - Tagged Values From MDG Toolbox Pages 1473
  - UML Profile Tagged Values And Constraints 1473
- Syntax Check
  - For UML, Option 608
- Syntax Grammar
  - Shape Scripts 1613
- Syntax Highlighting
  - Code Editor Options 2251
  - Code Editor, Common 2157
- SysML
  - Activity Elements, Toolbox Page 2295
  - Activity Relationships, Toolbox Page 2295
  - Block Definition Elements, Toolbox Page 2300
  - Block Relationships, Toolbox Page 2300
  - Concepts 2294
  - Design Model 2328
  - Diagrams 2294
  - Disable 2294
  - Extend Profile 1491
  - Flow Direction (Port) 2306
  - Interaction Elements, Toolbox Page 2306
  - Interaction Relationships, Toolbox Page 2306
  - Internal Block Diagram 2328
  - Internal Block Elements, Toolbox Page 2308
  - Internal Block Relationships, Toolbox Page 2308
  - MDG Technology 2294
  - MDG Technology For, Enterprise Architect 1481
  - Migrate 1.1 Model To 1.2 2331
  - Migrate 1.2 Model To 1.3 2331
  - Model Elements, Toolbox Page 2310
  - Model Relationships, Toolbox Page 2310
  - Operational Domain Model 2326
  - Parametric Elements, Toolbox Page 2312
  - Parametric Model, Simulate 2322
  - Parametric Models 2320
  - Parametric Relationships, Toolbox Page 2312
  - Requirement Elements, Toolbox Page 2314
  - Requirement Extensions, Toolbox Page 2314
  - Requirement Relationships, Toolbox Page 2314
  - Requirements Model 2325
  - Reusable Subsystems 2330
  - State Machine Elements, Toolbox Page 2316
  - State Machine Relationships, Toolbox Page 2316
  - Toolbox Pages 2294
  - Use Case Elements, Toolbox Page 2318
  - Use Case Relationships, Toolbox Page 2318
- System
  - Testing 2607
  - Users 323
- System Boundary
  - Element 1347
  - Insert New From Toolbar 140
- System Design,
  - Compose, Systems Engineering Modeling 2328
- System Document Template
  - Description 2648
- System Engineering Modeling
  - Create Reusable Subsystems 2330
- System Output
  - ArcGIS Model Validation Tab 1970
- System Output Window
  - Context Menu 169
- System Window
  - Project Glossary Tab 534
  - Project Issues Tab 528
  - Project Tasks Tab 526
- SystemC
  - Code Generation 2275
  - Diagram Toolbox Pages 2100
  - Language Options 2275
  - Modeling Conventions 2100
- Systems Engineering Edition
  - Of Enterprise Architect 20
- Systems Engineering Model
  - Create 2288
- Systems Engineering Modeling
  - Compose System Design 2328
  - Create SysML Parametric Model 2320
  - Overview 2288
  - Process 2288
  - Simulate SysML Parametric Model 2322
  - SysLM Operational Domain Model 2326
  - SysML Requirements Model 2325
- Systems Modeling Languages
  - MDG Technology For, Enterprise Architect 1481
- Systems Modelling Language (SysML) 2294
- T -
- Tab
  - Asterisks 790
  - Close 790

- Tab
  - Context Menu 790
  - Context References 987
  - Diagram 790
  - Operation, System Output Window 169
  - View 790
- Tabbed Frame
  - Autohide Windows In 132
  - Combine Windows In 128
  - Remove Window From 128
  - Reveal Autohidden Windows In 132
- Table
  - Change Owner, Database Engineering 2347
  - Create 2339
  - Create Using Document Editor 1075
  - Data Modeling 2338
  - Data Type Conversion 2350
  - DDL Script For 1937
  - Detail 1942
  - Document Report, Apply Styles In MS Word (Legacy) 2734
  - Document Report, Manipulate In MS Word (Legacy) 2734
  - Document Report, Resize In MS Word (Legacy) 2734
  - Element 1942
  - In Documents 1075
  - Introduction 2338
  - Owner Tagged Value 2342
  - Properties 1942
  - Set Database Type 2341
  - Set Owner 2342
  - Set Properties 2340
  - State Machine 1212
- Table Aggregation
  - For DecisionTable 1931
- Table Of Contents
  - For Document Report, Add In MS Word 2733
  - Select Template For Document Report 2652
- Table Of Figures
  - For Document Report, Add In MS Word 2733
- Tables in Document Reports
  - For Document Sections 2700
- Tables of Contents
  - Create For Document Reporting 2706
- Tabs
  - Diagram, Switch Between 791
  - Set, Document Editor 1067
- Tabular Sections
  - In Documents 1075
- Tag
  - Compartment, Element 955
  - Management, Advanced 1140
  - Profile 1492
- Tag Grouping
  - In Stereotyped Elements, Define With Special Attributes 1514
- Tag Type
  - Predefined, Assign To Stereotype 1499
- Tagged Region
  - Code Editor Find And Replace 2161
- Tagged Value
  - Add 1136
  - Add To Operations 1022
  - Assign Information To 1138
  - Assign To Item 1137
  - Associated With Stereotype 1473
  - Behavior Parameters 1030
  - By Dragging Element From Toolbox 1473
  - Connector, Use 1498
  - Custom, Create 1626
  - Drag Onto New Element 989
  - Duplicate Values 1134
  - Element Package, Automation Interface 2907
  - Entity Relationship Diagram 1939
  - For Oracle Table Properties 2343
  - For Schema Owner 2342
  - For Table Owner 2342
  - Fully Qualified, Show 845
  - Fully-Qualified Value 1134
  - In BPMN Simulation 2523
  - In UML Profiles 1473
  - Include In Document Report 2698
  - Inherited, Show 845
  - Insert Column In Package Browser 679
  - Macros, Code Template Syntax 1676
  - Masked, Create 1626
  - Model Components And 1134
  - Modify 1134, 1138
  - Modify Value 1137
  - Of Attributes 1005
  - Quick Add 1136
  - Reference Data, Create 1628
  - Section, Document Template Designer 2698
  - Show Duplicates 1139
  - Structured, Create 1622
  - Synchronize, And Constraints 1473
  - Synchronize, From MDG Toolbox Pages 1473
  - Toolbar 1134
  - Types 1150
  - Types Of Value Field 1137
  - Types, Export As Reference Data 376
  - Use In Kanban Charts 857
  - View 1134



- Tagged Value
  - What Is A? 1134
  - Window 1134
- Tagged Value Broadcast Events
  - Add-In Model 3084
  - EA\_OnAttributeTagEdit 3084
  - EA\_OnConnectorTagEdit 3085
  - EA\_OnElementTagEdit 3086
  - EA\_OnMethodTagEdit 3087
- Tagged Value Type
  - Add In Specification Manager 1758
  - Add To MDG Technology 1551
  - Filters 1622
  - Introduction 1621
  - MatrixOverlay 733
  - Predefined Structured 1622
  - Reference Data, Predefined 1629
- TaggedValue
  - Automation Interface, Element Package 2907
- Target
  - Role 1132
  - Set For Message 1124
- Target Types
  - Implementation Report, Change 2739
- Task
  - Assign Multiple Resources 515
  - Auto Naming 527
  - Auto Numbering 527
  - Automation Interface Repository 2874
  - Completion 512
  - Details 527
  - Element 2010
  - Items, Element Maintenance 2623
  - Overdue, Highlight 538, 542, 594
- Task Guide
  - Learning Center Topic 74
- Task Management
  - In Project Calendar 581
- Tasks Report
  - On Elements, Generate 2740
- TCF
  - Value 585
  - Weighting 585
- Team
  - Develop Projects In, Introduction 306
  - Development, Introduction 197
  - Development, Project Sharing 307
- Team Deployment
  - And Version Control 389
  - Version Control Branching 389
- Team Foundation Server
  - Version Control Options 421
- Version Control, Create Local Working Copy 409
- Version Control, Exclusive Checkout 411
- Team Foundation Server (TFS)
  - Create Environment For Version Control 408
- Team Review
  - Access 343
  - Access From Shortcut 204, 207
  - Access From Shortcut (Direct Definition) 206
  - Add External File Link To Post 359
  - Add New Category 350
  - Add New Document 353
  - Add New Topic 351
  - Add Object Link To Post 359
  - Add Team Review Link To Post 359
  - Capture Diagram Image As Resource 361
  - Category 343
  - Comment (Reply) 343
  - Comment On Document 355
  - Connections To Other Team Reviews 363
  - Context Menu Options 346
  - Copy Path To Clipboard 346
  - Create Category 350
  - Create Document 353
  - Create Topic 351
  - Delete Category 346
  - Delete Document 346
  - Delete Resource 346
  - Delete Resources 361
  - Delete Topic 346
  - Document 343
  - Edit Item 358
  - Editor 358
  - Export Package As Resource 361
  - Formal Reviews Category 343
  - Hyperlink To 2002
  - Icons 343
  - Import Resource Package 361
  - Introduction 343
  - Item 343
  - Link To Clipboard Images 361
  - Link To Active Profiler Report 361
  - Link To Active Website 361
  - Load Data When Required 362
  - Loading Behavior 362
  - Mark All Posts Unread 362
  - Options 362
  - Password Protect Items 357
  - Postings in Model Views 686
  - Preload 362
  - Reply To Document 355
  - Resources 361

- Team Review
  - Save Profiler Report As Resource 2566
  - Search 362
  - Tab 343
  - Topic 343
  - View Diagram Image Resource 361
  - Was Discussion Forum 343
  - Window 343
- Technical Complexity Factor
  - Estimate Project Size 588
  - Value 585
  - Weighting 585
- Technology
  - Create Learning Center Pages To Support 74
- Technology Developer
  - And Enterprise Architect 193
  - Project Role 193
- Technology Event
  - EA\_OnInitializeTechnologies 3089
- Technology Events
  - Add-In Model 3088
  - EA\_OnDeleteTechnology 3092
  - EA\_OnImportTechnology 3094
  - EA\_OnPostActivateTechnology 3090
  - EA\_OnPreActivateTechnology 3089
  - EA\_OnPreDeleteTechnology 3091
- Technology File
  - Generate, Option 104
  - Import, Option 104
- Technology-Defined Model View
  - Set Up 693
- Template
  - Applying Stylesheets 2648
  - Behavioral Model 1688
  - Code Generation, Call From Other Template 1700
  - Cover Page, Create For Document Reports 2707
  - Custom Document Report, Scroll And Search Options 1050
  - Custom Document, Scroll And Search Options 1059
  - Custom Linked Document, Scroll And Search Options 1050
  - Document Report, Design 2684
  - Document Report, Print Options 1094
  - Document, Import 2707
  - Editor In MDG Development 1701
  - Editor, Code Templates 1641
  - Element 1472
  - Fragments 2708
  - Fragments, Add To Document Template 2718
  - Fragments, Edit In Document Template 2718
  - Fragments, Override In Document Template 2718
  - Fragments, Switch In Document Template 2718
  - Fragments, View In Document template 2718
  - Instantiated 1365
  - Linked Document, Print Options 1094
  - Linked Document, Scroll And Search Options 1059
  - Masked Tagged Value 1626
  - Model, Incorporate In Technology 1576
  - Package, Settings 929
  - Parameterized Classes 1365
  - Report, Selecting 2648
  - Script 2792
  - Stylesheet, Create For Document Reports 2705
  - System Document, Description 2648
  - Tables of Contents, Create For Document Reports 2706
  - Transformation, Call From Other Template 1700
  - Transformation, Default 2052
- Template Binding
  - Connector 1444
  - Transformation, Parameter Substitution Macros 1677
- Template Fragment
  - Create 2709
  - Custom Query 2711
  - Script 2714
  - Script Output, Example 2717
  - Script, Example 2716
  - SQL 2712
- Template Parameter
  - Field Substitution Macros 1677
  - Substitution In MDA-Style Transformation 2069
  - Transformation 2069
- TemplateBinding
  - Automation Interface, Connector Package 2946
- TemplateParameter
  - Automation Interface, ElementFeatures Package 2931
- Templates Tab (Enhanced) 2681
- Term
  - Automation Interface Repository 2876
- Terminate
  - Element 1350
- Test
  - Application, Analyzer Menu Option 112
  - Automation Interface, Element Package 2908
  - Copy Between Categories 2610
  - Create Defect From 2615
  - Documentation 2617

- Test
  - Field Substitution Macros 1677
  - Generate From Scenario 986
  - Model Template 765
  - Move Between Categories 2610
  - Report 2617
  - Result Output 2617
  - Script Output 2617
  - Unit, In Execution Analysis 2573
  - Unit, Record Results In Execution Analysis 2575
  - Unit, Run In Execution Analysis 2575
  - Unit, Set Up In Execution Analysis 2573
- Test Case
  - Element 2010
- Test Cases
  - Generate From Scenario 986
- Test Cut 2577, 2580
  - Element 2588
  - Element, Create 2586
- Test Details
  - Dialog, Customize Status 1165
- Test Details Dialog 2607
- Test Domain 2577
  - Elements And Connectors 819
  - Toolbox Page 819
- Test Domain Diagram
  - Create 2586
  - Generate 2586
- Test Field
  - Value Type, Define 1165
- Test Item
  - Drag Onto New Element 989
- Test Script
  - Execute, Build Toolbar Option 147
  - Introduction 2182
  - JUnit 2573
  - NUnit 2573
- Test Scripts
  - Compartment, Show 2616
- Test Set 2577
  - Element 2588
  - Element, Create 2586
- Test Suite 2577
  - Element 2589
  - Element, Create 2586
- Tester
  - And Enterprise Architect 191
  - Project Role 191
- Testing
  - A Model 2593
  - Acceptance 2607
  - Asterisk On Testing Window Tabs 2605
  - Autonaming 2605
  - Compartment, Element 955
  - Import Element Scenarios 2611
  - Import Internal Constraint 2614
  - Import Internal Requirement 2614
  - Import Package Scenarios 2611
  - Import Responsibility 2614
  - Import Test From Other Element 2613
  - Integration 2607
  - MDG Technology For, Enterprise Architect 1481
  - Model Template 755, 765
  - Overview 2604
  - Report, Generate 2742
  - Scenario 2607
  - Support 2604
  - System 2607
  - Type, Define 1170
  - Unit 2607
  - Window 2605
  - Window, Acceptance Test Tab 2607
  - Window, Integration Test Tab 2607
  - Window, Scenario Test Tab 2607
  - Window, System Test Tab 2607
  - Window, Unit Test Tab 2607
  - Workspace 2605
  - Workspace, Customize Status 1165
- Testing Details Report
  - Generate 2736
- Testing Feature
  - Insert In Element 1043
- Testpoint 2184
  - Combine 2586
  - Constraint Composition 2581
  - Diagram 819
  - Examples 2581
  - Introduction 2577
  - Management 2577
  - Testpoint Editor Dialog 2584
  - Toolbox Page 819
  - Window 2578
  - Window Toolbar 2580
- Testpoint Management
  - Constraint Composition 2581
  - Testpoint Editor Dialog 2584
  - Testpoints Window 2578
- Testpoints 2184
- Text Element
  - Create 923
  - Insert New From Toolbar 140
- Text File

## Text File

Create Element From 1773

## TFS

Create Environment For Version Control 408

Package Version Control History, In 440

Version Control Options 421

Version Control, Create Local Working Copy 409

Version Control, Exclusive Checkout 411

Working Folder For Enterprise Architect Version Control 421

Workspace For Enterprise Architect Version Control 421

## The Open Group Architecture Framework

MDG Technology For, Enterprise Architect 1481

## Theme

Apply To Reports 2644

Diagram, Built-In 611

Diagram, User Defined 611

Microsoft Office Styles 168

Microsoft Visual Basic Styles 168

Options 168

Visual Style 84

## Thesaurus

In Document Editor 1089

## Tidy Line Angles 1114

## Tile

For Diagram Background 613, 622

## Time Event 1327

## Time Interval

Compress 1244, 1246

Context Menu 1241

Copy and Paste 1246

Create 1241

Delete 1241

Description 1241

Move 1241

Operations 1246

Resize 1241

Select 1245

Shift Left Or Right 1246

Transitions 1246

## Time Range

Set For Timing Diagram 1228

## Time Series Chart

Artifact 1358

Change Appearance 2785

Define 2767

Example 2767

Set Frequency 2767

Use Of Cloud Services 267

## Timeline Element States

Add Via Configure Timeline Dialog 1237

Delete via Configure Timeline Dialog 1237

Edit Via Configure Timeline Dialog 1237

Maintain 1237

Numeric Range Generator 1239

## Timeline Range

Set For Timing Diagram 1228

## Timeline Start Position

Set For State Lifeline Element 1229

## Timing

Constraint 1424

Details, Change 1424

Elements and Connectors 806

Message 1431

Message, Create 1432

Observation 1424

Toolbox Pages 806

## Timing Diagram

Add Value Lifeline Element 1234

Create 1227

Description 1225

Edit Options 1228

Edit Value Lifeline Element 1234

Elements And Connectors 1225

Example 1227

Set Time Range 1228

## TOGAF

MDG Technology For, Enterprise Architect 1481

## Tomcat

Server, Configuration 2199

Server, Debugging 2195

Service Configuration 2199

## Toolbar

(File) Search, Debugging 2164

Add Commands 153

Add Menu Commands 156

Add Toolbar Commands 156

Build 147

Change Command Icon Appearance 153

Code Generation 144

Console Tab, Scripting Window 2796

Create 156

Current Connector 143

Current Element 141

Customize 156

Customize Button, Show/Hide 84

Debug 146

Default Tools 135

Diagram 138

Diagram List 679

## Toolbar

- Display Labels 156
- Docked 134
- Element 140
- Floating Buttons, Display 631
- Floating, Structural Specification 976
- Format (Element Appearance) 785
- Hide 156
- Large Icons 163
- Manage Searches 709
- Missing Icon 152
- Model Search 700
- Notes 1143
- Options, Customize Appearance 163
- Package Browser 679
- Project 136
- Project Browser 669
- Record & Analyze 148
- Record & Analyze Window 2544
- Remove 156
- Remove Commands 153
- Rename 156
- Screen Tips 163
- Scripting Window 2792
- Show 156
- Submenu 84
- Testpoints Window 2580
- UML Elements 140
- Workspace 134
- Workspace Layouts 149

## Toolbox

- Activity Pages 808
- Add Stereotype To Diagram Element 792
- Add To MDG Technology 1551
- Analysis Pages 812
- Appearance Options 797
- ArcGIS Core 1946
- ArcGIS Geometric Network 1946
- ArcGIS Topology 1946
- ArchiMate Groups 1926
- Artifact 2763
- BPMN 1.0 1848
- BPMN 1.1 1848
- BPMN 2.0 1851
- BPMN 2.0 Business Process 1852
- BPMN 2.0 Choreography 1854
- BPMN 2.0 Collaboration 1856
- BPMN 2.0 Conversation 1859
- BPMN 2.0 Type 1861
- BPMN Group 1845
- Business Modeling Group 1805
- Chart Item 2763

- Class Pages 803
- Collapse Page 792
- Common Page 800
- Communication Pages 805
- Component Pages 809
- Composite Pages 804
- Connectors For Extending, MDG Technology 1569
- Create Elements And Connectors 792
- Custom Pages 813
- Custom, Page Attributes 1563
- Customize 1538, 1560
- Dashboard 2763
- Data Flow Diagram Group 1797
- Data Modeling Page 818
- Default, Override In Profile 1567
- Deployment Pages 809
- Documentation Group 2669
- Elements For Extending, MDG Technology 1567
- Entity Relationship Diagram 1939
- Eriksson-Penker Group 1929
- Expand Page 792
- Gang Of Four Pattern Group 2285
- GML Elements 1984
- GML Relationships 1984
- GoF Pattern Group 2285
- Hidden, Sub-Menu 1543
- Hide 792
- Hide Labels 797
- ICONIX Group 2282
- Interaction Pages 806
- Maintenance Pages 815
- MDG Technology Groups 1476
- Metamodel Pages 812
- Mind Mapping Group 1794
- Object Pages 804
- ODM Pages 1972
- Override Default In Toolbox Profile 1567
- OWL Pages 1972
- Page, Change Sequence In Profile 1561
- Pin Pages 797
- Profile Pages 810
- Profile, Change Icon Order 1561
- Profile, Change Toolbox Page Order 1561
- Profile, Create For MDG Technology 1561
- Profiles 1538, 1543, 1560
- RDF Pages 1972
- Requirement Pages 814
- Set Toolbox Visibility 797
- Shortcut Menu 799
- Show 792

- Toolbox
  - Show Labels 797
  - SoaML Pages 2451
  - SOMF 2.1 Pages 2454
  - SPEM 1920
  - State (Machine) Pages 807
  - Synchronize Stereotyped Tagged Values 792
  - SysML Groups 2294
  - SystemC Group 2100
  - Test Domain Page 819
  - Timing Pages 806
  - Unpin Pages 797
  - Use Case Pages 801
  - User Interface Pages 816
  - Verilog Group 2105
  - VHDL Group 2106
  - WSDL Page 817
  - XML Schema Pages 818
- Toolbox Profile
  - Create Hidden Submenu In 1564
  - Custom Items, Assign Icons For 1565
  - Pages That Can be Overridden 1567
  - Submenu Items, Assign Icons For 1564
- Tools
  - Custom 158
  - External Applications 158
  - For Modeling 67
  - Menu 104
  - Menu, Customize Applications Available 157
  - Pass Parameters To External 160
- Tooltips
  - Display On Diagram 608
- Top Margin
  - Sequence Diagram, Change 1258
- Topic
  - Add To Team Review 351
  - Bind To Project Browser Package 346
  - Create 351
  - Delete 346
  - Password Protect Text 357
- Topology
  - ArcGIS, Example 1954
  - Package, ArcGIS 1954
- TopologyMembership
  - Attribute, ArcGIS 1954
- TopologyRule
  - Connector, ArcGIS 1954
- TortoiseCVS
  - In Version Control 407
- TortoiseSVN
  - In Version Control 403
- Trace
  - Connector 1446
  - Relationship 1446
  - Statements 2227
  - Variable Changes 2230
- Trace Statement
  - Specifying 2227
- Trace Statements
  - Output 2215
- Trace Tools 644
- Traceability
  - Diagrams 743
  - For Diagram 725
  - For Elements 725
  - In Requirements Models 1767
  - Introduction 723
  - Themes 723
  - Tools 724
  - Window 725
  - Window, In Traceability 724
  - With Dependency Report 724
  - With Implementation Report 724
  - With Relationship Matrix 724
  - With Traceability Window 724
- Traceability Window
  - In The Specification Manager 1744
- Tracepoint
  - In Code 2227
  - Marker 2227
  - Set 2227
  - Statement Output 2215
- Track Changes
  - Auditing 445
  - Baselines 445
  - Introduction 445
- Track Changes In Document Editor
  - Accept Changes 1091
  - Hide Changes 1091
  - Locate Changes 1091
  - Reject Changes 1091
- Trademarks 29
- Transfer
  - Project Data Between Repositories 504
- Transform
  - Connector End 2066
  - Connectors 2063
  - Copy Information 2066
  - Copy Tagged Values 2066
  - Duplication Of Connectors 2063
  - Elements, MDA-Style Transformations 2017
  - Model, MDA-Style Transformations 2017
  - Names 2067
- Transform Package

- Transform Package
  - Package Context Menu 654
- TRANSFORM\_CLASSIFIER
  - Macro 2068
- TRANSFORM\_CURRENT
  - Macro 2063, 2066
- TRANSFORM\_REFERENCE
  - Macro 2063, 2068
- TRANSFORM\_TAGS
  - Macro 2066
- Transformation
  - Built In, List Of 2020
  - C# 2021
  - C++ 2023
  - Communication Diagram To Sequence Diagram 2041
  - Connector Syntax 2060
  - Data Model To ERD 2024
  - DDL 2025
  - Debug Intermediary Language 2053
  - Dependencies 2013
  - EJB 2029
  - Entity Bean 2029
  - ERD To Data Model 2032
  - Foreign Keys 2065
  - Java 2034
  - JUnit 2037
  - NUnit 2038
  - Object Syntax 2055
  - PHP 2040
  - Sequence Diagram To Communication Diagram 2041
  - Session Bean 2029
  - Template Parameter Substitution 2069
  - Template, Call From Other Template 1700
  - VB.Net 2042
  - Write 2051
  - WSDL 2043
  - XSD 2044
- Transformation Dependency
  - Trace With Traceability Window 725
- Transformation Template
  - Default 2052
  - Modify 2048
  - Transfer Between Models, Export 1637
  - Transfer Between Models, Import 1637
- Transition
  - Add To State Lifeline Elements 1232
  - Add Via Configure Timeline Dialog 1239
  - Change In State Machine Table 1218
  - Change Time, State Lifeline Element 1232
  - Connector 1446
  - Context Menu Options 1103
  - Delete On State Lifeline Element 1232
  - Delete Via Configure Timeline Dialog 1239
  - Edit In Time Intervals 1246
  - Edit On State Lifeline Elements 1232
  - Edit Via Configure Timeline Dialog 1239
  - Effect 1446
  - Guard 1446
  - Highlight Associated Trigger or State 1218
  - Insert In State Machine Table 1218
  - Internal (On State) 1449
  - Locate In State Machine Diagram 1220
  - Locate In State Machine Table 1220
  - Merge On State Lifeline Element 1232
  - Move On State Lifeline Elements 1232
  - Properties 1446
  - Relationship 1446
  - State Machine Table Conventions 1221
  - Trigger 1446
- Transitions Collection
  - Automation Interface, ElementFeatures Package 2932
- Translation
  - Document Report Generation 2666
- Tree Style Hierarchy
  - Create 1125
  - Set Default Link Style 1125
- Trial Version
  - Extend Trial Period 18
  - Of Enterprise Architect 18
  - Select Edition To Trial 18
  - Select Workspace Layout 18
- Triangle
  - Red 591
- Tricks and Traps
  - Create Add-In 3016
- Trigger
  - Auto-Firing 2512
  - Create 1350
  - Create In State Machine Table 1218
  - Create In Transition Properties 1446
  - Data Modeling, Create 2369
  - Data Modeling, What Is A? 2368
  - Element 1350
  - Find In Project Browser 1446
  - For Transition 1446
  - Intermediate Event, BPEL 1.1 1878
  - Intermediate Event, BPEL 2.0 1901
  - Locate 2494
  - Locate In State Machine Diagram 1220
  - Locate In State Machine Table 1220
  - Parameters 2510

## Trigger

- Ports 1350
- Properties Tab 1350
- Reposition In State Machine Table 1219
- Resignalling 2509
- Sets 2512
- Sets To Simulate Event Sequence 2515
- Simulation 2494
- Simulation Events Window 2494
- Start Event, BPEL 1.1 1877
- Start Event, BPEL 2.0 1900
- State Machine Table Conventions 1221
- Type 1446
- Use Of 2494
- Waiting 2508

## Trigger Property

- Find In Project Browser 1103

## Type Hierarchy Dialog 908

## Type Libraries

- For Script Editor 2798

## Type Specific Menu Section

- Connector 1103

**- U -**

## UAC

- And Debugging 2187

## UI Control

- Element 1992

## Ultimate Edition

- Of Enterprise Architect 20

## UML

- 1.3 473, 475, 597
- 1.3, Import From XMI 478
- 1.4 475
- 1.4, Import From XMI 478
- 1.5 631
- 2.0 Migration 597
- 2.0, Import From XMI 478
- 2.3 - Definition 9
- Analysis Tool - Enterprise Architect 5
- Build Systems 9
- Connectors 1389
- Data Modeling Profile 1937
- Default Mappings To XML Schema 2415
- Default Mappings To XSD 2415
- Definition 1179
- Design Systems 9
- Design Tool - Enterprise Architect 5
- Dictionary 1179
- DTD 483

## Elements 1265

## Enterprise Architect Modeling Platform 750

## Extend 1179

## Extending 1471

## Manage Complexity 9

## Model Complexity 9

## Model Structure, Manage - Enterprise Architect 9

## Models Under Single Root 480

## Recommended Reading 1179

## Shared Model Development - Enterprise Architect 9

## Simulation Of Activity, Compare With BPMN Process 2524

## Support - Enterprise Architect 9

## Syntax Checking Option 608

## Visualize Systems - Enterprise Architect 9

## UML 2.1

## Data Modeling Notation 2384

## UML Behavioral Diagram

## Overview 1198

## UML Business Process Model

## In Enterprise Architect 755, 758

## UML Class Model

## In Enterprise Architect 755, 761

## UML Component Model

## In Enterprise Architect 755, 763

## UML Database Model

## In Enterprise Architect 755, 762

## UML Deployment Model

## In Enterprise Architect 755, 764

## UML Diagram

## Add To Project 822

## Copy, Deep 842

## Copy, Shallow 842

## Create 822

## Duplicate 842

## Extended 1181, 1793

## In Enterprise Architect 778

## MDG Technology 1181

## Overview 1181

## Paste 842

## Types 1181

## What Is A? 1181

## UML Domain Model

## In Enterprise Architect 755, 760

## UML Element

## Behavioral Diagram Elements 1265

## Structural Diagram Elements 1357

## Toolbar 140

## UML Maintenance Model

## In Enterprise Architect 755, 766



- UML Model
  - Add Connectors In Enterprise Architect, Tutorial 55
  - Add Diagram In Enterprise Architect, Tutorial 50
  - Add Element In Enterprise Architect, Tutorial 52
  - Add Element To Diagram, Tutorial 52
  - Add Package In Enterprise Architect, Tutorial 50
  - Add View, Tutorial 49
  - CSV Import And Export In Enterprise Architect 497
  - Define Connector Properties In Enterprise Architect 56
  - Define Element Properties In Enterprise Architect 56
  - Domain Based 1789
  - Project Roles And Tasks In Enterprise Architect 183
  - Specialized 1789
- UML Model Pattern
  - Introduction 755
- UML Model Template
  - In Enterprise Architect 755
  - Introduction 755
- UML Modeling
  - And MDG Technologies 750
  - And Requirements Management In Enterprise Architect 750
  - Build Models 750
  - Business Modeling In Enterprise Architect 750
  - What Is? 750
  - With Enterprise Architect 750
  - With Packages 772
  - With UML Patterns In Enterprise Architect 750
  - With UML Profiles In Enterprise Architect 750
  - With UML Stereotypes In Enterprise Architect 750
- UML Modeling Tool
  - Enterprise Architect - Key Features 13
- UML Pattern
  - Actions 1464
  - Add To Diagram 1467, 1469
  - Create From Diagram 1464
  - In Resources View 1464
  - Save 1464
  - Save From Diagram 1464
- UML Profile
  - And Element Templates 1472, 1485
  - Import From XML 1472, 1485
  - Import To Resources Window 1526
  - Stereotypes 1472, 1485
- Synchronize Elements 1473
- Synchronize Stereotypes 1473
- Synchronize Tagged Values And Constraints 1473
- UML Project Management Model
  - In Enterprise Architect 755
- UML Project Model
  - In Enterprise Architect 767
- UML Requirements Model
  - In Enterprise Architect 755, 758
- UML Resources
  - Patterns 1464, 1466
- UML Structural Diagram
  - Overview 1182
- UML Syntax Compliance
  - Turn Off 2596
- UML Testing Model
  - In Enterprise Architect 755, 765
- UML Types
  - Cardinality (Multiplicity) 1151
  - Dialog 1146
  - Stereotypes 1147
  - Tagged Values 1150
- UML Use Case Model
  - In Enterprise Architect 755, 759
- UML\_EA.DTD 483
  - File 478
- UML2
  - File Import 480
- Unadjusted Use Case Points 588
- Undo
  - Last Action, Diagram Edits 873
  - Option (Edit Menu) 80
- Unicode Support 43, 605
- Unified Modeling Language 1179
- Union Element
  - Member Elements 2408
  - XML, Toolbox Icon 2408
- Unique
  - Index 2365
- Unit Test Command
  - Introduction 2182
- Unit Test Script
  - Create 2182
- Unit Testing 2607
  - Define Tests, Execution Analysis 2573
  - Introduction, Execution Analysis 2573
  - JUnit 2573
  - NUnit 2573
  - Record Test Results, Execution Analysis 2575
  - Run, Execution Analysis 2575
  - Set Up, Execution Analysis 2573

- Unlock
  - Connector 332
  - Element 332
- Unpin
  - Toolbox Pages 797
- Update
  - Element Phase, For Package 590
  - Element Status, For Package 590
  - Element Version, For Package 590
  - Package Phase 590
  - Package Status 590
  - Package Version 590
- Update All Packages
  - To Latest Revision, Version Control 435
- Update Index Statistics
  - On Firebird Project File 110
- Update Package
  - To Latest Revision, Version Control 435
- Upgrade
  - Existing License (v. 6.5 And Earlier) 3169
  - Existing License (v. 7.0+) 3169
  - Model 600
  - Model, Upgrade Wizard 599
  - Project 599
  - Replicas 315, 600
  - Wizard 599
- Usage
  - Connector 1450
  - Of Element 910
  - Relationship 1450
- Use
  - Pattern 1467
  - Spell Checker 553
- Use (Class Diagram Usage)
  - Connector 1450
  - Relationship 1450
- Use (Use Case Connector)
  - Connector 1450
  - Relationship 1450
- Use Case
  - And Requirements 1765
  - Arrowhead, Show 1125
  - Create For Method 2155
  - Element 1352
  - Elements and Connectors 801
  - Extension Points 1354
  - Keyword 588
  - Link To Method 2155
  - Metrics 584
  - Metrics Dialog 588
  - Model Template 755, 759
  - Phase 588
  - Points, Unadjusted 588
  - Quick Generation Of Realization 1779
  - Scenarios 1253
  - Toolbox Pages 801
  - View 769
- Use Case Diagram
  - Description 1201
  - Elements And Connectors 1201
  - Example 1203
- Use Element Extras
  - Automation Interface Code Example 3001
- Use Repository Extras
  - Automation Interface Code Example 3004
- User
  - Default Fonts 624
  - Defined Searches, Storage 711
  - Dictionary, For Spell Checker 554
  - Directory 1153
  - Forum 40
  - Groups 323
  - ID, Import From Active Directory 324
  - Settings 2256
- User Default Diagram
  - Redisplay 784
- User Interface
  - Components 67
  - Control Element 1992
  - Customization 167
  - Customized 151
  - Element 1992
  - Elements and Connectors 816
  - MDG Win32® User Interface Technology 1996
  - Model Template 755
  - Screen Prototype 1991
  - Toolbox Pages 816
- User Interface Diagram
  - Description 1990
  - Elements And Connectors 1990
  - Example 1991
- User Lock
  - Identify Owner 342
  - Indicators 340
- User Security
  - Add Connector To Locked Element 388
  - Apply User Lock 339
  - Assign User To Group 326
  - Basics 316
  - Change Password 334
  - Disable 318
  - Enable 318
  - Identify Lock Owner 342
  - Import User ID From Active Directory 324

## User Security

- List Of Permissions 329
- Lock Elements & Diagrams 336
- Lock Elements & Diagrams, User/Group Locking 336
- Lock Packages 336
- Lock Packages, User/Group Locking 337
- Locked Element Indicators 340
- Maintain Groups 320
- Maintain Users 323
- Manage User-level Locks. 342
- Password Encryption 333
- Policy 319
- Re-enable 318
- Release Elements & Diagrams 336
- Release Elements & Diagrams, User/Group Locking 336
- Release Package, User/Group Locking 337
- Release Packages 336
- Reset Password 334
- Set Group Permissions 321
- Set Password 334
- Set Up Single Permissions 327
- Tasks 316
- View All Permissions 328
- View And Manage Locks 332
- What Is User Security? 316

## User Security Groups

- Assign User To 326
- Set Up 326

## User Story

- Artifact 1358

## User/Group Lock Policy 319

## Uses Context List Facility 974

## Using Enterprise Architect

- Remove Recent Project 73
- Start Page 68

## Utility

- Compare 457
- Diff 457

## UUCP

- Estimate Project Size 588

**- V -**

## Validate

- ArcGIS Workspace 1970
- BPEL Model 1917
- Business Rules 1841
- Model To Generate BPEL 1.1 1893
- Package Configuration For Version Control 443
- Rule Composer 1841

## Validation

- Cancel 89
- Of Model, Configure For MDG Technology 1576

## Validation, Element Composition

- Rules 2600

## Validation, OCL Conformance

- Rules 2601

## Validation, Of Model

- Cancel 2596
- Configure 2596
- Execute 2596
- Of Diagram 2594
- Of Element 2594
- Of Package 2594
- Rules 2597
- Run 2596

## Validation, Properties

- Attribute 2600
- Element 2600
- Feature 2600
- Relationship 2600

## Validation, Well Formedness

- Attribute 2599
- Diagram 2599
- Element 2599
- Feature 2599
- Relationship 2599

## Value Lifeline Element 1355

- Add States 1235
- Add To Timing Diagram 1234
- Add Transitions 1235
- Change Transition Time 1235
- Delete Transitions 1235
- Edit Transitions 1235
- Sizing and Scale 1234
- States 1234
- Transitions 1234

## ValueOf Field

- Document Template Designer 2698

## ValueSpecificationAction

- Simulation 2497

## Variable

- Comparison Of Values 2236
- Debug, Break On Change In Value 2229
- Definitions, Code Template Syntax 1644
- Definitions, Examples 1644
- References, Code Template Syntax 1644
- References, Examples 1644
- Snapshots 2236
- Trace Change In Value 2230

## Variable Actions

- AddVariableValue Action 1273

- Variable Actions
  - ClearVariable Action 1273
  - ReadVariable Action 1273
  - RemoveVariable Action 1273
  - WriteVariable Action 1273
- VB
  - Set Up In Automation Interface 2808
- VB.NET
  - Code Generation 2275
  - Language Options 2275
  - Modeling Conventions 2102
  - Transformation 2042
- VBScript 2791, 2792
- Verify
  - CVS Workspace 406
  - SVN Workspace 400
- Verilog
  - Code Generation 2276
  - Diagram Toolbox Pages 2105
  - Language Options 2276
  - Modeling Conventions 2105
- Version Control
  - And Reference Data 387
  - And Replication 422
  - And Sequence Diagram 1254
  - Apply To Enterprise Architect Model 386
  - Apply To Model Branch 426
  - Basics 387
  - Branching 389, 394
  - Check In Model Branch 434
  - Check In Package 432
  - Check Out Model Branch 433
  - Check Out Package 431
  - Client Configuration 394
  - Compatible Products 395
  - Configuration 413, 415, 438, 439
  - Configuration, Team Deployment 389
  - Configuration, Use Previously-Defined 414
  - Configure Package 424
  - Considerations 383
  - Controlled Packages 484
  - Copy-Modify-Merge Policy 387
  - Create CVS Environment 404
  - Create SCC Environment 411
  - Create Subversion Environment 397
  - Create TFS Environment 408
  - CVS Options 419
  - CVS, Create Local Working Copy 405
  - CVS, TortoiseCVS 407
  - Discussion Of File Control 386
  - Export Model Branch 437
  - Facilities 383
  - File History 427
  - File Properties 427
  - Import Model Branch 438
  - Import Model Branch File 439
  - In Team Deployment 389
  - Include Other Users' Packages 436
  - Introduction 383
  - Locking - Necessary? 387
  - Lock-Modify-Unlock Policy 387
  - Manual, With XML 494
  - Menu (Package) 427
  - Of Nested Packages 386
  - Of Packages 386
  - Offline 392
  - Package Configuration, Validate 443
  - Policies 387
  - Prepare Subversion Environment Under Wine 402
  - Private Model 385
  - Product Configuration Requirements 395
  - Product Setup 394
  - Project Browser Indicators 391
  - Recommendations 422
  - Refresh View Of Shared Model 308
  - Resynchronize Package Version Control Status 444
  - Review Version Control History (SCC Example) 442
  - Review Version Control History (SCC) 441
  - Review Version Control History (TFS, CVS, Subversion) 440
  - Revision, Check Out 440
  - SCC Options 417
  - SCC, Providers Dialog 417
  - SCC, Upgrade For Enterprise Architect 4.5 412
  - Server Configuration 394
  - Set Up 413
  - Settings 415
  - Shared Model 385
  - Shared Packages 385
  - Standard Packages 385
  - Submenu (Project Menu) 91
  - Subversion Options 420
  - Subversion, Create Local Working Copy 399
  - Subversion, Create New Repository Subtree 398
  - Subversion, TortoiseSVN 403
  - Subversion, Using With Enterprise Architect Under WINE Crossover 401
  - SVN Options 420
  - System Requirements 395
  - Tasks 422

## Version Control

- TFS Options 421
- TFS, Create Local Working Copy 409
- TFS, Exclusive Checkout 411
- Undo Package Check Out 432
- Update All Packages To Latest Revision 435
- Update Package To Latest Revision 435
- Usage Scenarios 385
- Use Nested Version Control Packages 390
- Using 422
- Verify CVS Workspace 406
- Verify SVN Workspace 400
- Who Has Checked Out A Package? 427

## Version Controlled Package

- Icon 670

## Vertical Coordinate System

- ArcGIS, Set 1958

## VHDL

- Code Generation 2277
- Diagram Toolbox Pages 2106
- Language Options 2277
- Modeling Conventions 2106

## View

- Add To UML Model, Quick Start 49
- Audit 451
- Data Modeling, What Is A? 2371
- Diagram List 684
- Diagram, Switch Between 791
- Float Views 790
- Locks 332
- Menu 83
- Model, Add To MDG Technology 1558
- Next Diagram 869
- Of Model 686
- Package Browser 673
- Personal Tasks 554
- Previous Diagram 869
- Project Statistics 87
- Project Status 525
- Requirements 1783
- Submenus 84
- Tabs, Close 790

## View Elements Of Array

- Debugger 2238

## View Header

- Model Search 708

## View Options

- Diagram List 684
- Diagram View 784
- Gantt Chart 784
- Package Browser 673

## View Tab Switcher 791

## Views

- Add 770
- Class 769
- Component 769
- Delete 771
- Deployment 769
- Dynamic 769
- Manage 769, 770
- Rename 771
- Simple 769
- Use Case 769

## Virtual Document

- Add Packages As Attributes 2675
- Bookmarking In 2669
- Change Package Sequence 2677
- Create Master Document For 2672
- Create Model Document For 2673
- Delete Package Attributes 2676
- Delete Packages 2676
- Document Order 2677
- Generate Document Report From 2679
- Introduction 2669
- Master Document Element 2669
- Model Document Element 2669
- Model Search Sequence 2677
- Move Package Attributes Between Elements 2677
- Sequence of Model Document Elements 2677
- Sequence of Package Attributes 2677
- Tagged Values For Master Document 2672

## Virtual Document (HTML) 2759

## Virtual Machine 2191

## Visibility Indicators 830

- Values 2080

## Visibility Of Elements

- Customize 922

## Visible Class Members

- Set Diagram Appearance Options 831

## Visio

- MDG Link For, Enterprise Architect 1481

## Vista

- Permissions For Enterprise Architect 34

## Visual Basic

- Code Generation 2277
- Connect To Automation Interface 2805
- Debugging, General Setup 2189
- Import, Reverse Engineering 2139
- Language Options 2277
- Modeling Conventions 2109
- Set Up In Automation Interface 2808

## Visual Basic.Net

- Import, Reverse Engineering 2139

## Visual Check

Diagram Against Baseline 466

## Visual Execution Analyzer

Access 2527

Advanced Debug Techniques, Java 2194

Availability 2527

Break On Variable Changing Value 2229

Breakpoint Management 2224

Breakpoints And Markers Window, Record Sequence Diagrams 2224, 2540

Build Script, Create 2180

Control Recording, Record Sequence Diagrams 2544

Create Sequence Diagram, Call Stack 2241

Debug .NET 2200

Debug .NET CLR Versions 2201

Debug .NET With COM Interop Process 2202

Debug Another Process 2243

Debug Apache Tomcat Server Configuration 2199

Debug Apache Tomcat Windows Service 2199

Debug ASP .NET 2203

Debug Java 2191

Debug Java Applets In Internet Browsers 2194

Debug Java Web Servers 2195

Debug JBOSS Server Configuration 2198

Debug Symbols, C++ And Native Applications 2191

Debugger Frameworks 2222

Debugger System Requirements 2186

Debugger, Overview 2222

Debugging Actions 2231

Deploy Script, Create New 2218

Diagram Features, Generate Sequence Diagrams 2535

Difference Between Recording Marker And Breakpoint 2538

File Search, Use 2164

For C++ Applications 2189

For Microsoft Native Applications 2189

For WINE Applications 2188

General Debug Setup, Java 2191

Generate Sequence Diagram 2549

Inspect Process Memory 2242

Introduction 2527

Java Debug Session, Attach To VM 2194

Map State Changes 2553

Marker Management 2224

Marker Types, Record Sequence Diagrams 2538

MDDE Basic Setup 2175

MDDE External Tools 2175

MDDE, Analyzer Scripts, Manage 2175

MDDE, Build Scripts 2180

MDDE, Generate Code 2173

MDDE, Script Actions, Define 2179

MDDE, Synchronize Code 2173

Model Simulation 2463

Object Workbench, Create Variables 2568

Object Workbench, Introduction 2567

Object Workbench, Invoke Methods 2570

Object Workbench, Overview 2567

Outputs 2527

Overview 2527

Place Markers, Recording Sequence Diagrams 2536

Profiler Overview 2555

Record Activity For Class 2542

Record Activity For Method 2538

Record Sequence Diagrams, Introduction 2531

Record Sequence Diagrams, Overview 2532

Record Sequence Diagrams, Recording Options 2219

Record Sequence Diagrams, Set Up 2535

Record State Changes 2553

Record State Transitions 2550

Record Unit Test Results 2575

Recording History 2533

Recording Markers, Activate, Record Sequence Diagrams 2224

Recording Markers, Disable, Record Sequence Diagrams 2224

Recursive Builds 2180

Run Script, Create New 2217

Run Unit Test 2575

Sample Model, Generate 2529

Save Recording History 2549

Script Search 2164

Search Window 2164

Sequence Diagrams, Limit Auto Recording 2538

Set Code Breakpoint 2226

Set Data Breakpoint 2229

Set Up Debug Session 2185

Set Up Debug Session For .NET 2200

Set Up To Capture State Changes 2551

Show Loaded Modules 2243

Show Output 2231

Start Debugger 2231

State Machine 2551

State Transitions 2551

Step Into Function Calls 2231

Step Out Of Functions 2231

Step Over Lines Of Code 2231

Step Though Function Calls 2548

- Visual Execution Analyzer
    - Stop Debugger 2231
    - Tooltips In Code Editor 2236
    - UAC-Enabled Operating Systems 2187
    - Unit Test Script, Create 2182
    - Unit Test Script, Introduction 2182
    - Unit Test Script, Set Up 2573
    - Unit Test, Record Results 2575
    - Unit Testing, Introduction 2573
    - Uses Of 2527
    - View Call Stack 2239
    - View Local Variables 2234
    - View Local Variables, Long Values 2235
    - View Variables In Other Scopes 2237
    - Work With Marker Sets, Record Sequence Diagrams 2541
    - Workspace Layouts 2174
  - Visual Execution Profiler
    - Attach To Process 2564
    - Function Line Report 2564
    - Getting Started 2558
    - Launch 2564
    - Operation 2557
    - Overview 2555
    - Prerequisites 2556
    - Report, Example 2555
    - Report, Load 2560
    - Report, Save 2560
    - Report, Save As Resource In Team Review 2566
    - Save Profile Report - Document Artifact 2560
    - Save Profile Report - Team Review Resource 2560
    - Set Options 2563
    - Set Sample Intervals 2563
    - Start 2564
    - Stop 2564
    - Supported Platforms 2556
    - System Requirements 2556
    - Team Review, Save Report As Resource 2566
    - Toolbar 2558
  - Visual Execution Sampler
    - Attach To Process 2564
    - Function Line Report 2564
    - Getting Started 2558
    - Launch 2564
    - Operation 2557
    - Overview 2555
    - Prerequisites 2556
    - Report, Example 2555
    - Report, Load 2560
    - Report, Save 2560
  - Report, Save As Resource In Team Review 2566
  - Save Profile Report - Document Artifact 2560
  - Save Profile Report - Team Review Resource 2560
  - Set Options 2563
  - Set Sample Intervals 2563
  - Start 2564
  - Stop 2564
  - Supported Platforms 2556
  - System Requirements 2556
  - Team Review, Save Report As Resource 2566
  - Toolbar 2558
  - Visual Representation
    - Of Elements, Introduction 953
  - Visual Studio 2005/2008
    - MDG Integration For, Enterprise Architect 1481
  - Visual Studio.NET
    - MDG Link For, Enterprise Architect 1481
  - Visual Style
    - Microsoft Office Styles 168
    - Microsoft Visual Basic Styles 168
    - Options 168
    - Select 168
  - Visual Styles
    - Submenu 84
    - Themes 84
    - View Menu Option 83
  - Visualize Systems
    - Enterprise Architect 9
  - VM
    - Attach To In Java Debug Session 2194
- W -**
- W3C XSD
    - Elements and Connectors 818
  - W3CWSDL
    - Elements 817
  - WAN Optimization
    - Enterprise Architect Performance 218, 225, 233, 239, 248, 255, 257
    - For ASA Data Repository 239
    - For MySQL Data Repository 225
    - For Oracle Data Repository 255, 257
    - For PostgreSQL Data Repository 233
    - For Progress OpenEdge Data Repository 248
    - For SQL Server Data Repository 218
  - WAN Optimizer
    - Introduction 259
    - Performance Of Enterprise Architect 259
    - Transmission Diagram 259

- Watched Items
  - Debugger 2237
- Watches Window 2237
  - Break On Variable Changing Value 2229
- Watermark
  - Diagrams 622
- Web
  - Connect To Project Via 266
  - Connecting To Model Via 262
  - Documentation 2638, 2744
  - Page Modeling 1995
  - Report 2638, 2744
  - Report Style Templates 2747
  - Stereotypes 1995
  - Templates, Reporting 2747
- Web Browser
  - Home Website, Access 170
  - Internet Email, Access 170
  - Web Search Engine, Access 170
- Web Ontology Language 1971
  - Definition Diagram, Example 1979
  - Facts Diagram, Example 1979
- Web Page
  - Prototype 1991
- Web Report
  - Create 2744
  - Locate Page By GUID 2744
  - Publish As HTML Dialog 2744
  - Quick Start, Generate Report 2744
  - View 2744
- Web Server
  - Java, Debug 2195
- Web Service
  - Create In BPEL 1.1 1890
  - Submenu (Tools Menu) 108
- Web Service Definition Language 2423
- Web Service Operation
  - BPEL 2.0, Create 1914
- Web Services (WSDL) 2423
  - Generate WSDL File 2445
  - Import WSDL Files 2447
  - Model WSDL, Binding Element 2437
  - Model WSDL, Document Element 2443
  - Model WSDL, Message Element 2430
  - Model WSDL, Message Part Attribute 2431
  - Model WSDL, Namespace 2429
  - Model WSDL, Port Type Operation 2434
  - Model WSDL, PortType 2433
  - Model WSDL, Service Element 2441
  - Model, Create 2426
- Web Services Business Process Execution Language (WS-BPEL) 1870
- Web Site
  - Access Any 170
  - Access Home 170
  - Home, Define Default 605
  - Sparx Systems 38
- Web Template
  - Fragments, HTML 2749
- Webinar Registration 68
- Website
  - SOAML Document Page 2449
- Well Formedness Validation
  - Attribute 2599
  - Diagram 2599
  - Element 2599
  - Feature 2599
  - Relationship 2599
- What Is
  - A Connector? 1389
  - A Foreign Key? 2358
  - A Model? 753
  - A Pattern? 1464
  - A Primary Key? 2355
  - A Project? 200
  - A Requirement? 1726
  - A Stored Procedure? 2364
  - A Tagged Value 1134
  - Enterprise Architect? 5
  - User Security? 316
  - XMI? 473
- WHERE Parameter
  - OSLC Query Capability 275
  - OSLC, Combine With SELECT 277
- Whiteboard
  - Diagram Mode 825
  - Diagram Mode, API 1573
- Whitespace
  - In Package Names 2120
- Wide Area Network (WAN) Optimizer
  - Introduction 259
  - Performance Of Enterprise Architect 259
  - Transmission Diagram 259
- Win32
  - Export Dialog To Resource Script 2141
  - Import Dialogs From Resource Script 2141
- Win32 Dialog Simulation 2520
- Win32® User Interface
  - Combo Box Resizing 1996
  - MDG Technology 1996
- Window
  - Autohide Docked 132
  - Autohide Floating (Dock First) 132
  - Breakpoints And Markers 2224



## Window

- Breakpoints And Markers, Record Sequence
- Diagrams 2540
- Bring To Top 121
- Broad Function 168
- Call Stack 2236, 2239
- Close All 121
- Close All Except Current 121
- Close Current 121
- Combine In Frame 128
- Connections 742
- Constraints 992
- Debug, Step Over Lines 2231
- Diagram Filter 720
- Dock 128
- Element Browser 989
- Float View Tab As 121
- Floating 128
- Generalized Functions 168
- Hide 121
- Layout Chains 894
- Layout Tools 874
- Learning Center 74
- Links 742
- Locals 2234
- Locals, View Long Values 2235
- Maintenance 2623
- Make Active 121
- Memory Viewer 2242
- Menu 121
- Model Views 686
- Modules 2243
- Notes 1142
- Output 169
- Output, DebuggerCode 2231
- Pan And Zoom 698
- Project Management 510
- Project Task Allocation 538
- Properties (Element) 992
- Record & Analyze 2533
- Relationships 742
- Reload 121
- Remove From Frame 128
- Requirements 992
- Resources 1173
- Reveal Autohidden 132
- Scenarios 992
- Scenarios & Requirements 992
- Scripting 2791
- Search, Debugging 2164
- Simulation Events 2504
- System Output 168, 169

- Tagged Value 1134
- Team Review 343
- Testpoints 2578
- Traceability 725
- Watches 2237
- Watches, Break On Variable Changing Value 2229
- Web Browser 168

## Windows

- Authentication 323
- Dockable 125
- Service, Apache Tomcat 2199
- Standard 125

## Windows 7

- Use Debugger 2187

## Windows Active Directory

- Import User Login ID From 324

## Windows Authentication

- Accept 324

## Windows Vista

- Permissions For Enterprise Architect 34
- Use Debugger 2187

## WINE

- Debugging 2188
- DIB Data Access Violation 2188
- Prepare Subversion Environment, Version Control 402

## WINE-Crossover

- Using Subversion With Enterprise Architect Under 401

## Wizard

- Interface Customization 68

## Word Substitution

- Document Report Generation 2665

## Work Flow

- Monitor With Model Views 593

## Work With

- Attributes, Automation Interface Code Example 3006
- Enterprise Architect 7
- Methods, Automation Interface Code Example 3007

## Workbench

- Analyzer Menu Option 112
- Java 2216
- Microsoft .NET 2216
- Setup 2216

## Workbench Variables

- Constructors 2568
- Create 2568

## Workflow

- Data Structures 368

- Workflow
    - Monitor 560
    - Objects 368
    - Script Functions 368
    - Script, Import 2792
    - Scripts 2794
    - Scripts, Execute 560
    - Scripts, Introduction 367
    - Searches 372
    - Status 372
    - Tab, Personal Information Window 560
  - Workflow Script
    - Search Functions With User Tasks 370
    - System-Filled Data Structures 371
    - User Filled Data Structures 372
    - User Input Validate And Control Functions 369
    - User-Called Functions 373
  - Working
    - With MDG Technologies 1476
    - With UML Connectors 1102
  - Working Set
    - Apply 561
    - Artifact 1358
    - Artifact Element 561
    - Copy 561
    - Create 561
    - Delete 561
    - Edit 561
    - Global 561
    - History 561
    - Include Workspace Layout 561
    - Open 561
    - Overview 561
    - Share 561
    - Tab, Personal Information Window 561
    - Track 561
  - Workspace
    - Maintenance 2623
    - Status Bar 150
    - Toolbars, Docked 134
    - Toolbars, Introduction 134
  - Workspace Layout
    - Add To MDG Technology 1557
    - Apply 149
    - Change 163
    - Copy 163
    - Create 149
    - Delete 163
    - For Execution Analysis 2174
    - Manage 163
    - Save 149
    - Select 149, 163, 167
  - Toolbar 149
  - Workspace Layouts
    - View Menu Option 83
  - WriteVariable Action
    - Simulation Variable Action 1273
  - WS-BPEL 1870
  - WSDL
    - Binding Diagram 2437
    - Binding Element 2437
    - Development 2423
    - Document Element 2443
    - Element Types 2423
    - Elements 817
    - Example Model Hierarchy 2423
    - File Source 2443
    - Generate File 2445
    - Import Files 2447
    - Message Diagram 2430
    - Message Element 2430
    - Model Structure 2423
    - Model, Create 2426
    - Namespace Element 2429
    - Package Diagram, Template 2426
    - Package Hierarchy, Template 2426
    - Port Type Operation 2434
    - PortType Diagram 2433
    - PortType Element 2433
    - Service Diagram 2441
    - Service Element 2441
    - Split Files 2447
    - Template 2426
    - Toolbox Page 817
    - Toolbox Pages 2426
    - Transformation 2043
  - WSDL Support
    - Introduction 2423
- X -**
- XMI
    - Export 473, 475
    - Export MOF Model To 2460
    - Export To 476
    - Import 473, 478
    - Import And Auditing 456
    - Limitations 482
    - Manual Version Control 494
    - Report Deletion Of Cross Package References 496
    - Specifications 473
    - UML DTD 483
    - XMI 1.1 475

- XMI
  - XMI 2.1 475
- XMI Import
  - Impact On Auto Increment Columns 598
- XMIType Enum
  - Automation Interface 2825
- XML
  - BPMN 2.0 1869
  - Code Page 639
  - Cross Package Reference Deletions, Report 639
  - Default XMI Version 639
  - Documents, Default Editor 639
  - Export To 476
  - External Reference Placeholders 639
  - Import ArcGIS Workspace 1968
  - Import Referenced Schema 2420
  - Package 484
  - Pattern File 1464
  - Set Default XML Directory 639
  - Set Default XML Editor 639
  - Specifications, Options Dialog 639
  - Structure Tree In Source Code Viewer 2146
  - XMI 1.0 Prefix 639
- XML Any Attribute
  - Toolbox Icon 2407
- XML Any Element
  - Toolbox Icon 2405
- XML Attribute
  - Global, Toolbox Icon 2396
  - Local, Toolbox Icon 2398
- XML Attribute Group
  - Element, Toolbox Icon 2399
- XML Complex Type
  - Toolbox Icon 2401
- XML Element
  - Toolbox Icon (Global) 2392
  - Toolbox Icon (Local) 2394
- XML Engineering
  - Service Oriented Architecture 2386
  - Service Oriented Architecture Modeling Language 2386
  - Service-Oriented Modeling Framework 2386
  - SOA 2386
  - SoaML 2386
  - SOMF 2386
  - XML Schema 2386
  - XSD 2386
- XML Enumeration Element
  - Toolbox Icon 2412
- XML Global Attribute
  - Element, Toolbox Icon 2396
- XML Global Element
  - Toolbox Icon 2392
- XML Group
  - Toolbox Icon 2404
- XML Local Attribute
  - Toolbox Icon 2398
- XML Local Element
  - Toolbox Icon 2394
- XML Model Group Element
  - Toolbox Icon 2410
- XML Schema
  - Any Attribute 2407
  - Any Element 2405
  - Attribute Group Element 2399
  - Complex Type Element 2401
  - Default Mappings From UML 2415
  - Element 2390
  - Elements and Connectors 818
  - Enumeration Element 2412
  - Example Diagram 2389
  - Generate From Abstract Class Model 2413
  - Generate In Garden Of Eden Style 2418
  - Global Attribute Element 2396
  - Global Element 2392
  - Group Element 2404
  - Local Attribute 2398
  - Local Element 2394
  - Model Group Element 2410
  - Schema Package 2390
  - Simple Type Element 2402
  - Submenu (Tools Menu) 108
  - Toolbox Icon 2390
  - Toolbox Pages 818
  - Union Element 2408
  - XSD 2387
- XML Simple Type
  - Toolbox Icon 2402
- XML Union Element
  - Member Elements 2408
  - Toolbox Icon 2408
- XSD
  - Any Attribute 2407
  - Any Element 2405
  - Attribute Element, Global 2396
  - Attribute Group Element 2399
  - Attribute, Any 2407
  - Attribute, Local 2398
  - Complex Type Element 2401
  - Default Mappings From UML 2415
  - Element, Any 2405
  - Element, Attribute Group 2399
  - Element, Complex Type 2401

## XSD

- Element, Enumeration 2412
- Element, Global 2392
- Element, Group 2404
- Element, Local 2394
- Element, Model Group 2410
- Element, Simple Type 2402
- Element, Union 2408
- Enumeration Element 2412
- Example Diagram 2389
- Generate 2417
- Generate From Abstract Class Model 2413
- Global Attribute Element 2396
- Global Element 2392
- Group Element 2404
- Import 2420
- Import, Global Element Behaviour 2421
- Local Attribute 2398
- Local Element 2394
- Model 2387
- Model Group Element 2410
- Schema Package 2390
- Simple Type Element 2402
- Transformation 2044
- Union Element 2408
- XML Schema 2387

## - Z -

## Zachman

- Profile 854

## Zachman Framework

- MDG Technology For, Enterprise Architect 1481

## Zoom

- Control On Status Bar 150
- Diagram View 868
- Diagram, Diagram Toolbar 138
- Submenu 868

## Zooming

- Code Editor, Common 2157

## Z-Order

- Diagram Structure 778
- Element 844
- Modify, Of Elements 778
- Submenu 953



# Enterprise Architect User Guide

[www.sparxsystems.com](http://www.sparxsystems.com)