ENTERPRISE ARCHITECT

User Guide Series

Model Repositories

Author: Sparx Systems

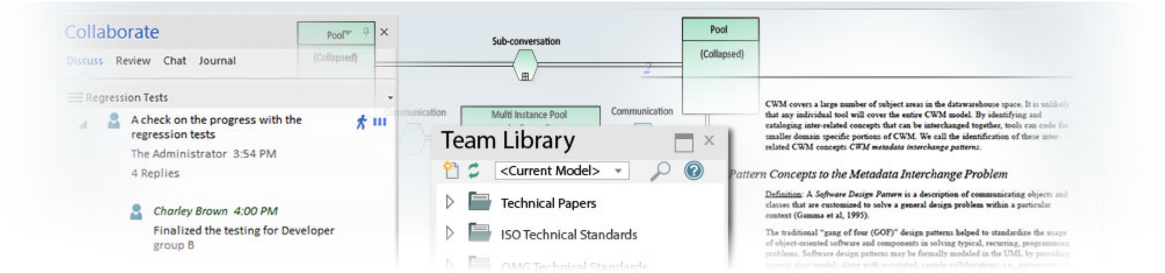
Date: 2021-09-02

Version: 15.2

CREATED WITH ENTERPRISE ARCHITECT

Table of Contents

Model Repository



A check on the progress with the regression tests
The Administrator 3:54 PM
4 Replies

Charley Brown 4:00 PM
Finalized the testing for Developer group B

CWM covers a large number of subject areas in the datawarehouse space. It is unlikely that any individual tool will cover the entire CWM model. By identifying and cataloging inter-related concepts that can be interchanged together, tools can code for smaller domain specific portions of CWM. We call the identification of these inter-related CWM concepts *CWM metadata interchange patterns*.

Pattern Concepts to the Metadata Interchange Problem

Definition: A *Software Design Pattern* is a description of communicating objects and classes that are customized to solve a general design problem within a particular context (Gamma et al, 1995).

The traditional "gang of four (GOF)" design patterns helped to standardize the usage of object-oriented software and components in solving typical, recurring, programming problems. Software design patterns may be formally modeled in the UML by providing

Enterprise Architect is a rich modeling platform that can be used in a wide range of modeling and design situations, from a single user accessing a file based repository to a large and distributed team of thousands of modelers working all over the world using a **Cloud Based Repository**. The repositories can be scaled up and down, and a project that started with a single user can be seamlessly scaled up to a large multi-user repository.

Enterprise Architect has been built to support inter-disciplinary modelers collaborating to create a single and unified view of a system or enterprise, and the platform has numerous built in features to support this team development. By taking advantage of the base UML modeling language, in addition to the numerous extension languages such as BPMN and SysML, plus highly effective frameworks such as TOGAF and UAF/UPDM, modelers, designers, managers and others can build comprehensive, inter-connected models that fully expresses the capabilities, structure and behavior of fully functioning, multi-layered, complex systems and/or enterprises.

In addition, there are features that allow modelers to discuss the model, such as the **Collaborate window** - including sending **Model Mail** - and to review work that has been in progress in the **Library window**. Model Security can be implemented, which facilitates collaboration and prevents work accidentally being overridden.

There is a range of project management features, such as tasks and resource allocation, a Gantt Chart to view progress on tasks over time, and a Project Calendar that keeps track of important project events.

Integration is so easy, and the feature sets so powerful, that strategic thinkers, business managers, architects, information and data modelers, testers, software engineers and many others will derive great benefit from using the tool and incorporating their models into the overall vision.

Key Concepts

Repository Format

Enterprise Architect models are stored in **standard relational databases**. This approach has been built into Enterprise Architect from the very first version. Using a relational database provides a huge advantage over other tools that are limited in scope and capability by relying on a simple text file based model storage system. With support for two file based repository formats (MS JET and Firebird) plus a number of industry standard DBMS servers (such as SQL Server, MySQL and Oracle) plus Cloud based storage, Enterprise Architect provides fast, scalable, flexible and transparent access to models in a wide range of deployment scenarios.

Modeling Languages

Enterprise Architect is based on industry standard UML, a specification maintained by the Object Management Group (OMG). Although initially conceived as a modeling language primarily for software development, over the years the capabilities and formal (standard) extensions have allowed the UML to develop into a rich set of constructs and modeling behaviors that provide very rich and detailed support for everything from software, to business processes, embedded software, enterprise architecture, organizational charts, strategic model, mind mapping, requirements management and much more. The success of UML has been proven over and over again in the real world and the number of new technologies and modeling profiles that use UML as the underlying 'meta-language' are tribute to the power and effectiveness of UML.

Team Development

From the very first version of Enterprise Architect the goal has been to provide the best and most accessible support for Team based development. A wealth of tools and capabilities have been designed into the platform over the years to support security, scalability, information sharing, concurrent access, management, reporting and querying. With the growth of the internet and distributed development scenarios, Enterprise Architect has evolved to be the pre-eminent platform for developing models and designs in a globally distributed situation. The Cloud Server based deployment offers high availability, excellent performance and simple configuration to rapidly bring diverse and distributed teams into a single, living repository where work can be easily shared, reviewed, discussed and managed.

Scalability

By leveraging the power of today's DBMS servers and Cloud Based technologies, Enterprise Architect can be easily scaled up to truly epic proportions. Millions of objects can be stored and accessed by large and dispersed teams in real time. The level of scalability is limited solely by the choice of DBMS or Cloud Technology and the capabilities of the network and server infrastructure the model is deployed on. In practice, over the many years that Enterprise Architect has been used in industry, modelers and designers have taken advantage of the powerful in-built scalability to construct stunning and often huge models that capture an enormous amount of information about complex and highly inter-connected systems and enterprises.

Security

Due to the nature of the Repository used by Enterprise Architect and the possible deployment scenarios, security can be managed in a number of ways. In the Corporate, Unified and Ultimate Editions of Enterprise Architect, there is built-in support for defining users and groups that permit locking and management of elements and Packages within the model. This will be discussed later in the section on Security. In addition, it is also possible to restrict general access to models by using file-based security access for the MS JET and Firebird-based models (.EAP and FDB) and DBMS server authentication and access for the larger database server-based systems. As a further refinement it is also possible to restrict access to a model using the HTTPS protocol when connecting to a Cloud Based server.

Versioning

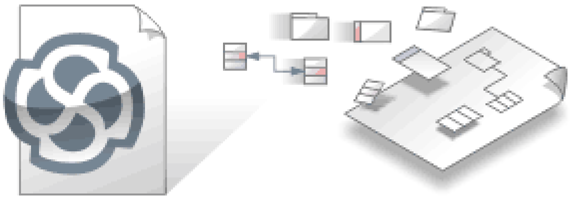
Enterprise Architect uses industry standard XMI (XML Interchange) format files to store and version model Packages. By integrating the import and export of XMI model fragments into a **Version Control** system (such as SVN) supported by Enterprise Architect, you can maintain multiple versions of the same model in different repositories and exercise fine control over the development and design process. Given the nature of this versioning system, there are several different scenarios that can be used to maintain different models and how teams access information. For example, where the versioning system is used as the single point of truth and de-facto Repository, it could be convenient to use many workgroup or single user based Repositories that reflect the actual model at a particular point in time. In contrast, where the versioned files are mainly held for backup and baseline purposes only, a larger, Team based DBMS server would be used to provide the model sharing and collaboration capabilities. So whatever the actual needs are, Enterprise Architect has a number of solution scenarios that can be adapted to provide the right level of change management and collaborative development.

Example Repository Tasks

Task

Create a project in a simple, file-based repository (.EAP and .FEAP files)
Create the project in one of a range of DBMS repositories - these provide for larger models with more concurrently connected users (Corporate and extended editions)
Manage change within the model
Maintain the integrity of the project data
Share the reference data used across the project, between models and between projects
Allow each user to configure their personal preferences for how project tools display and behave on their workstation
Document your model through document or web reports, which you can tailor to your purposes

Repository Overview



Enterprise Architect repositories contain valuable information about an enterprise and the projects that it runs. The information could be at a strategic, tactical or operational level, and the models can be based on a range of disciplines from business through technology to engineering. The models can represent the current or future state of the enterprise or one of its business or technology divisions or services. The repository will become the central hub of corporate knowledge and, through integration with a range of other tools, can provide a unified view of all enterprise knowledge and its connections.

As a modeler you create content in the form of graphs of interconnected elements that are visualized on diagrams, lists and specialized representations such as Matrices, Kanban Boards, Charts, Graphs, Road Maps and more. Stakeholders outside the modeling groups are able to view these visualizations and collaborate through discussions, reviews and chat, giving feedback and asking questions and so participating in the evolution of the models.

An Enterprise Architect Repository is stored in a relational database; this can be a file based database such as MS Access or Firebird, or a database management system (DBMS) such as Oracle or MS SQL Server. Enterprise Architect can connect to DBMS-based repositories via a direct connection or with the use of the Pro Cloud Server - of the type usually referred to as Cloud Based Repositories. It is common to start with a **File Based Repository** and, when the project gets some momentum and there are a number of modelers accessing the repository, to transfer it to a DBMS. Whichever type of repository is implemented, the user interface remains the same and users can seamlessly move from one type of repository to another and continue to contribute as a team member on this powerful modeling platform.

Concepts

Concept	Description
Repository as a File	In any version of Enterprise Architect you can 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, 2003 or 2013, or any other reporting tool that can work with JET databases • A .feap file is a Firebird project file
Repository in a DBMS (Database Management System)	In the Enterprise Architect Corporate, Unified and Ultimate Editions you can also use a suitable DBMS repository for projects. DBMS projects have the same logical structure as .eap and .feap files, but provide much greater scalability and concurrent access. You connect to a DBMS using ADO/ODBC (see *Connect to a Data Repository*). Note that when configuring an ODBC data source on a 64 bit machine, you should access the ODBC data manager using the 'Configure > User Tools > ODBC Data Sources' ribbon option, as this will load the correct ODBC manager to access 32 bit data sources as used by Enterprise Architect.
Repository in the Cloud	In the Corporate, Unified and Ultimate Editions of Enterprise Architect you can further enable the distributed capabilities of Enterprise Architect and use the Internet or local network to connect to a Pro Cloud Server on which one or more Repositories have been installed and configured for access. The Pro Cloud Server is free software available from Sparx Systems, and is relatively simple to set up and

	configure on a suitable Windows-based machine. Firebird files or any DBMS server are the required Repository types supported by the Pro Cloud Server, and these can reside on the same machine for best performance, or on another machine to maximize local and network access. Cloud-based repositories provide the ultimate in distributed development and minimal client configuration. It is, in fact, much easier to deploy a Firebird-based repository on a Pro Cloud Server for distributed development than to set up and manage a full DBMS solution.
Models	Enterprise Architect uses this terminology when discussing Repositories and their contents. 1. A Model is a related group of content that expresses some focused aspect of a system or enterprise that is related by the modeling tools and paradigm used. For example, you might build a 'Usecase Model' that describes behavior and scenarios using UML Use Cases. Process models, architectural models, behavioral models, structural models, Class models and so on are all Models within the current Project. 2. A Project is a group of models that has a single unifying purpose. For example, a Project to design the navigation systems for a new aircraft or automobile would be considered a single Project. Within this Project there would be multiple models. Projects typically define levels of security, necessary types of view and model, necessary procedures and other management tasks to deliver the product. Project management capabilities in Enterprise Architect are very rich and allow for resource management, security, change management, testing and other typical project management tasks. 3. A Repository is a single File, DBMS database or Cloud Server address that contains one or more Projects. Typically, file based repositories such as MS JET and Firebird models contain a single Project with multiple models that can be shared by a small workgroup or only accessed by a single modeler. DBMS and Cloud based Repositories will often contain more than one Project, especially where the Projects are in some way inter-related or co-dependent.
Adding Initial Content	After creating and accessing your new project in whatever form the Repository takes, you can now use the **Model Wizard** to create Model Packages that range from generic and simple views to more complex and pattern based content. You can add Models to a project from the **Browser window** by: - Right-clicking on an existing model and selecting the 'New Model' or 'Add a New Model using Wizard' options - Right-clicking on a Package and selecting the 'Add | Add a New Model using Wizard' option - Clicking on an existing model, pressing the **Insert key** and selecting the 'New Model' or 'Add a New Model using Wizard' menu options - Clicking on a Package, pressing the Insert key and selecting the 'Add a New Model using Wizard' menu option
Opening Existing Projects	Existing Projects are accessible via the 'Recent Documents' list under the 'File Management' icon (), the 'Recent' list on the **Start Page**, and through the 'Open Project' option under the 'File Management' icon. Connecting to an existing DBMS or Cloud based Repository is discussed in more detail under the relevant sections on working with those Repository types. For new users, you can use the 'Start > Help > Help > Open the Example Model' ribbon option to open and explore the Enterprise Architect Example Repository supplied with Enterprise Architect. This example is not so much a fully worked example as a collection of models that show how you can leverage the power of the Enterprise Architect platform and the UML to create a wide range of visual and text based models.

DBMS Repositories	You can connect to any of these data repositories: • MS Access 2007 onwards • Firebird • SQL Server 2000 onwards • SQL Server Express 2005 onwards • MySQL • MariaDB • Oracle 9i, 10g, 11g and 12c • PostgreSQL In brief, to create a new database repository, you first create a new database with the DBMS management software, then run supplied scripts to create the logical structure. You then use Enterprise Architect data transfer functions to move a project from a .eap, .feap or DBMS model into the new project. Details are available under the *Server Based Repositories* section.

Brief Introduction

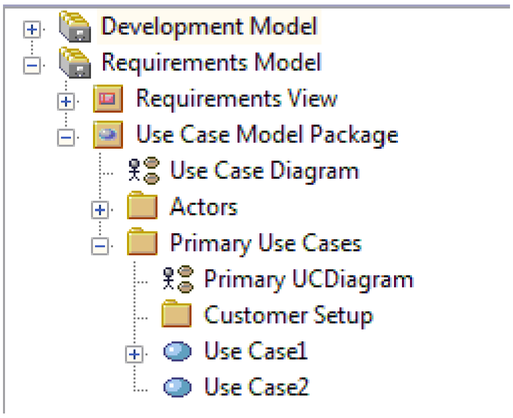
Enterprise Architect is used to model business, technology and engineering systems. You create models for a variety of reasons including specifying the requirements, analyzing the design options, describing the architecture, designing programming classes and databases, documenting an existing system and much more. All the artifacts associated with these models are stored in the repository not as files but in tables in a relational database. These model can all be stored and accessed from a single repository sometimes called a project. The repository can be of two fundamental types:

- FIle based Database - MS Access *.eapx file or Firebird *.feap
- Server Based RDBMS - Oracle, MS SQL Server MySQL, PostgreSQL etc.

Enterprise Architect makes connections to these databases

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, hosted in a larger DBMS system or based in the Cloud.

Concepts

Concept	Detail
Project	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. These components are organized into a hierarchy of Packages, which help to group and manage related components.
Model	Different aspects of the process or system - or their development - are defined by Model Packages, which you generate from templates specifically structured to support the aspects that the Model Packages represent, such as requirements or deployment. 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.
View	The top-level Packages in a model can also be Views, which represent partitions of the model that you define yourself. You can start with standard Views such as Class or Component, or create whatever partitions are appropriate to your model.
Example Project Structure	 Development Model Requirements Model Requirements View Use Case Model Package Use Case Diagram Actors Primary Use Cases Primary UCDiagram Customer Setup Use Case1 Use Case2

	Each View or Model Package contains Packages; the Use Case Model Package contains: • Actors and • Primary Use Cases It also contains the Use Case diagram, which could be an overview of the Package structure or function. Each Package itself can contain one or more diagrams, one or more Packages, and several elements; the Primary Use Cases Package contains the: • Primary UCDiagram • Customer Setup Package • Use Case1 element • Use Case2 element 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. Note that models do not have to be constructed this way. Depending on the methodology you are following and information you need to model, the project structure might look completely different. In addition, many architectural frameworks such as TOGAF and ArchiMate specify their own project structure as part of their overall solution architecture.

Managing Connections to Projects

An Enterprise Architect project is used for storing and managing the components of one or more UML models. The Professional Edition of Enterprise Architect works 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, or a Cloud Based server 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
- Connect to a project via the Cloud
- Create a new Enterprise Architect project file

Access

Ribbon	> Open Project
Keyboard Shortcuts	**Ctrl+O**
Other	**Start Page** | New | Manage projects

Options

Field/Button	Action
Local File	Click on this button to open a file browser dialog, and locate and select an existing .EAP, .EAPX or .FEAP file; the .EAP file can be a project file or a shortcut to a project hosted on a DBMS. Alternatively, click on the **drop-down arrow** and select the 'New Project' option to create a new project.
Connect to Server	Click on this button to specify connection details for opening an Enterprise Architect project that is hosted on a DBMS. The 'Windows 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: • 'Connection Wizard' - the 'Data Link Properties' dialog again displays • 'Connection String' - the 'Connection String' dialog displays; type in or paste the connection string and click on the **OK button** to connect directly to the project Note that you can also display the 'Connection String' dialog directly, by selecting the 'Custom' option on the Start page.
Connect to Cloud	Click on this button to connect to a project through the Cloud. The 'Cloud Connection' dialog displays, on which you specify the server, URL and model name

	to access via the Cloud.
Recent Projects	This panel lists the most recently opened projects, up to a maximum of ten. To open one of these projects either double-click on the name, or click on it and then click on the **Open button**. Each project name is shown with its access path. You can edit the access path by right-clicking on it and selecting the 'Edit Connection String' context menu option; make the changes on the 'Connection String' dialog, and click on the **OK button**. Similarly, if the project has a connection string (not a shortcut), you can change the name of the project by right-clicking on it and selecting the 'Edit Project Name' menu option; type a new name in the 'Edit Project Name' dialog and click on the OK button. The list of recent projects is also displayed: • On the **Start Page**, under the 'Recent' heading • As a drop-down menu under the **Open Project button** on the Default Tools toolbar • With the list of options
Remove Selection from List	Click on a project name and click on this button to remove that project name from the 'Recent' list.
Show this Dialog at Startup	Select the checkbox to show this 'Open Enterprise Architect Project' dialog automatically upon starting Enterprise Architect. Deselect the checkbox to hide the dialog.
Default Project to Open	Click on a project name in the 'Recent' list and select this checkbox to open the selected project automatically whenever you start up Enterprise Architect.
Open	Click on this button to open the currently selected project in the 'Recent Projects' list.
Cancel	Click on this button to close this dialog without any further action.
Help	Click on this button to display this Help topic.

Notes

- Use of a DBMS and/or Cloud for hosting the model repository is available in the Corporate, Unified and Ultimate Editions of Enterprise Architect
- Enterprise Architect supports these DBMS products for hosting model repositories:
  - SQL Server and SQL Server Express;
  - MySQL
  - Oracle 9i, 10g, 11g or 12c
  - Postgre SQL and
  - ASA

Obtain Repository Connection String

A 'connection string' defines all the connection information that Enterprise Architect requires to open a repository.

Connection strings are typically only available from 'Manage Projects'.

1. You can open the 'Manage Projects' dialog by:
   - Pressing **Ctrl+O**, or
   - Clicking on > Open Project... or
   - Selecting **Start Page** | New | Manage projects
2. Find the repository in the 'Recent Projects' list. Note the 'Path' column will show the connection string; however, if it is too long for the screen it will be abbreviated and '...' shown.
3. Right-click on the repository and select the 'Edit Connection String' context menu option. This will show the 'Connection String' dialog.

When the 'Connection String' dialog is first shown the complete text of the connection string will be automatically selected. Press **Ctrl+C** to copy it to the clipboard.

It is usually a good idea to include this connection string in communications with Sparx Systems as it provides useful information about how you connect to the model, which can help Sparx Systems understand your environment.

Project Shortcuts

Enterprise Architect enables you to create a desktop shortcut (or proxy file) to an Enterprise Architect project (Cloud, DBMS or file-based). Each shortcut is a file containing the connection string for the model. In addition, the shortcut also defines views that Enterprise Architect should open when it opens the model, as outlined here. In this manner it is a convenient and customizable way of quickly sharing a connection to a Cloud based model without requiring other users to know the details of the connections string, model name and other parameters.

Access

Ribbon	> Save As Shortcut

Topics

View	Detail
Diagrams	Define one or more specific diagram(s) to be loaded on opening the project.
Model Search	Open the **Model Search** with a specific text string and search type. For searches operating on the current tree selection, a diagram in the target Package must be opened first. If you use a custom SQL search, the SQL must include ea_guid AS CLASSGUID and the object type. You specify a single Model Search to open.
The Relationship Matrix with a saved profile	Open the **Relationship Matrix** with a saved profile. You specify a single Relationship Matrix profile to open.
The default Library	Open the default Library document. You specify the Library once.
Working Set	Open a working set. You specify a single Working Set to open. This is very similar to the shortcut itself, opening a defined set of diagrams and views. However, the working set can also open source code editors, therefore widening the capabilities of the shortcut alone. Working sets make it easy to customize the main views you want to open without having to resave your project shortcuts. The working set is easy to tailor to your changing requirements, adding greater flexibility to any shortcut that calls the working set. 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 same set up each time you use it. The working set always captures the current work environment each time the model

	closes. This option can be very useful in, for example, a coding environment when you want to return to the last files you were editing.
Workspace Layout	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. You specify a single Workspace Layout to open. However, by creating and using separate shortcuts you can open Enterprise Architect in exactly the configuration you need for the work you want to do - modeling, coding, or debugging, for example.
An example shortcut	You might create a shortcut to open, in sequence: • A Development module • The **Model Search** for a simple search on the term Issue • The module Issues diagram • The module 'Changes' diagram 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. 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 'Changes' diagram displayed in the **Diagram View**.

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, 2003 or 2013, 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

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

Ribbon	: Save as Shortcut

Create a project shortcut

Step	Action
1	Open Enterprise Architect.
2	Open the required project.
3	Select the ' |Save as Shortcut' option. The 'Save Project Shortcut' dialog displays.
4	Click on the ... button at the end of the 'Target File' field. The 'Save Project As' dialog displays.
5	Browse for the appropriate file location and, in the 'File name' 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 **Save button** to return to the 'Save Project Shortcut' dialog.
7	Click on the **Add Other button** and select the required option to define: • A diagram to open • A **Relationship Matrix** profile to open • The **Library window** • A **Model Search** to perform • A working set to apply • A workspace layout to apply
8	The appropriate browser or dialog displays to define the view to display; enter the details and click on the **OK button**. The view is added to the 'Actions when model is opened' field; the entry is automatically selected, with a tick in the checkbox.

9	Repeat steps 7 and 8 for as many additional views as you require.
10	Review the items in the 'Actions when model is opened' field. • If you decide not to have an item in the shortcut, deselect its checkbox • If you want to clear all selected items, click on the **Include None button** Unselected entries are deleted when you save the shortcut.
11	If you decide to change the sequence and/or make a different view display first in the **Diagram View**: • Click on the appropriate entry • Click on the 'Up Hand' or 'Down Hand' buttons
12	Click on the **OK 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

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

Ribbon	: Save as Shortcut

Capture your current work environment

Step	Action
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: • You have opened all diagrams you require • If necessary, you have opened the **Library window**, **Model Search** (with appropriate search term and type) and/or **Relationship Matrix** (at the appropriate profile) • The view you want to resume work on is the last one opened
4	Select the 'Save As Shortcut' 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 'Target File' field displays the file location of that shortcut. Otherwise, click on the ... button at the end of the 'Target File' field. The 'Save Project As' dialog displays.
6	Browse for the appropriate file location and, in the 'File name' 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 **Save button** to return to the 'Save Project Shortcut' dialog.
8	In the 'Actions when model is opened' field, click on the **Include All button**.
9	If you also want to save current window positions click 'Add Other' followed by 'Add Workspace Layout'.

	The dialog shown allows you to select an existing layout or save the current layout.
10	Click on the **OK button** to save the shortcut.

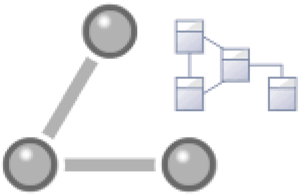
Encrypt Repository Password

If your model is developed on a DBMS repository, the 'Save Project Shortcut' dialog has an 'Encrypt Connection String' checkbox.

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.

File Based Projects



Enterprise Architect provides a convenient light-weight, low-barrier and portable file-based repository out-of-the-box, which is ideal for modelers who want to start work immediately; with the file-based repository you can be modeling within a few minutes. When a number of colleagues want to contribute to the models, the file can be moved to a file-share and it is possible to work with this setup even when the model becomes quite large and the number of users increases to five or more. All editions of Enterprise Architect support this type of repository in the form of a .eap file in an Access database, or a .feap file in a Firebird repository.

You can create your own repository from scratch, but a default repository (EABase.eap) is conveniently supplied, located in the Enterprise Architect installation directory. The file acts as a template for creating new projects and can be copied to give you a head start in creating a repository. You can tailor this repository to suit your organization or create your own base repository.

A **File Based Repository** can be easily transferred to a DBMS based repository using the **Project Transfer** feature, allowing a project to be scaled up when the team increases in size or when organizational policy dictates it.

File-based project tasks

Task	Detail
Create .EAP/.EAPX Project Files (MS Access)	Select one of: • ⊗ ▾ | New Project menu option • 'Local File | New Project' option on the 'Manage Projects' dialog, or • 'Create a new project' option on the **Start Page** All of these options display the 'New Project' dialog; select a directory and enter a file name for your project (.eap under JET 3 or .eapx under JET 4). Once the project has been saved, the **Model Wizard** displays, which makes a selection of Model Patterns available; select the Model Patterns to use. Enterprise Architect adds a model containing the selected Model Packages to the **Browser window**.
Create .FEAP Project Files (Firebird)	As for a .eap/.eapx file, except that in the 'New Project' dialog, in the 'Save as Type' field, click on the **drop-down arrow** and select the .feap file extension. Enterprise Architect will check your input and display an error message if: • Your Firebird repository is not located on the local drive; a Firebird repository is not appropriate for a network shared project • The file path contains characters that are not in the system codepage
Location of default (EABase.eap) repository template	The default installation directories, depending on which version you have installed, are: • Registered Edition: C:\Program Files\Sparx Systems\EA • Trial Edition: C:\Program Files\Sparx Systems\EA Trial • Lite Edition: C:\Program Files\Sparx Systems\EA Lite

	A base project contains templates and reference data from which you can quickly develop your own project. Note that in addition to deriving a new model from EABase.eap. it is also possible to start a new project by copying an existing one. While it is possible to simply 'copy' the .EAP file at the file system level, this results in two identical models and should only be used when that is the required behavior; for example, when distributing a model to a client for verification. If you use the built in functions of Enterprise Architect to create a new model based on an existing one, Enterprise Architect will modify the unique identifiers (GUIDs) for all elements and related constructs so that the new model is essentially unique, and not a simple copy of the existing one.
Configure Project	Having created your project, there are a number of things that you might want to do to set up your project for use by yourself or other team members. You might want to: • Enable Security and set up Users, Groups and Permissions • Set Project Options and Personal Preferences • Set up Technologies, **Perspectives** and Ribbon Sets • Create a Repository structure in the Browser • Use patterns to create initial project content • Set up Report Templates • Set up default database and programing language options
Create Custom Templates	You can customize any Enterprise Architect project as a template project - meaning that you can use it as the basis for creating other project files. This could include 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.

Best Practices for File Based Repositories

Single User	Simple file based repositories (.eap and .feap files) are best suited to single user development with the model file placed on the modelers local drive. This is fast, very efficient and provides a powerful means of getting started in modeling a particular solution. Many models will start out as single user repositories and only later be migrated into a DBMS or Cloud based scenario for larger team access. If you need to quickly start work on a project and do some exploratory work, or you are working alone as a consultant, the .eap or .feap file format is ideal.
Small Workgroup	With all versions of Enterprise Architect it is possible to share a .eap file at the network file level. This is really only a suitable solution for small workgroups (up to, say, five members) who are collaborating on a shared model. This solution is very much at the mercy of the quality of the network connection and could be less than ideal if collaborating users increase in number and/or the model grows very large. As Enterprise Architect on the user's machine is still running all the queries and performing all database operations on a network file, a slow network will result in less than ideal performance. As team size and model size increase it is best to upsize to either a DBMS or Cloud-based architecture.
In Conjunction with Version Control	One additional scenario in which file based repositories can be used to great effect is with a **Version Control** system such as SVN. If the SVN system holds the

	master content, then individual developers/modelers can check out the material of interest to them into a local .eap or .feap file. When they have made their changes they can migrate their work back into the shared master. This style of development is quite effective, and as the import and export of model material from the Version Control system works best with single user local project files, it is a good match and a good use of the two technologies.
Copy of Repository for Storage or Review	A further use of the single file repository is for distribution to clients and others for review and comment, where the original model(s) is located in a DBMS or Cloud that the recipient would not have access to. As it is relatively easy to bulk copy a DBMS model to a local .eap file, this is a good method for passing on a model to a client or another interested party. If some of the material is restricted, it is possible to delete sections from this single file repository after the bulk copy is complete and before passing it on.
Replication	If network access is limited and DBMS and Cloud based solutions are not possible, another (less than optimal but still effective) team based solution is to use the built in replication features of the .EAP file format. By creating a design master and distributing replicas to other modelers, it is possible to effectively share and contribute to a model without having direct shared access to the common model. On the down side, someone will have to perform the model merges as required and if there are replication conflicts (two changes to the same item) this will require manual resolution.

Notes

- You can also add Model Packages to a project by clicking on the **Browser window** header bar and selecting the 'New Model From Pattern' menu option

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. You can also copy an operational .EAP or .EAPX 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

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 provides the option to reset all the unique identifiers for Packages and elements, so that your new project is truly unique - otherwise you create an exact copy of the original project.

Access

Open the template project or existing operational project then;

Ribbon	⊗ ▾: Save Project As

Create a new project from a base project

Option	Action
Target Project	Type the new project filename. You can either type in the file path as well, or click on the **Browse button** and browse for the required file path. If this is to be a shared project, store the file on a shared network resource such as a Network Server or Workgroup Server.
Source Project	Defaults to the path and name of the currently-open template or project.
Reset New Project GUIDs	If the target project is to be a different project from the source (such as when you copy a base template), select this checkbox to replace all GUIDs from the base model with fresh GUIDs in the new model. However, if the new project is to be a direct copy of the source, or based on a project that is under **Version Control**, we recommend that you deselect this checkbox to prevent duplication of Packages when the 'Get Latest' facility is used.
Save As	Click on this button to save and open the new project.

Sharing File Based Projects

Enterprise Architect offers a diverse range of functionality designed specifically for sharing projects in team-based and distributed development environments, through network deployment of model repositories, DBMS based repositories, Cloud based models, replication and XMI Import/Export.

Features

Feature	Detail
Network Deployment	Network deployment is possible under three different schemas: • File based repositories • DBMS server based repositories, or • Cloud based deployment (recommended) DBMS server based repositories offer better: • Response times than files on networks, due to the inherent structure of the DBMS • Solutions when networking problems are encountered, as they have the ability to backtrack transactions caused by external breakdowns
Replication	Replication enables data interchange between .EAP based repositories and is suitable for where many different users work independently in parallel development. Modelers merge their changes into a Design Master only as required; it is recommended that a backup is carried out prior to replication. Replication cannot be performed on repositories stored on a DBMS server.
XMI Import and Export	XMI Import/Export can be used to export and share discrete Packages between developers; XMI enables the export of Packages into XML files that can then be imported into any model. Package control can be used to set up Packages for **Version Control** and to enable batch export of Packages using XMI; 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.

Notes

- DBMS Repository support is available with the Corporate, Unified and Ultimate Editions of Enterprise Architect

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.

Facilities

Facility	Detail
By Sharing a project	• Many people can work on the model at the same time and contribute their particular skill • Team members can always see what the latest changes are, keeping the team informed and up to date with the project status
You can share a project in three ways	• Place the project in a shared network directory • Use replication • Use a shared DBMS-based repository

Notes

- Project Sharing and Replication are available in the Professional, Corporate, Unified and Ultimate Editions of Enterprise Architect
- DBMS repositories are supported in the Corporate, Unified and Ultimate Editions of Enterprise Architect

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 the view

Object	Options
Project	• Right-click on the Package name in the **Browser window** and select the 'Contents | Reload Current Package' option, or • Press **F5** to refresh the Package, or • Select the 🔄 (Reload Project) icon in the Project Tools toolbar, or • Press **Ctrl+Shift+F11**, or • Close the project and reopen it
Diagram	• Select the 'Start > Desktop > Workspaces > Manage Views > Reload View' ribbon option, or • Right-click on the opened diagram tab in the **Diagram View**, and select the 'Reload <diagram name>' option

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

Enterprise Architect accepts a number of concurrent connections without issue; however, there are points you should consider:

Issue	Description
Lock Outs	There can be occasional 'lock-outs' when one user tries to access or update something another user is in the process of modifying.
Other Changes	Changes to the **Browser window** (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.
Diagrams	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.
System crashes	(.EAPX/.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 ('Configure > Model > Integrity > Manage .EAPX/.EAP File > Repair .EAPX/.EAP File') to carry out this task.

Notes

- Firebird-based projects (.feap files) are not suitable for sharing over a network

Distributed Development

Enterprise Architect supports distributed development using two different techniques.

Replication

Using replication, geographically separated analysts can update and modify parts of the model in replicas, then merge these back together at a central location.

XMI Import/Export

Using XML-based import/export, you can export discrete Packages to XML to share among the development team; this has several benefits over replication:

- You can assemble a model from only the parts necessary to get your job done
- You can assemble a full model if required
- You can assemble a model from different Package versions for different purposes (such as customer visible, internal release only)
- You can roll-back parts of a model as required
- There is less chance of 'collisions' between developers if each works on a discrete Package
- The process is controllable using a **Version Control** system, or through Package control

XML-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.

XML-based Import/Export is accessed through the 'Publish > Model Exchange > Export-XML' and 'Import-XML' ribbon options.

Replication

Apart from sharing Enterprise Architect projects in real time over a network, you can also share projects using replication.

Guide

Facility	Detail
Replication	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. In Replication: • A project is converted to a design master, then replicas are made of the master • Users take the replicas away, modify the project, then bring their replicas back to be synchronized with the master file
Use Replication	To use replication: 1. Convert the base project to a design master. 2. Create replicas from the design master. 3. Take the replica away and work on it as required, then bring it back for synchronization with the design master. 4. Synchronize the replicas, during which all changes to both the master and the replica are propagated in both directions, so they both finally contain the same information.
Enterprise Architect Merge Rules	Enterprise Architect follows these rules in merging: • Additions are cumulative; that is, two replicas each creating three new Classes result in six new Classes after merging • 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 Conflicting modifications appear in the 'Resolve Replication Conflicts' dialog.
Upgrades and Replicas	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.
Avoid Change Collisions	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. However, since Enterprise Architect does not enforce this rule, it is possible for users' work to conflict; to minimize difficulties, please note these guidelines: • 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 • If small pieces of information have been lost, they should be typed into one of the merged models after synchronization • If a large piece of information has been lost (for example, an overwritten large Class note) use the 'Resolve Replication Conflicts' dialog

Disable or Remove Replication Features	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.

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

Ribbon	Configure > Model > Integrity > Manage .EAPX/.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 **Browser window**.
3	Select the 'Make Design Master' menu option and follow the on-screen instructions.

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

Ribbon	Configure > Model > Integrity > Manage .EAPX/.EAP File > Create New Replica

Create a replica

Step	Action
1	First create a design master, then select the 'Create New Replica' 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, Unified and Ultimate Editions of Enterprise Architect, if security is enabled you must have 'Manage Replicas' permission to create a replica

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

Ribbon	Configure > Model > Integrity > Manage .EAPX/.EAP File > Synchronize Replicas

Synchronize replicas

Step	Action
1	Open the design master project file (or the first required replica).
2	Select the 'Synchronize Replicas' 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, Unified and Ultimate Editions of Enterprise Architect, if security is enabled you must have 'Manage Replicas' permission to create a replica

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

Ribbon	Configure > Model > Integrity > Manage .EAPX/.EAP File > Remove Replication

Remove replication from your model

Step	Action
1	Open a .EAP file (any .EAP file, other than the one having replication removed) - the menu option is not available if no file is open.
2	Select the 'Remove Replication' menu option. The 'Remove Replication Wizard' dialog displays.
3	Enter the full path and file name of the project to have replication removed. Click on the **Next button**. If the project displays Version Number 7.0 this indicates that **Auditing** is enabled. Auditing should be disabled before proceeding (see *Notes*).
4	Enter the full path and file name of the base Enterprise Architect model (with no replication) to act as template. Click on the **Next button**.
5	Enter the full path and required file name for the output file. Click on the **Next 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 **Run 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:

1. Disable Auditing.
2. If prompted to do so, allow Enterprise Architect to roll back the database version.

3. Remove replication.

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
Open Design Master Project first	After installing the new version of Enterprise Architect, it is very important that you open the design master before opening any of the replicas with the updated version.
Changes to the database design	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.
Propagate Changes	Design changes are propagated through to the replicas the next time the replicas are synchronized with the master.
Alternatively	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.

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.

Access

Ribbon	Configure > Model > Integrity > Manage .EAPX/.EAP File > Resolve Replication Conflicts

Considerations

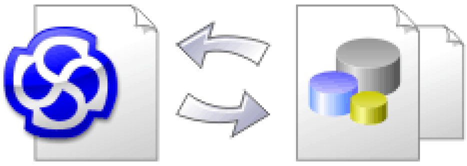
Consideration	Detail
Avoid the problem	Ensure that each team member always works in a separate area of the model within their replica.
Check for conflicts	After synchronizing replicas, open the 'Resolve Conflicts' dialog (see next table) and check if there were any conflicts.
Response to conflicts	When 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. 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. 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.

Resolve conflicts in changes to the same model object in two separate replicas

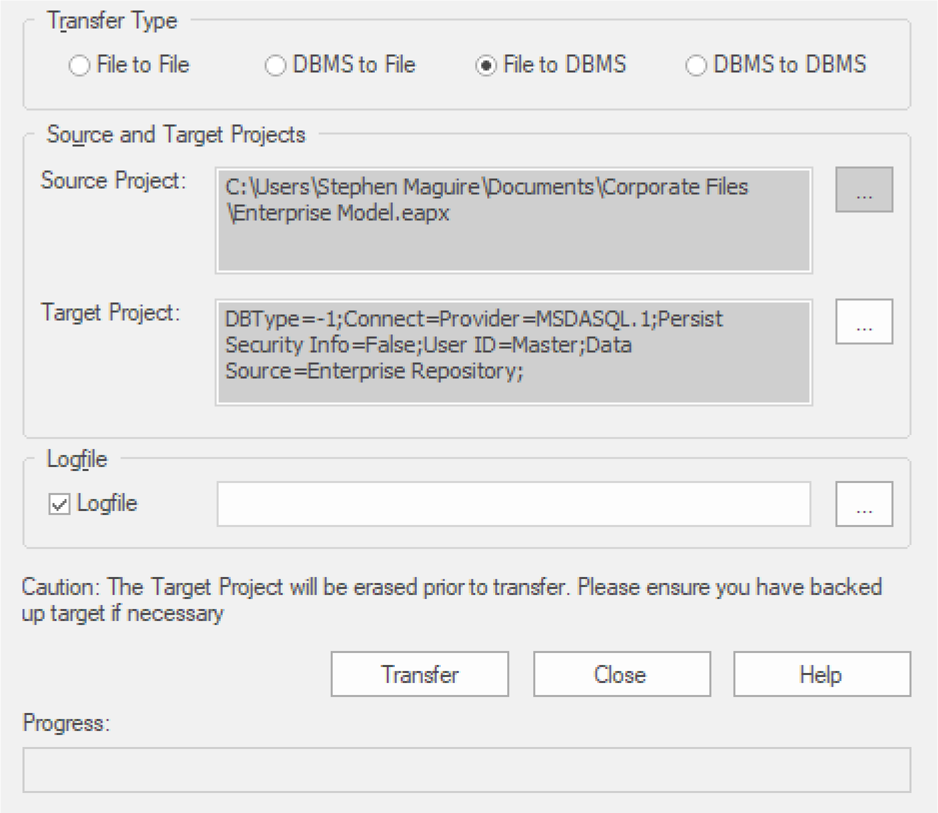
Step	Action
1	Synchronize a replica with the Design Master, and display the 'Resolve Conflicts' dialog.
2	In the 'Table with Conflicts' list, click on the entry that is likely to contain the lost information.

3	Click on each entry in the Conflicting Records list. When the lost information appears in the Conflict Details list, click on the **Overwrite with Conflict button**.
4	In the 'Conflicting Records' list, you can also copy each Row ID to the clipboard by right-clicking on the row and selecting the 'Copy to Clipboard' menu option. You can locate the object that is in conflict in the **Browser window** or diagrams, by right-clicking on it and selecting the 'Find in Project Browser' or 'Find in Diagrams' option.

Server Based Repositories



As an Enterprise Architect repository becomes larger, or the number of concurrent users increases or as organizational policy dictates, it might be more appropriate to use a database management system (DBMS) to store the repository. A **File Based Repository** can be created and used in any Edition of Enterprise Architect; however, if you decide to use a DBMS based repository you will need to use the Corporate, Unified or Ultimate Editions. If you have started modeling with a file based repository Enterprise Architect has a function to transfer your repository from a file based repository to a server based one, helping you to get started quickly. The same user interface is used for both file and server based repositories so the applications will for all intents and purposes appear the same.



Transfer Type

○ File to File ○ DBMS to File ● File to DBMS ○ DBMS to DBMS

Source and Target Projects

Source Project: C:\Users\Stephen Maguire\Documents\Corporate Files \Enterprise Model.eapx

Target Project: DBType=-1;Connect=Provider=MSDASQL.1;Persist Security Info=False;User ID=Master;Data Source=Enterprise Repository;

Logfile

☑ Logfile

Caution: The Target Project will be erased prior to transfer. Please ensure you have backed up target if necessary

The performance of the Repository as experienced by end users will depend very much on the quality and power of the size and performance of the server computer and the network infrastructure on which the DBMS and user are located. Using a DBMS over a very high latency (10ms or higher) network can result in significant delays and visibly inferior performance. When network latency is an issue, Sparx Systems recommends using a Cloud Based Server as the interactions are optimized to reduce the effect of network latency.

Also it is important to note that each model is different and although Sparx Systems does its best to maximize performance based on what is expected to be held in a repository, very occasionally this won't be optimal. In these rare cases a review of the database indexes would be good practice to maximize data retrieval and access. This will ensure that end users receive the best possible performance even when models contain millions of constructs.

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 database.
2. Create the required tables in your database, 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 base model, 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:

- Firebird from v2
- MySQL from v5
- MariaDB
- Microsoft SQL Server from 2005, all editions including Express and Azure SQL Database
- Microsoft Access from 2007
- Oracle from 9i (all editions)
- PostgreSQL from v8

For information on creating a project on a specific DBMS from this list, see *Learn more*.

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

Create a Project in a MariaDB Database

To create a project in a MariaDB repository, you work through these stages:

- Stage 1: Create an empty database repository and set up the data tables
- Stage 2: Set up the ODBC Driver
- Stage 3: Set up initial model data

Prerequisites

- A machine with MariaDB version 10.1.2 or higher installed and running
- MySQL ODBC

1. Create a MariaDB Repository

Step	Action
1.1	Create a new empty database and configure it to your installation's defaults.
1.2	Open your SQL management console of choice (such as HeidiSQL) and connect to the new database.
1.3	Load the EASchema_1220_MySQL.sql file from the Sparx Systems website.
1.4	Run EASchema_1220_MySQL.sql to create the required database schema.

2. Set up the ODBC DSN

Step	Action
2.1	Create a suitable ODBC Data Source (using the MySQL ODBC driver and NOT the MariaDB driver), to the new database. Make sure you select the extended option: • 'Return matched rows instead of affected rows'

3. Set up Initial Data

Enterprise Architect repositories require a basic amount of reference data to function correctly.

You can set up this initial data in one of two different ways:

- Transfer the entire contents of an existing model into the new database, or
- Load initial data using a SQL Script

3.1. Load Initial Data - Transfer existing model

If you choose to set up your new database with the contents of an existing model then Sparx Systems recommend that a Project Integrity check is performed on the existing model to ensure it is 'clean' and free from data-related errors before being copied into the new database.

The Enterprise Architect installation provides a model called EABase.eap, which is copied into the installation folder specifically for the purpose of providing basic data for a new model. However, if you have customized the various drop-down lists in another model then you might prefer to use that. Regardless of what model you choose to use, the process is the same and is outlined here.

Step	Action
3.1.1	In Enterprise Architect, open the existing repository and select the 'Configure > Model > Integrity > Project Integrity' ribbon option.
3.1.2	In the 'Action' panel select the 'Report Only' radio button, and in the 'Checks to Run' panel select all check boxes. Click on the **Go button**.
3.1.3	If Enterprise Architect detects that the existing model contains integrity issues, they will be listed in the main panel at the bottom of the dialog. These issues should be resolved before continuing with the project transfer.
3.1.4	With Enterprise Architect open, ensure that there is no project loaded, by clicking on the button and selecting 'Close Project'.
3.1.5	Select the 'Configure > Model > Transfer > **Project Transfer**' ribbon option. The 'Project Transfer' dialog displays.
3.1.6	In the 'Transfer Type' panel, select 'File to DBMS'.
3.1.7	In the 'Source Project' field, click on the ... button and browse for 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.
3.1.8	At the right of the 'Target Project' field, click on the ... button. The 'Datalink Properties' dialog displays.
3.1.9	Select 'Microsoft OLE DB Provider for ODBC Drivers' from the list. Click on the **Next button**.
3.1.10	In the 'Use Data source name' field, click on the **drop-down arrow** and select the ODBC Data Source you configured to point to your new database. Click on the **OK button**. The '**Project Transfer**' dialog redisplays.
3.1.11	If required, select the 'Logfile' checkbox and type a path and filename for the data transfer log file.
3.1.12	Click on the **Transfer button** to begin the data transfer process.

3.1.1 3	When the process is complete, you have created a project on a MariaDB database and can now connect to it and open it from Enterprise Architect.

3.2. Load Initial Data - SQL Script

The Sparx Systems website provides a SQL Script that contains a series of INSERT statements that will load an empty database with all the data needed by Enterprise Architect.

Step	Action
3.2.1	Open your SQL management console of choice (such as HeidiSQL) and connect to the new database.
3.2.2	Load the EABase_1220_MySQL.sql file from the Sparx Systems website.
3.2.3	Run EABase_1220_MySQL.sql to create the required database schema.

Create a MariaDB Repository

Creating a MariaDB database is fully documented in the product information provided with your MariaDB installer or on the MariaDB website.

There are many ways to create a new MariaDB database. One common method is to use a SQL-based Management Console such as HeidiSQL, which provides a convenient graphical user interface to enable creating databases, executing scripts, and performing backups and restores. However, the exact method used is not important to this discussion, therefore you are free to use the most convenient or familiar method at your disposal.

Having created your MariaDB database, use the MySQL SQL script provided by Sparx Systems to create the required table definitions for Enterprise Architect. The EASchema_1220_MySQL.sql script is freely available on the *DBMS Based Repositories* page on the Sparx Systems website.

Notes

- Ensure that the collation is set to the alphabet you use, such as Latin or Cyrillic

Set up a MariaDB ODBC Driver

After you have created a MariaDB database, you will need to set up an ODBC DSN to the new database in order for Enterprise Architect to connect to it. Sparx Systems recommend that the MySQL ODBC driver is used to connect to MariaDB-based repositories, as the native MariaDB ODBC driver is not fully compatible with connection methods employed by Enterprise Architect.

Prerequisites

Install:

- MariaDB DBMS and repository
- MySQL ODBC driver software

Set up the ODBC Driver

Step	Action
1	Under both 32-bit and 64-bit operating systems, Enterprise Architect requires 32-bit ODBC drivers to connect to a repository through ODBC. You can quickly load the correct 32-bit ODBC Data Source Administrator by selecting the 'Configure > User Tools > ODBC Data Sources' ribbon option. The **ODBC Data Source Administrator window** displays.
2	Click on the **Add button**. The 'Create New Data Source' dialog displays, enabling you to add a new DSN.
3	Select the appropriate MySQL ODBC Driver from the list.
4	Click on the **Finish button**. The 'MySQL Connector/ODBC' dialog displays.
5	Enter these configuration details: • A data source name for the connection • A description (optional) • The host address of the DBMS server • User name and password • The database name on the selected server
6	To set the advanced options, click on the Details>> button.
7	Select these checkboxes (where provided): • 'Return matched rows instead of affected rows' ('Connection' or 'Cursors/Results' tab) • 'Allow big result sets' ('Connection' tab)
8	Click on the **Test Connection button** to confirm that the details are correct.
9	If the test succeeds, click on the **OK button** to complete the configuration.

	If the test does not succeed, review your settings.

Notes

- Connecting to DBMS based Repositories is available in the Enterprise Architect Corporate, Unified and Ultimate Editions

Connect to a MariaDB Repository

To access your MariaDB repository, you must provide the connection details to Enterprise Architect.

Prerequisites

- The MariaDB repository and the project already exist
- You know the database user credentials that have SELECT, INSERT, UPDATE, DELETE, EXECUTE and SHOW VIEW access permissions
- The MySQL ODBC driver has been set up

Access

Ribbon	▾ : Open Project
Keyboard Shortcuts	**Ctrl+O**

Connect to the repository

Step	Action
1	In the 'Open Project' dialog, select the 'Connect to Server' checkbox.
2	Click on the ... button. The 'Data Link Properties' dialog displays.
3	Select 'Microsoft OLE DB Provider for ODBC Drivers' from the list.
4	Click on the Next>> button. The 'Connection' tab displays.
5	Click on the 'Use data source name' 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 MariaDB repository.
6	If required, type in a user name and password.
7	If required, type in an initial catalog.
8	Click on the **Test Connection button** to confirm that the details are correct.
9	If the test does not succeed, revise your settings. If the test succeeds, click on the **OK button**; the 'Connection Name & 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 required, select the 'Encrypt Connection String' checkbox. This encrypts and hides the connection details of the database from the users that the connection string is given to.
12	If required, select the 'Lazy Load' 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.
13	If required, select the 'Use WAN Optimization' checkbox. Note: this option is only included for historical reasons, the WAN Optimizer product has been replaced by the vastly improved Pro Cloud Server. The WAN Optimizer was originally created to improve performance over a Wide Area Network by compressing all database requests from Enterprise Architect to reducing transfer times. If you select this checkbox, complete the next three fields (see your administrator for the correct values); otherwise go to step 17.
14	In the 'Server' field, type the network name or address of the optimizer server.
15	In the 'Port' field, type the Port on which the server is running on the remote machine.
16	In the 'DSN' field, type the data source name of the database as it appears on the remote machine.
17	Click on the **OK button** to complete the configuration and open the project. This also adds the project name to the 'Recent' list on the '**Start Page**'; from now on, you can open the project again just by clicking on this name.

Notes

- Connecting to DBMS based Repositories is available in the Corporate, Unified and Ultimate Editions of Enterprise Architect

Create a Project in an MS Access 2007 Database

Setting up a project in a Microsoft Access 2007 repository typically involves the conversion of an existing .EAP or .EAPX file, which can be achieved in two different ways:

- You have a working copy of MS Access 2007 itself, in which case you can convert your existing .EAP/.EAPX project directly to an MS Access 2007 (.ACCDB) file, or
- The existing repository can be transferred into an empty MS Access 2007 database

In either case, before you convert or transfer your existing project, you should perform a project integrity check on the source model to ensure that is 'clean' and free from data related errors.

Prerequisites

- The 'Microsoft Office 12.0 Access Database Engine OLE DB Provider' ODBC driver is installed

1. Perform a Project Integrity Check

While this step is optional, Sparx Systems do recommend that it is performed.

Step	Action
1.1	Using Enterprise Architect, open the source project .EAP/.EAPX file.
1.2	Select the 'Configure > Model > Integrity > Project Integrity' ribbon option.
1.3	Ensure that in the 'Action' panel 'Report Only' is selected, and that all checkboxes in the 'Checks to Run' panel are selected, then click on the **Go button**.
1.4	If Enterprise Architect detects that the existing model contains integrity issues, they will be listed in the main panel at the bottom of the dialog. These issues should be resolved before continuing with the project transfer.

2a. Convert source project with MS Access 2007

Step	Action
2a.1	Open MS Access 2007 and load the source .EAP/.EAPX file within this product. Allow Access to convert the source project to a .ACCDB file. Once this process is complete your project will be an MS Access 2007 repository.

2b. Transfer source project to an empty .ACCDB file

Step	Action
2b.1	Open Enterprise Architect. (If the 'Open Project' dialog displays, click on the **Cancel button** to open with no project loaded.)
2b.2	Select the 'Configure > Model > Transfer > **Project Transfer**' ribbon option. The 'Project Transfer' dialog displays.
2b.3	In the 'Transfer Type' panel, select 'File to DBMS'.
2b.4	In the 'Source Project' field, type the name of the .EAP/.EAPX file to copy to the repository. If the .EAP file has Replication enabled, this must be removed before performing the transfer.
2b.5	At the right of the 'Target Project' field, click on the ... button. The 'Datalink Properties' dialog displays.
2b.6	Select 'Microsoft Office 12.0 Access Database Engine OLE DB Provider' from the list. Click on the **Next button**.
2b.7	Obtain an empty MS Access 2007 base project from the Sparx Systems DBMS Based Repositories web page, and rename the file to suit the intended purpose.
2b.8	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 **OK button** to return to the '**Project Transfer**' dialog.
2b.9	If required, select the 'Logfile' checkbox and type a path and filename for the data transfer log file.
2b.10	Click on the **Transfer button** to begin the data transfer process. When the process is complete, you will have created your project in an Access 2007 database and can open it directly from Enterprise Architect, browsing for the .ACCDB file location in the 'Open Project' dialog.

3. Set up the ODBC DSN

Step	Action
3.1	Create a suitable ODBC Data Source to point to your new database.

Notes

- Connecting to DBMS based Repositories is available in the Enterprise Architect Corporate, Unified 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

Create a Project in a MySQL Database

To create a project in a MySQL repository, you work through these stages:

- Stage 1: Create an empty database repository and set up the data tables
- Stage 2: Set up the ODBC Driver
- Stage 3: Set up initial model data

Prerequisites

- A machine with MySQL version 5 or higher installed and running
- MySQL 32bit ODBC driver, between versions 5.2.4 and 5.3.6 (other versions are not recommended)

1. Create a MySQL Repository

MySQL supports two different storage engines - InnoDB and MyISAM. In older versions of MySQL, MyISAM was the default storage engine, but from MySQL v5.5 onwards InnoDB is the default. As of Enterprise Architect v13, Sparx Systems will no longer provide updates to the MyISAM script and we recommend all users to choose InnoDB, as it supports transactions and UTF8.

Step	Action
1.1	Create a new empty database and configure it to your installation's defaults.
1.2	Open your SQL management console of choice (such as MySQL Workbench) and connect to the new database.
1.3	Load the EASchema_1220_MySQL.sql file from the Sparx Systems website.
1.4	Run EASchema_1220_MySQL.sql to create the required database schema.

2. Set up the ODBC DSN

Step	Action
2.1	Create a suitable ODBC Data Source (using the MySQL ODBC driver) for the new database. Make sure you select these extended options: • 'Return matched rows instead of affected rows' • 'Allow big result sets'

3. Set up Initial Data

Enterprise Architect repositories require a basic amount of reference data to function correctly.

You can set up this initial data in one of two different ways:

- Transfer the entire contents of an existing model into the new database, or
- Load initial data using a SQL Script

3.1 Load Initial Data - Transfer existing model

If you choose to set up your new database with the contents of an existing model then Sparx Systems recommend that a Project Integrity check is performed on the existing model to ensure it is 'clean' and free from data-related errors before being copied into the new database.

The Enterprise Architect installation provides a model called EABase.eap, which is copied into the installation folder specifically for the purpose of providing basic data for a new model. However, if you have customized the various drop-down lists in another model then you might prefer to use that. Regardless of what model you choose to use, the process is the same and is outlined here.

Step	Action
3.1.1	In Enterprise Architect, open the existing repository and select the 'Configure > Model > Integrity > Project Integrity' ribbon option.
3.1.2	In the 'Action' panel, select the 'Report Only' radio button, and in the 'Checks to Run' panel select all checkboxes. Click on the **Go button**.
3.1.3	If Enterprise Architect detects that the existing model contains integrity issues, they will be listed in the main panel at the bottom of the dialog. These issues should be resolved before continuing with the project transfer.
3.1.4	With Enterprise Architect open ensure that there is no project loaded, by clicking on the ▾ button and selecting 'Close Project'.
3.1.5	Select the 'Configure > Model > Transfer > **Project Transfer**' ribbon option. The 'Project Transfer' dialog displays.
3.1.6	In the 'Transfer Type' panel, select 'File to DBMS'.
3.1.7	In the 'Source Project' field, click on the ... button and browse for 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.
3.1.8	At the right of the 'Target Project' field, click on the ... button. The 'Datalink Properties' dialog displays.
3.1.9	Select 'Microsoft OLE DB Provider for ODBC Drivers' from the list. Click on the **Next button**.
3.1.10	In the 'Use Data source name' field, click on the **drop-down arrow** and select the ODBC Data Source you configured to point to your new database. Click on the **OK button**. The '**Project Transfer**' dialog redisplays.
3.1.1	If required, select the 'Logfile' checkbox and type a path and filename for the data transfer log file.

1	
3.1.1 2	Click on the **Transfer button** to begin the data transfer process.
3.1.1 3	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.

3.2 Load Initial Data - SQL Script

Step	Action
3.2.1	Open your SQL management console of choice (such as MySQL Workbench), and connect to the new database.
3.2.2	Load the EABase_1220_MySQL.sql file from the Sparx Systems website.
3.2.3	Run EABase_1220_MySQL.sql to create the required database schema.

Create a MySQL Repository

Creating a MySQL database is fully documented in the product information provided with your MySQL installer.

There are many ways to create a new MySQL database. A common method is to use an SQL-based Management Console such as MySQL Workbench, which provides a convenient graphical user interface for the creation of databases, the execution of scripts, and backups and restores. However, the exact method used is not important to this discussion, and you can use the most convenient or familiar method at your disposal.

Having created your MySQL database, use the SQL script provided by Sparx Systems to create the required table definitions for Enterprise Architect. The EASchema_1220_MySQL.sql script is freely available on the *DBMS Based Repositories* page on the Sparx Systems website.

Notes

- Ensure that the collation is set to the alphabet you use, such as Latin or Cyrillic
- MySQL has a setting (max_allowed_packet) that sets a limit on the amount data that can be returned by the database server. To avoid issues in the future this value needs to be larger that the largest record in your repository. How large will the largest record be? We can't advise that, it depends on how you use Enterprise Architect. Typically the largest elements items like embedded documents and images, baselines of large packages etc. It is not unusual for customers to require a max_allowed_packet of 128M and in some environments even larger than that.

Set up a MySQL ODBC Driver

After you have created a MySQL database, you will need to set up an ODBC DSN to the new database in order for Enterprise Architect to connect to it.

Prerequisites

Install:

- MySQL DBMS and repository
- MySQL 32bit ODBC driver, between versions 5.2.4 and 5.3.4 (see ODBC version note below)

Set up the ODBC Driver

Step	Action
1	Under both 32-bit and 64-bit operating systems, Enterprise Architect requires 32-bit ODBC drivers to connect to a repository through ODBC. You can quickly load the correct 32-bit ODBC Data Source Administrator by selecting the 'Configure > User Tools > ODBC Data Sources' ribbon option. The **ODBC Data Source Administrator window** displays.
2	Click on the **Add button**. The 'Create New Data Source' dialog displays, enabling you to add a new DSN.
3	Select the appropriate MySQL ODBC Driver from the list.
4	Click on the **Finish button**. The 'MySQL Connector/ODBC' dialog displays.
5	Enter these configuration details: • A data source name for the connection • A description (optional) • The host address of the DBMS server • User name and password • Ensure 'Allow saving password' is checked • The database name on the selected server
6	To set the advanced options, click on the Details>> button.
7	Select these checkboxes (where provided): • 'Return matched rows instead of affected rows' ('Connection' or 'Cursors/Results' tab) • 'Allow big result sets' ('Connection' tab)
8	Click on the **Test Connection button** to confirm that the details are correct.
9	If the test succeeds, click on the **OK button** to complete the configuration. If the test does not succeed, review your settings.

Notes

- Connecting to DBMS based Repositories is available in the Enterprise Architect Corporate, Unified and Ultimate Editions
- ODBC Version: v5.3.5 and later of the ODBC can be used however special care is required to ensure the correct Microsoft Visual Studio redistributable is installed to avoid the database driver causing errors. For example versions 5.3.5 to 5.3.14 requires an updated version of the Visual Studio redistributable (that needs to be manually installed), in particular file msvcr120.dll should be at least v12.0.40649.5.

Connect to a MySQL Repository

To access a project in your MySQL data repository, you must connect to the data repository from Enterprise Architect.

Prerequisites

- The MySQL repository and the project already exist
- You know the database user credentials that have SELECT, INSERT, UPDATE, DELETE, EXECUTE and SHOW VIEW access permissions
- The MySQL ODBC driver has been set up

Access

Ribbon	: Open Project
Keyboard Shortcuts	**Ctrl+O**

Connect to the repository

Step	Action
1	In the 'Open Project' dialog, select the 'Connect to Server' checkbox.
2	Click on the ... button. The 'Data Link Properties' dialog displays.
3	Select 'Microsoft OLE DB Provider for ODBC Drivers' from the list.
4	Click on the Next>> button. The 'Connection' tab displays.
5	Click on the 'Use data source name' radio button and on the **drop-down arrow** in the corresponding field. From the list, select the ODBC driver you have set up to connect to your MySQL repository.
6	If required, type in a user name and password.
7	If required, type in an initial catalog.
8	Click on the **Test Connection button** to confirm that the details are correct.
9	If the test does not succeed, revise your settings. If the test succeeds, click on the **OK button**; the 'Connection Name & 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 required, select the 'Encrypt Connection String' checkbox. This encrypts and hides the connection details of the database from the users that the connection string is given to.
12	If required, select the 'Lazy Load' 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.
13	If required, select the 'Use WAN Optimization' checkbox. Note: this option is only included for historical reasons, the WAN Optimizer product has been replaced by the vastly improved Pro Cloud Server. The WAN Optimizer was originally created to improve performance over a Wide Area Network by compressing all database requests from Enterprise Architect to reducing transfer times. If you select this checkbox, complete the next three fields (see your administrator for the correct values); otherwise go to step 17.
14	In the 'Server' field, type the network name or address of the optimizer server.
15	In the 'Port' field, type the number of the Port on which the server is running on the remote machine.
16	In the 'DSN' field, type the data source name of the database as it appears on the remote machine.
17	Click on the **OK button** to complete the configuration and open the project. This also adds the project name to the 'Recent' list on the **Start Page**; from now on, you can open the project again just by clicking on this name.

Notes

- Connecting to DBMS based Repositories is available in the Corporate, Unified and Ultimate Editions of Enterprise Architect

Create a Project in an Oracle Database

To create a project in a repository on Oracle 9i, 10g, 11g or 12c, you work through these stages:

- Stage 1: Create a new Oracle database user to own the Enterprise Architect tables
- Stage 2: Connect to the Oracle repository; Sparx Systems recommends using Oracle Provider for OLE DB - connecting with either Microsoft OLE DB Provider for ODBC or Microsoft OLE DB Provider for Oracle could cause unexpected behavior or performance issues due to known limitations of those drivers
- Stage 3: Set up initial model data

Prerequisites

- A machine with Oracle 9i, 10g, 11 g or 12c installed and running
- Oracle Client installed on the client machine, please ensure that the OLE/DB drivers are installed (you can download the drivers from the Oracle Technology Network web site, as part of the Oracle Data Access Components (ODAC) package; see *Learn more*)

1. Create Database Repository

Step	Action
1.1	Create a new Oracle database user to own the Enterprise Architect tables.
1.2	Open your SQL management console of choice (such as Oracle SQL Developer) and connect to the newly create Enterprise Architect repository.
1.3	Load the EASchema_1220_Oracle.sql file from the Sparx Systems website.
1.4	Run the script to create the required database schema.

2. Set up Oracle connection

2.1	In order for an OLE/DB connection to connect to an Oracle based Enterprise Architect repository, an entry in each user's local TNSNAMES.ORA needs to be defined, stating the server name, Port and other details of the Oracle server.

3. Set up Initial Data

Enterprise Architect repositories require a basic amount of reference data to function correctly.

You can set up this initial data in one of two different ways:

- Transfer the entire contents of an existing model into the new database, or

- Load initial data using a SQL Script

3.1 Load Initial Data - Transfer existing model

If you choose to set up your new database with the contents of an existing model then Sparx Systems recommend that a Project Integrity check is performed on the existing model to ensure it is 'clean' and free from data-related errors before being copied into the new database.

The Enterprise Architect installation provides a model called EABase.eap, which is copied into the installation folder specifically for the purpose of providing basic data for a new model. However, if you have customized the various drop-down lists in another model then you might prefer to use that. Regardless of what model you choose to use, the process is the same and is outlined here.

Step	Action
3.1.1	In Enterprise Architect, open the existing repository and select 'Configure > Model > Integrity > Project Integrity' in the ribbon.
3.1.2	Ensure that the 'Action' panel is set to 'Report Only' and that all items in 'Checks to Run' are enabled, then click on the **Go button**.
3.1.3	If Enterprise Architect detects that the existing model contains integrity issues, they will be listed in the main panel at the bottom of the dialog. These issues should be resolved before continuing with the project transfer.
3.1.4	With Enterprise Architect open, ensure that it there is no project loaded, by selecting 'Close Project' from the ⊗ ▾ button.
3.1.5	Select the 'Configure > Model > Transfer > **Project Transfer**' ribbon option. The 'Project Transfer' dialog displays.
3.1.6	In the 'Transfer Type' panel, select 'File to DBMS'.
3.1.7	At the right of the 'Source Project' field, click on the ... button and browse for 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.
3.1.8	At the right of the 'Target Project' field, click on the ... button. The 'Datalink Properties' dialog displays.
3.1.9	Select 'Oracle Provider for OLE DB' from the list. Click on the **Next button**.
3.1.10	On the 'Connection' page of the 'Datalink Properties' dialog, enter the Oracle service name in the 'Data Source' field, and the user name and password as required. Click on the **OK button**. The '**Project Transfer**' dialog redisplays.
3.1.11	If required, select the 'Logfile' checkbox and type a path and filename for the data transfer log file.
3.1.1	Click on the **Transfer button** to begin the data transfer process.

2	
3.1.1 3	When the process is complete, you have created a project on an Oracle database and can now connect to it and open it from Enterprise Architect.

3.2. Load Initial Data - SQL Script

3.2.1	Open your SQL management console of choice (such as SQL Developer) and log in as the owner of the newly created Enterprise Architect repository.
3.2.2	Load the EABase_1220_Oracle.sql file from the Sparx Systems website.
3.2.3	Run EABase_1220_Oracle.sql to create the required database schema.

Notes

- When transferring a project to Oracle you must have access rights to execute the CREATE SEQUENCE command

Create an Oracle 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, 11g or 12c) and MDAC 2.6 or higher, and obtain access permission to create a new database.

There are many ways to create a new Oracle repository; a common method is to use a SQL-based Management Console such as SQL Developer, which provides a convenient graphical user interface for creating databases, executing scripts, and making backups and restores. However, the exact method used is not important to this discussion, and you can use the most convenient or familiar method at your disposal.

Having created your Oracle repository, use the SQL script provided by Sparx Systems to create the required table definitions and indexes for Enterprise Architect. The EASchema_1220_Oracle.sql is freely available on the DBMS Based Repositories page on the Sparx Systems website.

Notes

- Enterprise Architect only supports the 32-bit versions of Oracle's OLE/DB and ODBC drivers
- For performance reasons, Sparx Systems recommends the use of the Oracle's OLE/DB driver over the Oracle ODBC driver
- Enterprise Architect is not compatible with the Microsoft OLE DB Provider for Oracle driver
- Set the collation to the alphabet you use, such as Latin or Cyrillic

Set up an Oracle ODBC Driver

After you have created an Oracle repository, you can either set up an ODBC DSN to the new database in order for Enterprise Architect to connect to it, or configure Enterprise Architect to use the Oracle OLE DB provider in connection strings to the new database.

Notes:

- For performance reasons, Sparx Systems recommends using Oracle Provider for OLE DB
- Enterprise Architect does not support the 'Microsoft OLE DB Provider for Oracle' driver

Prerequisites

- Install the Oracle DBMS and repository

Set up the Oracle ODBC driver

Your Oracle driver is now available to connect to the repository from Enterprise Architect.

Step	Action
1	Under both 32-bit and 64-bit operating systems, Enterprise Architect requires 32-bit ODBC drivers to connect to a repository through ODBC. You can quickly load the correct 32-bit ODBC Data Source Administrator by selecting the 'Configure > User Tools > ODBC Data Sources' ribbon option. The **ODBC Data Source Administrator window** displays.
2	Click on the **Add button**. The 'Create New Data Source' dialog displays, enabling you to add a new DSN.
3	Select 'Oracle in OraDB11g_home1' from the list (or similar, depending on the ODBC installation).
4	Click on the **Finish button**. The 'Oracle ODBC Driver Configuration' dialog displays.
5	Enter these configuration details: • A data source name for the connection • A description (optional) • The TNS Service Name (click on the drop down arrow and select from the list) • User ID
6	Click on the **Test Connection button** and enter the Oracle user password to confirm that the details are correct.
7	If the test succeeds, click on the **OK button** to complete the configuration. If the test does not succeed, review your settings.

Connect to an Oracle Repository (ODBC)

To access a your Oracle repository, you must provide the connection details to Enterprise Architect.

Note: For performance reasons, Sparx Systems recommends using Oracle Provider for OLE DB (see *Connect to an Oracle Data Repository (OLE DB)*).

Note: Enterprise Architect does not support the 'Microsoft OLE DB Provider for Oracle' driver.

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

Access

Ribbon	: Open Project
Keyboard Shortcuts	**Ctrl+O**

Connect to an Oracle Data Repository using an ODBC Driver

Step	Action
1	In the 'Open Project' dialog, select the 'Connect to Server' checkbox.
2	Click on the ... button. The 'Data Link Properties' dialog displays.
3	Select 'Microsoft OLE DB Provider for ODBC Drivers' from the list.
4	Click on the Next>> button. The 'Connection' tab displays.
5	In the 'Data source' field, click on the **drop-down arrow** and select the data source name.
6	Type in the user name and password.
7	Click on the **Test Connection 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 **OK 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 'Encrypt Connection String' 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 'Lazy Load' checkbox.
13	If required, select the 'Use WAN Optimization' checkbox. Note: this option is only included for historical reasons, the WAN Optimizer product has been replaced by the vastly improved Pro Cloud Server. The WAN Optimizer was originally created to improve performance over a Wide Area Network by compressing all database requests from Enterprise Architect to reducing transfer times. If you select this checkbox, complete the next two fields (see your administrator for the correct values); otherwise go to step 16.
14	In the 'Server' field, type the network name or address of the optimizer server.
15	In the 'Port' field, type the number of the Port on which the server is running on the remote machine.
16	Click on the **OK button** to complete the configuration and open the project. This also adds the project name to the 'Recent' list on the **Start Page**; from now on, you can open the project again just by clicking on this name.

Notes

- Connecting to DBMS based Repositories is available in the Corporate, Unified and Ultimate Editions of Enterprise Architect

Connect to an Oracle Repository (OLE DB)

To access your Oracle repository, you must provide the connection details to Enterprise Architect.

Prerequisites

- Oracle Client already installed on the client machine; please ensure that the OLE/DB drivers are installed (you can download the drivers from the Oracle Technology Network web site, as part of the Oracle Data Access Components (ODAC) package - see *Learn more*)
- The Oracle repository and the project already exist
- You have SELECT, UPDATE, INSERT and DELETE access permissions

Access

Ribbon	: Open Project
Keyboard Shortcuts	**Ctrl+O**

Connect to an Oracle Repository using OLE DB

Step	Action
1	In the 'Open Project' dialog, select the 'Connect to Server' checkbox.
2	Click on the ... button. The 'Data Link Properties' dialog displays.
3	Select 'Oracle Provider for OLE DB' from the list. **Do not** select 'Microsoft OLE DB Provider for Oracle'; Enterprise Architect might not work as expected.
4	Click on the Next>> button. The 'Connection' tab displays.
5	In the 'Data source' field, click on the **drop-down arrow** and select the data source name (the service name of the Oracle database).
6	Type in the user name and password.
7	Click on the **Test Connection 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 **OK button**; the 'Connection Name and 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 you want to encrypt and hide the connection details of the database from the users that the connection string is given to, select the 'Encrypt Connection String' checkbox.
11	If required, select the 'Lazy Load' 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 'Use WAN Optimization' checkbox. Note: this option is only included for historical reasons, the WAN Optimizer product has been replaced by the vastly improved Pro Cloud Server. The WAN Optimizer was originally created to improve performance over a Wide Area Network by compressing all database requests from Enterprise Architect to reducing transfer times. 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 'Server' field, type the network name or address of the optimizer server.
14	In the 'Port' field, type the number of the Port on which the server is running on the remote machine.
15	Click on the **OK button** to complete the configuration and open the project. This also adds the project name to the 'Recent' 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, Unified and Ultimate Editions of Enterprise Architect

Create a Project in a PostgreSQL Database

To create a project in a repository on PostgreSQL, you work through these stages:

- Stage 1: Create an empty database repository and set up the data tables
- Stage 2: Set up the PostgreSQL ODBC Driver
- Stage 3: Set up initial model data

Prerequisites

- A machine with PostgreSQL 9 or higher installed and running
- psqlODBC, version 8.03 or higher has been installed (do not use version 8.3.4 or 8.4.1)

1. Create Database Repository

Step	Action
1.1	Create the empty database and configure it to your installation's defaults.
1.2	Open your SQL management console of choice (such as pgAdmin) and connect to the new database.
1.3	Load the EASchema_1220_PostgreSQL.sql file from the Sparx Systems website.
1.4	Run the EASchema_1220_PostgreSQL.sql to create the required database schema.

2. Set up the ODBC DSN

Step	Action
2.1	Create a suitable ODBC Data Source to the new database, making sure to select these extended options: Page 1: Disable Genetic Optimizer - **Uncheck** Use Declare/Fetch - **Check** Unknowns as LongVarChar - **Check** Bools as Char - **Uncheck** Max Varchar - **1024** Max LongVarChar - **1000000** Page 2: bytea as LO - **Check** Disallow Premature - **Check**

	Protocol - **7.4+**

3. Set up Initial Data

Enterprise Architect repositories require a basic amount of reference data to function correctly.

You can set up this initial data in one of two different ways:

- Transfer the entire contents of an existing model into the new database, or
- Load initial data using a SQL Script

3.1 Load Initial Data - Transfer existing model

If you choose to set up your new database with the contents of an existing model then Sparx Systems recommend that a Project Integrity check is performed on the existing model to ensure it is 'clean' and free from data-related errors before being copied into the new database.

The Enterprise Architect installation provides a model called EABase.eap, which is copied into the installation folder specifically for the purpose of providing basic data for a new model. However, if you have customized the various drop-down lists in another model then you might prefer to use that. Regardless of what model you choose to use, the process is the same and is outlined here.

Step	Action
3.1.1	In Enterprise Architect, open the existing repository and select 'Configure > Model > Integrity > Project Integrity' in the ribbon.
3.1.2	Ensure that the 'Action' panel is set to 'Report Only' and that all items in 'Checks to Run' are enabled, then click on the **Go button**.
3.1.3	If Enterprise Architect detects that the existing model contains integrity issues then it will list them in the main panel at the bottom of the dialog. These issues should be resolved before continuing with the project transfer.
3.1.4	With Enterprise Architect open, ensure that it there is no project loaded, by selecting 'Close Project' from the ▾ button.
3.1.5	Select the 'Configure > Model > Transfer > **Project Transfer**' ribbon option. The 'Project Transfer' dialog displays.
3.1.6	In the 'Transfer Type' panel, select 'File to DBMS'.
3.1.7	In the 'Source Project' field, click on the ... button and browse for 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.
3.1.8	At the right of the 'Target Project' field, click on the ... button. The 'Datalink Properties' dialog displays.
3.1.9	Select 'Microsoft OLE DB Provider for ODBC Drivers' from the list.

	Click on the **Next button**.
3.1.10	On the 'Use Data Source Name' field, click on the **drop-down arrow** and select the ODBC Data Source you configured to point to your new database. Click on the **OK button**. The '**Project Transfer**' dialog redisplays.
3.1.11	If required, select the 'Logfile' checkbox and type a path and filename for the data transfer log file.
3.1.12	Click on the **Transfer button** to begin the data transfer process.
3.1.13	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.

3.2 Load Initial Data - SQL Script

3.2.1	Open your SQL management console of choice (such as pgAdmin) and connect to the new database.
3.2.2	Load the EABase_1220_PostgreSQL.sql file from the Sparx Systems website.
3.2.3	Run EABase_1220_PostgreSQL.sql to create the required database schema.

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

Create a PostgreSQL Repository

Creating a PostgreSQL database is fully documented in the product information provided with your PostgreSQL installer.

There are many ways to create a new PostgreSQL database. A common method is to use an SQL-based Management Console such as pgAdmin, which provides a convenient graphical user interface for creating databases, executing scripts, and performing backups and restores. However, the exact method used is not important to this discussion, and you can use the most convenient or familiar method at your disposal.

Having created your PostgreSQL database, use the SQL script provided by Sparx Systems to create the required table definitions for Enterprise Architect. The EASchema_1220_PostgreSQL.sql script is freely available on the *DBMS Based Repositories* page on the Sparx Systems website.

Notes

- Ensure that the collation is set to the alphabet you use, such as Latin or Cyrillic

Set up a PostgreSQL ODBC Driver

After you have created a PostgreSQL database, you will need to set up an ODBC DSN to the new database in order for Enterprise Architect to connect to it.

Prerequisites

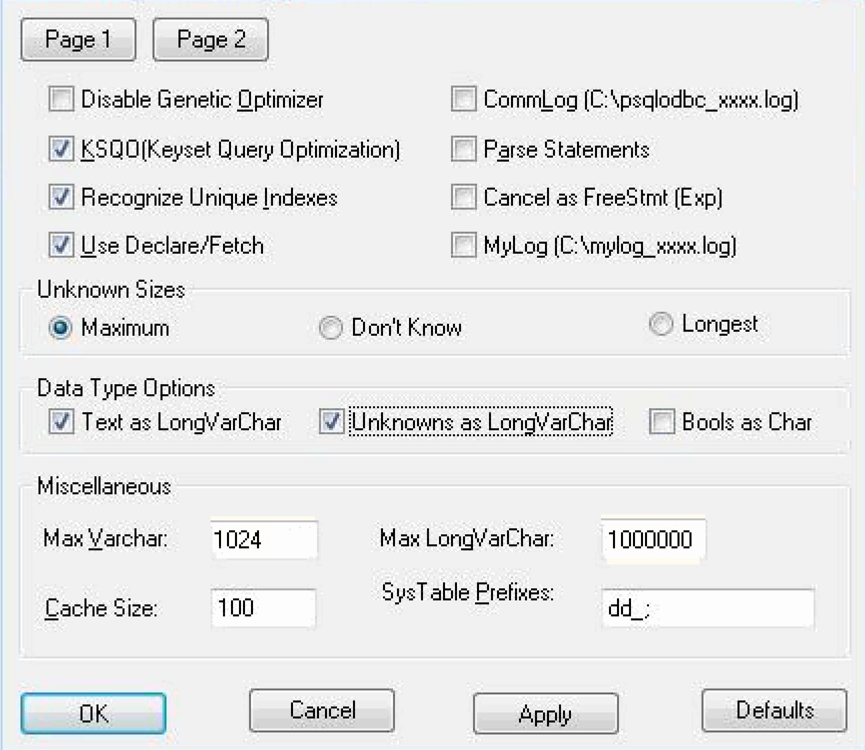
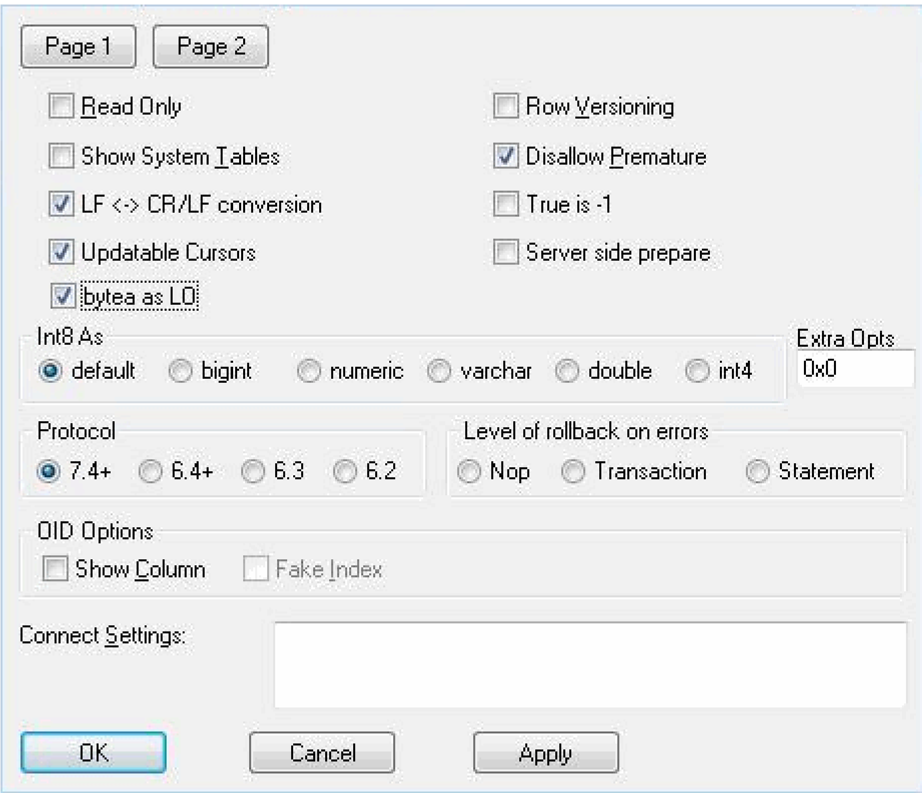
Install:

- PostgreSQL DBMS and repository
- PostgreSQL ODBC driver software version 7.03.01.00 or above (note that versions 8.3.4, 8.4.1 and 9.03 of the PostgreSQL ODBC Driver are not supported)

Set up the ODBC driver

Your PostgreSQL driver is now available to connect to the repository from Enterprise Architect.

Step	Action
1	Under both 32-bit and 64-bit operating systems, Enterprise Architect requires 32-bit ODBC drivers to connect to a repository through ODBC. You can quickly load the correct 32-bit ODBC Data Source Administrator by selecting the 'Configure > User Tools > ODBC Data Sources' ribbon option. The **ODBC Data Source Administrator window** displays.
2	Click on the **Add button**. The 'Create New Data Source' dialog displays, enabling you to add a new DSN.
3	Select 'PostgreSQL UNICODE' from the list.
4	Click on the **Finish button**. The 'Postgre SQL Connector/ODBC' dialog displays.
5	Enter these configuration details: • A data source name for the connection • The actual name of the database • The host address of the DBMS server • User name • A description (optional) • The password
6	To set the advanced options, click on the **Datasource button** and set the options on 'Page 1' as shown:

	 Page 1 Page 2 □ Disable Genetic Optimizer ☑ KSQO(Keyset Query Optimization) ☑ Recognize Unique Indexes ☑ Use Declare/Fetch □ CommLog (C:\psqlodbc_xxxx.log) □ Parse Statements □ Cancel as FreeStmt (Exp) □ MyLog (C:\mylog_xxxx.log) Unknown Sizes: ● Maximum ○ Don't Know ○ Longest Data Type Options: ☑ Text as LongVarChar ☑ Unknowns as LongVarChar □ Bools as Char Miscellaneous Max Varchar: 1024 Max LongVarChar: 1000000 Cache Size: 100 SysTable Prefixes: dd_; OK Cancel Apply Defaults Note: From version 9.5.4 onwards some options shown in this image are no longer available and can be ignored.
7	If you are using a version of PostgreSQL between 8 and 9.5.4, set the options on Page 2 as shown:  Page 1 Page 2 □ Read Only □ Show System Tables ☑ LF <-> CR/LF conversion ☑ Updatable Cursors ☑ bytea as LO □ Row Versioning ☑ Disallow Premature □ True is -1 □ Server side prepare Int8 As: ● default ○ bigint ○ numeric ○ varchar ○ double ○ int4 Extra Opts: 0x0 Protocol: ● 7.4+ ○ 6.4+ ○ 6.3 ○ 6.2 Level of rollback on errors: ○ Nop ○ Transaction ○ Statement OID Options: □ Show Column □ Fake Index Connect Settings: OK Cancel Apply • The 'Disallow Premature' checkbox should be checked • In the 'Protocol' panel, the '7.4+' radio button should be selected Note: From version 9.5.4 onwards the two options are no longer available and can be ignored.

8	Click on the **OK button** to complete the configuration.

Connect to a PostgreSQL Repository

To access your PostgreSQL repository, you must provide the connection details to Enterprise Architect.

Prerequisites

- The PostgreSQL repository and the project already exist
- You know the database user credentials that have SELECT, UPDATE, INSERT and DELETE access permissions
- The PostgreSQL ODBC driver has been set up

Access

Ribbon	: Open Project
Keyboard Shortcuts	**Ctrl+O**

Connect to the repository

Step	Action
1	In the 'Open Project' dialog, select the 'Connect to Server' checkbox.
2	Click on the ... button. The 'Data Link Properties' dialog displays.
3	Select 'Microsoft OLE DB Provider for ODBC Drivers' from the list.
4	Click on the Next>> button. The 'Connection' tab displays.
5	Click on the 'Use data source name' 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 **Test Connection 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 **OK 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 'Encrypt Connection String' 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 'Lazy Load' 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 'Use WAN Optimization' checkbox. Note: this option is only included for historical reasons, the WAN Optimizer product has been replaced by the vastly improved Pro Cloud Server. The WAN Optimizer was originally created to improve performance over a Wide Area Network by compressing all database requests from Enterprise Architect to reducing transfer times. 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 'Server' field, type the network name or address of the optimizer server.
13	In the 'Port' field, type the number of the Port on which the server is running on the remote machine.
14	In the 'DSN' field, type the data source name of the database as it appears on the remote machine.
15	Click on the **OK button** to complete the configuration and open the project. This also adds the project name to the 'Recent' 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, Unified and Ultimate Editions of Enterprise Architect

Create a Project in a SQL Server Database

To create a project on a repository in SQL Server 2005 and above, or SQL Server Express 2005 and above, work through these stages:

- Stage 1: Create an empty database repository and set up the data tables
- Stage 2: Set up initial model data

Prerequisites

- A machine with SQL Server installed and running
- MDAC 2.6 or higher on the client machine
- Permissions to create databases on SQL Server (see the *SQL Server Security Permissions* Help topic)
- ALTER permission on the t_image table in Enterprise Architect, to be able to import model images

1. Create a SQL Server Repository

Step	Action
1.1	Create a new empty database and configure it to your installation's defaults.
1.2	Open your SQL management console of choice (such as Microsoft SQL Server Management Studio) and connect to the new database.
1.3	Load the EASchema_1220_SQLServer.sql file from the Sparx Systems website.
1.4	Run EASchema_1220_SQLServer.sql to create the required database schema.

2. Set up Initial Data

Enterprise Architect repositories require a basic amount of reference data to function correctly.

You can set up this initial data in one of two different ways:

- Transfer the entire contents of an existing model into the new database, or
- Load initial data using a SQL Script

2.1 Load Initial Data - Transfer existing model

If you choose to set up your new database with the contents of an existing model then Sparx Systems recommend that a Project Integrity check is performed on the existing model to ensure it is 'clean' and free from data-related errors before being copied into the new database.

The Enterprise Architect installation provides a model called EABase.eap, which is copied into the installation folder specifically for the purpose of providing basic data for a new model. However, if you have customized the various drop-down lists in another model then you might prefer to use that. Regardless of what model you choose to use, the process is the same and is outlined here.

Step	Action
2.1.1	In Enterprise Architect, open the existing repository and select 'Configure > Model > Integrity > Project Integrity' in the ribbon.
2.1.2	Ensure that the 'Action' panel is set to 'Report Only' and that all items in 'Checks to Run' are enabled, then click on the **Go button**.
2.1.3	If Enterprise Architect detects that the existing model contains integrity issues, they will be listed in the main panel at the bottom of the dialog. These issues should be resolved before continuing with the project transfer.
2.1.4	With Enterprise Architect open, ensure that it there is no project loaded, by selecting 'Close Project' from the button.
2.1.5	Select the 'Configure > Model > Transfer > **Project Transfer**' ribbon option. The 'Project Transfer' dialog displays.
2.1.6	In the 'Transfer Type' panel, select '.EAP to DBMS'.
2.1.7	In the 'Source Project' 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.
2.1.8	At the right of the 'Target Project' field, click on the ... button. The 'Datalink Properties' dialog displays.
2.1.9	Select 'Microsoft OLE DB Provider for SQL Server' or 'Microsoft OLE DB Driver for SQL Server' from the list. Click on the **Next button**.
2.1.10	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 **OK button**. The '**Project Transfer**' dialog redisplays.
2.1.11	If required, select the 'Logfile' checkbox and type a path and filename for the data transfer log file.
2.1.12	Click on the **Transfer button** to begin the data transfer process.
2.1.3	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.

2.2 Load Initial Data - SQL Script

The Sparx Systems website provides a SQL Script that contains a series of INSERT statements that will load an empty database with all the data needed by Enterprise Architect.

Step	Action

2.2.1	Open your SQL management console of choice (such as Microsoft SQL Server Management Studio), and connect to the new database.
2.2.2	Load the EABase_1220_SQLServer.sql file from the Sparx Systems website.
2.2.3	Run EABase_1220_SQLServer.sql to create the required database schema.

Create a SQL Server Repository

Creating a SQL Server database is fully documented in the product information provided with your SQL Server installer.

There are many ways to create a new SQL Server database. A common method is to use an SQL-based Management Console such as Microsoft SQL Server Management Studio, which provides a convenient graphical user interface to create databases, execute scripts, and perform backups and restores. However, the exact method used is not important to this discussion, and you can use the most convenient or familiar method at your disposal.

Having created your SQL Server database, use the SQL script provided by Sparx Systems to create the required table definitions for Enterprise Architect. The EASchema_1220_SQLServer.sql script is freely available on the *DBMS Based Repositories* page on the Sparx Systems website.

SQL Server Management Studio example

Step	Action
1	In SQL Server Management Studio, locate the server on which to create your new repository (such as DBSERVER02\SQLEXPRESS) and login with administrator privileges.
2	Right-click and choose the 'New Database' 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 'New Query' menu option.
5	In the **Query window**, use the 'Open File' dialog to locate the EASchema_1220_SQLServer.sql script file.
6	Click on the **Open button**. In the drop-down menu, check that you have selected the correct database to run the script in.
7	Click on the **Execute button**; SQL Server executes the script, which creates the base tables for an Enterprise Architect project.

Notes

- You should have ALTER permission on the t_image table in Enterprise Architect, to be able to import model images
- 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

SQL Server Security Permissions

The security model implemented by Microsoft's SQL Server is quite powerful and highly configurable, supporting many different possible solutions for securing the data contained in SQL Server databases and ensuring it is only accessible to users with the required permissions. For a more detailed description and explanation of SQL Server permissions, see the SQL Server documentation.

Enterprise Architect users who plan to add, edit and delete contents in a SQL Server repository must have permissions to execute SELECT, UPDATE, INSERT and DELETE statements on all Tables in the Enterprise Architect database. The easiest way to achieve this is to grant the database roles of 'db_datareader' and 'db_datawriter' to each user.

Additional Permissions for Diagram Images

When an Enterprise Architect repository has not been defined with the optional schema update (EASchema_1220_SQLServer_Update1.sql) database users will require additional permissions in order to be able to perform Project Transfers and XML imports that include diagram images.

Why is this the case? In order for Enterprise Architect to be able to store unique images in each repository the value used as the key in T_IMAGE records needs to be set; however, the original schema definition of the T_IMAGE table did not allow this and as a result Enterprise Architect needed to execute SET IDENTITY_INSERT T_IMAGE {ON | OFF} commands, which requires higher permissions; i.e. the role of 'db_ddladmin'.

To avoid the need for each user to have an elevated level of permissions, an optional schema update (EASchema_1220_SQLServer_Update1.sql) was developed which unfortunately is not backward compatible with earlier versions of Enterprise Architect. Therefore the optional schema update (EASchema_1220_SQLServer_Update1.sql) should only be applied if all users are using Enterprise Architect 14 or later. see https://sparxsystems.com/resources/repositories/index.html

Does Enterprise Architect support Windows Authentication?

Enterprise Architect does support Windows Authentication. However, the type of authentication is determined by the configuration of the connection used, and not by Enterprise Architect.

Windows Authentication to SQL Server is commonly used by Enterprise Architect users, but this requires that all Windows users in Enterprise Architect be defined on the SQL Server server and be granted the security roles 'db_datareader' and 'db_datawriter' for the repository.

Connect to a SQL Server Repository

After you have created a Microsoft SQL Server database, you can connect to it immediately without any further configuration directly from Enterprise Architect.

Prerequisites

- The Microsoft SQL Server database/repository already exists and you know the details
- You have the details of a database user with the required permissions within the database, see SQL Server Security Permissions
- Microsoft SQL Server OLE DB driver

Access

Ribbon	: Open Project
Keyboard Shortcuts	**Ctrl+O**

Connect to the repository

Step	Action
1	In the 'Open Project' dialog, select the 'Connect to Server' checkbox.
2	Click on the … button. The 'Data Link Properties' dialog displays.
3	Select 'Microsoft OLE DB Provider for SQL Server' or 'Microsoft OLE DB Driver for SQL Server' from the list. Note: • 'Microsoft OLE DB Provider for SQL Server' is usually part of the Windows Operating System, otherwise it can be installed as part of MDAC 2.8 • 'Microsoft OLE DB Driver for SQL Server' is a newly released driver that adds support for TLS v1.2 and other more recent features of Microsoft SQL Server (such as multi-subnet failover capabilities); this driver is available from the Microsoft website
4	Click on the Next>> button. The 'Connection' tab displays.
5	Type in the server details, including Server Name, User Name and Password.
6	Click on the 'Select the database on the server' option and on the **drop-down arrow**.

	From the list, select the project to connect to.
7	Click on the **Test Connection 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 **OK 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 'Encrypt Connection String' 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 'Lazy Load' 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 'Use WAN Optimization' checkbox. Note: this option is only included for historical reasons, the WAN Optimizer product has been replaced by the vastly improved Pro Cloud Server. The WAN Optimizer was originally created to improve performance over a Wide Area Network by compressing all database requests from Enterprise Architect to reducing transfer times. 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 'Server' field, type the network name or address of the optimizer server.
14	In the 'Port' field, type the number of the Port on which the server is running on the remote machine.
15	Click on the **OK button** to complete the configuration and open the project. This also adds the project name to the 'Recent' list on the **Start Page**; from now on, you can open the project again just by clicking on this name.

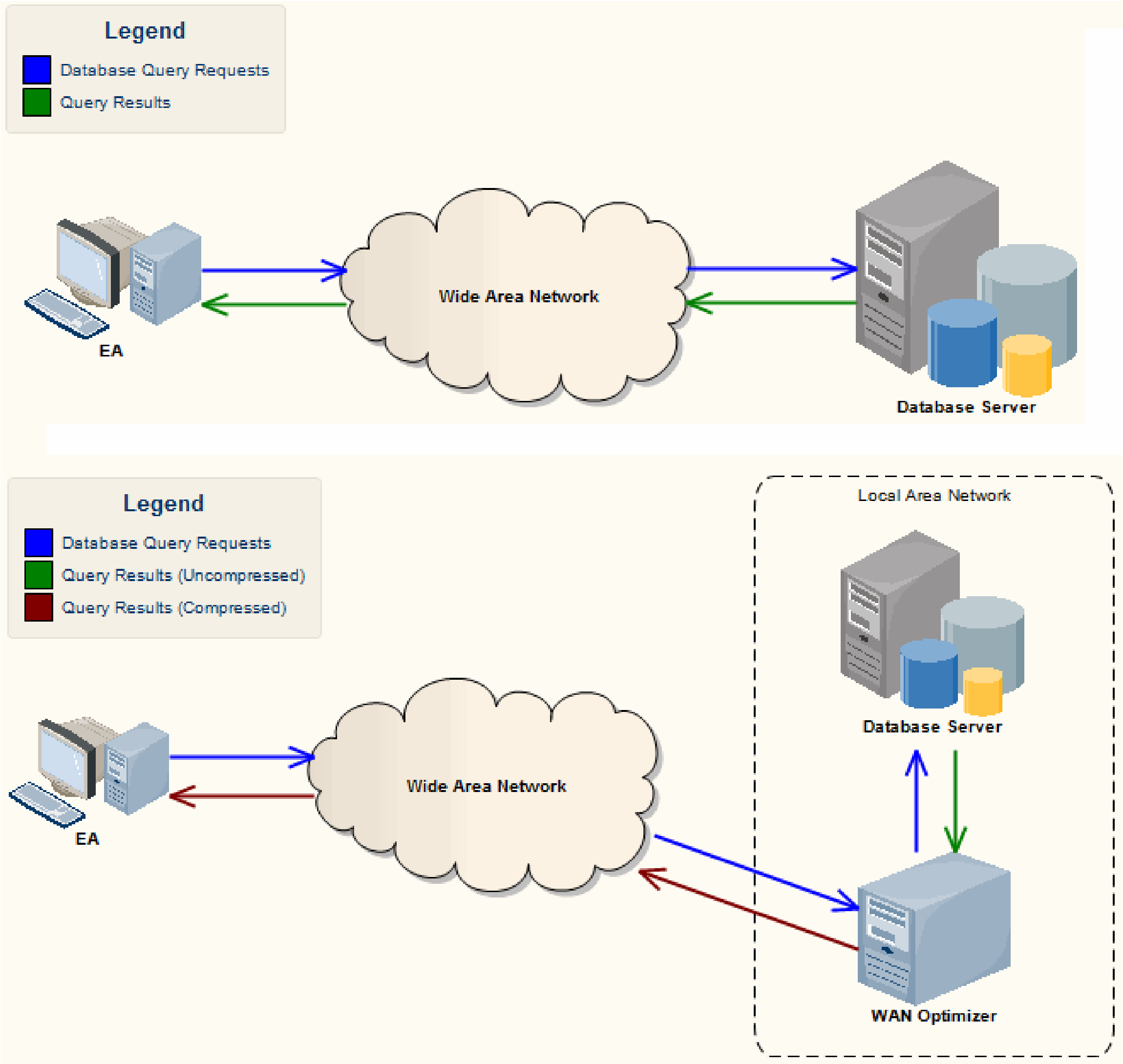
Notes

- Connecting to DBMS based Repositories is available in the Corporate, Unified and Ultimate Editions

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 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 this 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 Systems WAN Optimizer User Guide*. See that Guide for more information on:

- WAN Optimizer Components
- Installing and Starting the WAN Optimizer Service

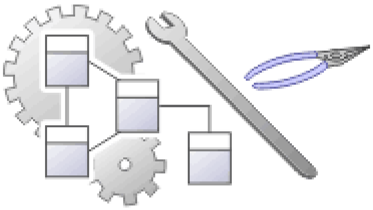
- Configuring the Service
- Troubleshooting

Notes

- The WAN optimizer was the first generation product
- The Sparx Systems Cloud Services was the second generation and replaced the WAN optimizer
- The Sparx Systems Pro Cloud Server is the third generation and has replaced the Sparx Systems Cloud Services
- All Cloud products provided similar basic functionality of the WAN Optimizer, but the later generations do so faster and more securely; they also have many more additional features

Whilst you can still use the WAN Optimizer as described here, we recommend that the Pro Cloud Server should now be used instead of both the WAN Optimizer and Cloud Services products.

Change Management



As users contribute new content and change existing content, a repository will become a valuable data store of organizational information assets. It is imperative that this asset is protected and changes are managed and controlled including being able to revert models back to previous versions or states. Enterprise Architect has a number of sophisticated tools to ensure the information is protected, including full integration with all the leading **Version Control** Systems, **Baselines** that are snapshots of your model that can be taken at important milestones, and **Auditing** that can track the finest changes to a model. A **Project Transfer** function helps you to make backups of models without involving information technology personnel. There are also **Model Validation** and **Project Integrity Checkers** to ensure the repository is maintained with a clean bill of health, and a powerful role based **Security System** to ensure users can collaborate easily and sections of the model can be locked to users or groups.

Facilities

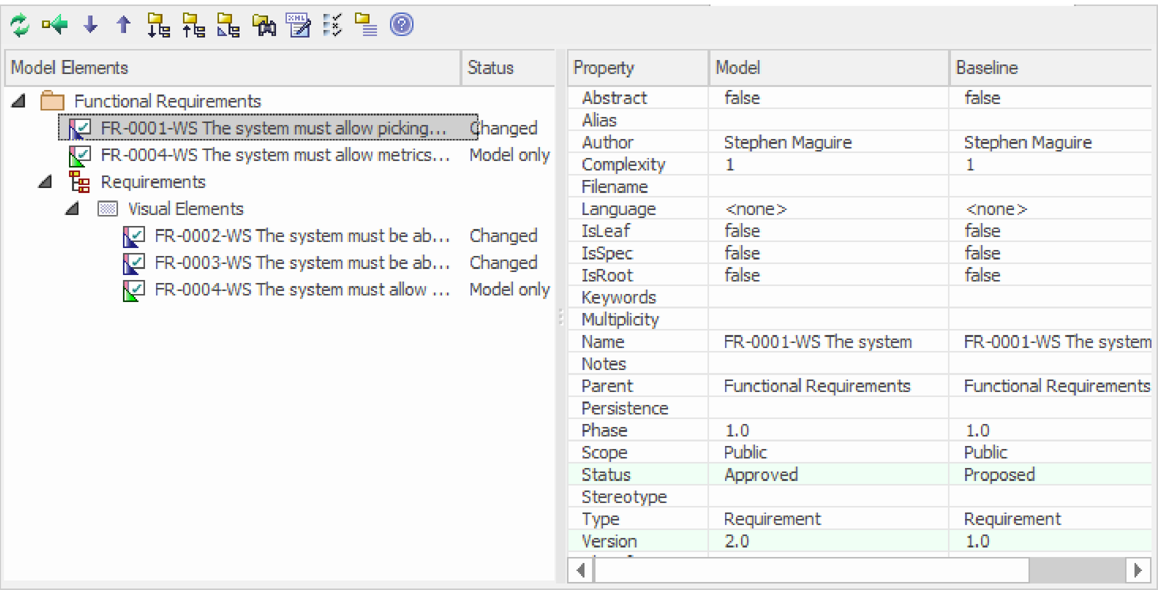
Facility	Detail
Baselines	Baseline create snapshots of model content, a changed model can then be compared to the snapshot and changes can be reverted to the baseline content if required. • **Baselines** are created at the package level providing snapshots of an entire package including elements, features and diagrams and optionally also the packages it contains. • As a model is altered the changed package can be compared with a baseline, including visual diagram changes and if required the current model can be reverted to the content stored in the baseline. • Baselines require no setup and provide an easy way for business and technical users to restore models to previous states at any level of granularity.
Version Control	**Version Control** allows model content to be versioned allowing check-out and check-in of packages to facilitate keeping track of different versions of parts of the model and allowing rollback to prior versions. • 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 • Allow check-out and check-in of packages to and from working storage including users working offline. • Save and retrieve a history of changes to Packages To use Version Control in Enterprise Architect, you require a third-party source-code control application such as: • Subversion • CVS • MS Team Foundation Server (TFS), or

	• Any other Version Control product that complies with the Microsoft Common Source Code Control standard
Auditing	**Auditing** records changes to the content in a repository and provides a powerful visualization view that can be used to drill down into changes from the package level down to changes to element features. • Auditing is typically used by librarians and administrators to investigate who made a change to the model and the date and time it was changed. • It can be used to locate problems in model development or for contractual reasons where third parties have been given access to manage model content and the source of a change needs to be identified. • Auditing can be enabled and disabled at any time and the audit logs can be saved and reload as needed.
Project Data Transfer	Enterprise Architect enables you to transfer project data between project data repositories either for: • Sections of the project (XMI and CSV) or • The whole project, row by row, table by table (in the Corporate, Unified and Ultimate Editions of Enterprise Architect)

Baselines

Snapshot parts of the repository at a given point in time, for later comparison and restoration if required

A Baseline creates a snapshot of a model Package and its contents at a point in time and, as the model changes, allows you to compare the current state and the snapshot; if required you can revert to the previous (baselined) state. **Baselines** are created at the Package level and can include child Packages; any number of Packages can be baselined, and a Package can be baselined any number of times, typically at important project milestones. Baselines are by default conveniently stored within the repository making them available to any model user who has the security privileges to work with them. Alternatively, a Baseline can be stored in a Cloud-based **Reusable Asset Service**. These options will be explored in a later topic.



Model Elements	Status
Functional Requirements	
FR-0001-WS The system must allow picking...	Changed
FR-0004-WS The system must allow metrics...	Model only
Requirements	
Visual Elements	
FR-0002-WS The system must be ab...	Changed
FR-0003-WS The system must be ab...	Changed
FR-0004-WS The system must allow ...	Model only

Property	Model	Baseline
Abstract	false	false
Alias		
Author	Stephen Maguire	Stephen Maguire
Complexity	1	1
Filename		
Language	<none>	<none>
IsLeaf	false	false
IsSpec	false	false
IsRoot	false	false
Keywords		
Multiplicity		
Name	FR-0001-WS The system	FR-0001-WS The system
Notes		
Parent	Functional Requirements	Functional Requirements
Persistence		
Phase	1.0	1.0
Scope	Public	Public
Status	Approved	Proposed
Stereotype		
Type	Requirement	Requirement
Version	2.0	1.0

This diagram illustrates the Baseline comparison tool, showing model and Baseline properties; Status and Version have changed.

Baselines are a type of insurance policy that ensures you can always revert to a previously captured state. In this topic you will learn how to create Baselines, and some of the common business and technical purposes of the technique. You will also learn to use the comparison too,l which allows you to identify what has changed in the model since a Baseline (snapshot) was created. You can reverse changes back to the Baseline state at any level of granularity. This will be indispensable when working with Project Teams using conventional or agile techniques, and where there are important governance or contractual requirements to manage change. You will learn how to manage Baselines, including exploring the options and benefits of storing them in the repository or alternatively in a Reusable Asset Service (RAS).

Brief Introduction

Baselines are easy to set up and require only a small amount of meta-data, such as the Version Number and some Notes that are pre-filled with the time and date the Baseline was created. You can add your own comments such as 'After Requirements Sign-Off' to indicate the significance of a milestone or reason for the Baseline.

The basic steps in working with Baselines are:

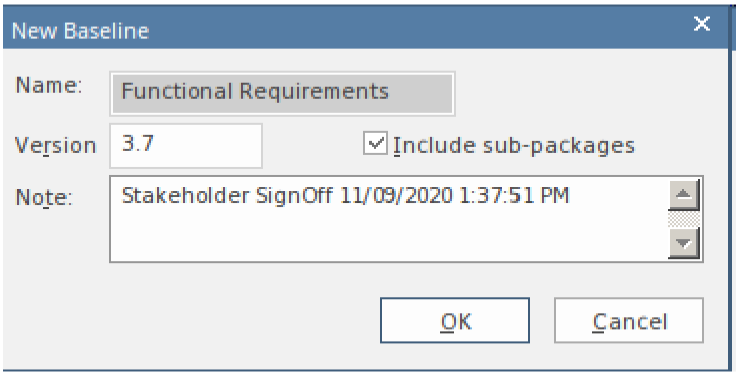
1. Create a Baseline - this will be for a selected Package and optionally its child Packages.
2. Compare the model to the selected Baseline - there can be multiple Baselines of the same Package.
3. Revert the model to one or more Baseline items - or choose to ignore the differences.

It is important to understand a fundamental difference between Baselines and **Version Control**. Baselines are user-initiated, so if a user has not created a Baseline for a given Package at a previous time it is not possible to compare the model to an earlier 'version'. However, once Version Control is set up for a Package, any changes that are made to the Package will be automatically versioned and will be available at a later time. These two facilities can be used in conjunction and it would be the responsibility of a Librarian or Administrator to determine how best to use them together.

Baselines are stored in the same XML format as is used for Version Control, but are stored in compressed format either within the project or in a **Reusable Asset Service** Registry. By default, Baselines are stored within the model; you can save a Baseline to an external XML file for storage or archive, or for distributing to other users working on models derived from a master project.

Creating Baseline

With a package selected in the Browser Window choose the Create Baseline option to create a new Baseline of the selected package. You will be required to complete some details for the baseline including the following:


New Baseline

Name: Functional Requirements

Version 3.7

☑ Include sub-packages

Note: Stakeholder SignOff 11/09/2020 1:37:51 PM

New Baseline window showing details entered when creating a Baseline including the version number.

- Version - a user defined version number used to identify the baseline when running comparisons
- Include sub-packages - choose whether to baseline subordinate packages as well as the main package
- Notes - used to describe the baseline e.g. 'Requirements Sign-Off' pre-filled with the baseline time and date.

Comparing to the Model

The management of baselines makes use of the comparison tool and presents a tabular presentation of the differences. The comparison tool shows the differences between the elements, their properties and features, relationships or visual changes in diagrams at the current time and the time of the baseline or model export. A tree view presents the comparison items and can be used to navigate through the changes which are presented on the panel on the right hand side.

Reverting to Baselines

Once the comparison (differencing) is complete, you can revert changes by merging information from the Baseline into the current project; it is not possible to go the other way.

You can:

- Merge information manually, change by change
- Merge information automatically by electing to merge in all changes in one batch procedure
- Revert completely to the original Baseline by importing the stored XMI directly
- Merge information and elements from a Baseline in a different project, making it possible to keep multiple versions of a single model in synch

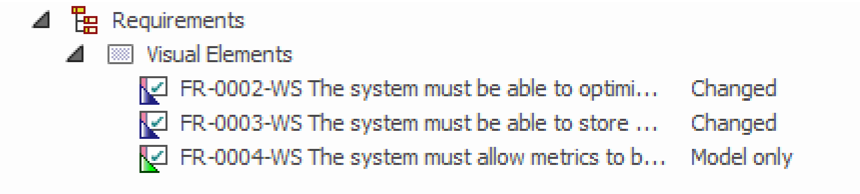
The merge options are available through the toolbar, context menus and the keyboard on the 'Compare Utility' view, which shows the results of a comparison.

Visual Differencing

Changes to a model might include:

- Adding or removing or changing elements and connectors on a diagram, or
- Changing the position of elements or the overall layout of a diagram

You might believe that a diagram has changed, and select to compare it with a baseline using a ribbon option. Alternatively, you might perform a baseline comparison on a Package or a model and select from the comparison output any diagrams that are flagged as changed.



Requirements
Visual Elements
FR-0002-WS The system must be able to optimi... Changed
FR-0003-WS The system must be able to store ... Changed
FR-0004-WS The system must allow metrics to b... Model only

Visual (diagram) differences for a number of existing and new requirement elements.

Creating Baselines

This topic details the basics of creating new **Baselines** of model Packages. You can create Baselines in the model (using the 'Baselines' dialog) or in a **Reusable Asset Service** Registry (using the 'Reusable Asset Service - Baseline' view). The appropriate screen displays by default according to whether the 'Store Package Baselines in a Reusable Asset Service Registry' checkbox is selected on the 'Baselines' page of the 'Manage Project Options' dialog.

Access

Select a Package in the **Browser window**, then open the appropriate screen for the storage system using one of the methods outlined here.

Ribbon	Design > Model > Manage > Manage **Baselines** : 'Baselines' dialog : New Baseline (or) Design > Model > Manage > Manage Baselines : '**Reusable Asset Service** - Baseline' view : New
Keyboard Shortcuts	**Ctrl+Alt+B** : '**Baselines**' dialog > New Baseline (or) Ctrl+Alt+B : '**Reusable Asset Service** - Baseline' view : New

Create a new baseline

Field	Action
Name	Display the Package name of the currently selected model branch.
Version	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.
Include sub-Packages	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 (XMI stubs) of the Package are included in the Baseline.
Note	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).
OK	Click to create a new Baseline and return to the 'Package Baselines' dialog.

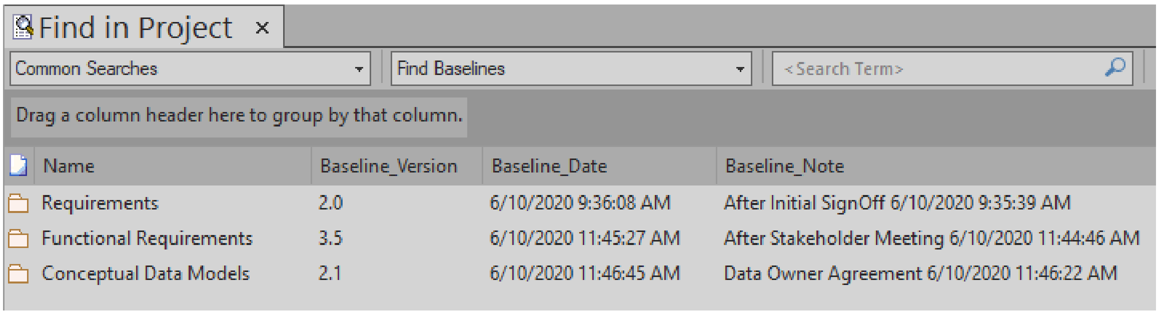
Compare a Model to Baselines

A baseline is created with an intention that it may be required as a backup or point of reference at some time in the future so that a model that has changed can be compared with the snapshot and differences determined. This may be needed just to identify that differences exist or it may be needed so that the entire package or individual changes can be rolled back to the state contained in the baseline (snapshot).

The first step in this process is to locate the required baseline and then to identify the differences. The baseline comparison tool can be used to visualize the changes that have occurred between the time of the baseline and the current time. Individual changes are not recorded as they are in the auditing facility but rather the accumulative results of the changes are presented.

Find Baselines

You can locate a model baselines by selecting a package in the Browser and then opening the Manage **Baselines** window. In the event that you have forgotten or are unsure of the location of one or more baselines you can use the **Model Search** utility to locate all packages that have baselines. The window conveniently displays the name of the Package and the Baseline: Version, Date and Notes. From this window you can locate the package in the **Browser window** and then launch the Manage Baselines window.



Name	Baseline_Version	Baseline_Date	Baseline_Note
Requirements	2.0	6/10/2020 9:36:08 AM	After Initial SignOff 6/10/2020 9:35:39 AM
Functional Requirements	3.5	6/10/2020 11:45:27 AM	After Stakeholder Meeting 6/10/2020 11:44:46 AM
Conceptual Data Models	2.1	6/10/2020 11:46:45 AM	Data Owner Agreement 6/10/2020 11:46:22 AM

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 XMI, 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.

Using the Compare Utility you can Compare a model branch in Enterprise Architect with:

- A Baseline created using the Baseline functionality (Corporate, Unified and Ultimate Editions)
- A Baseline stored in a different model
- An XML 1.1 file created previously using the Enterprise Architect XML export facility (user selects file)
- The current version-controlled XMI 1.1 file as created when using **Version Control** in Enterprise Architect (file automatically selected)

Access

Select a Package in the **Browser window**, then open the '**Baselines**' dialog or '**Reusable Asset Service** - Baseline' view (depending on whether Baselines are stored in model or in a Registry) using one of the methods outlined here.

Ribbon	Publish > Model Exchange > **Package Control** > Compare Package to XMI Design > Model > Manage > Manage **Baselines** : Show Differences ('Baselines' dialog) Design > Model > Manage > Manage Baselines : Compare ('**Reusable Asset Service** - Baseline' view)
Keyboard Shortcuts	**Ctrl+Alt+B** : Show Differences ('**Baselines**' dialog) Ctrl+Alt+B : Compare ('**Reusable Asset Service** - Baseline' view)

Differencing With Baselines

As a Baseline contains all information about 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.

The Differencing engine first builds a representation of the current Package in memory, based on what is currently in the model.

It then compares this with the stored Baseline, highlighting changes, new elements, missing elements and elements that have been moved to other Packages.

It is possible to filter the resultant output to display only one particular kind of change: for example, additions to the model.

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.

See the example provided in the *Example Comparison* Help topic.

Notes

- This utility is available in the Professional, Corporate, Unified and Ultimate Editions of Enterprise Architect
- You cannot compare the current model with an XMI 2.1 file; the utility can only compare with an XMI 1.1 file

Compare Options

You use the 'Baseline Compare Options' dialog 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, or
- Click on the 'Compare Options' icon in the 'Compare Utility' view toolbar

If the 'Compare Utility' view 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.

Options

Option	Action
Always Expand to Differences	Always display the list of elements fully expanded to show changes. If you deselect the checkbox, when the 'Compare Utility' view is first opened it lists the Package contents to element level, and you expand each element as required to show the changed items. For large branches of the model, it is better to leave the checkbox unselected.
Show Elements that are	List elements that: • Have been changed since the Baseline was created • Are in the Baseline only (that is, have been deleted from the model since the Baseline was created) • Are in the model only (that is, have been created since the Baseline was created) • Have not changed since the Baseline was created (you might generally leave this checkbox unselected)
Suppress these Changes	Exclude: • Changes to diagrams • Changes to the 'Date Modified' field for an item • Changes to the 'Date Created' field for an item • Child items of a deleted item • Changes to advanced properties (defaults to selected)
Baseline Diagram Compare Option	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 **Browser window**.

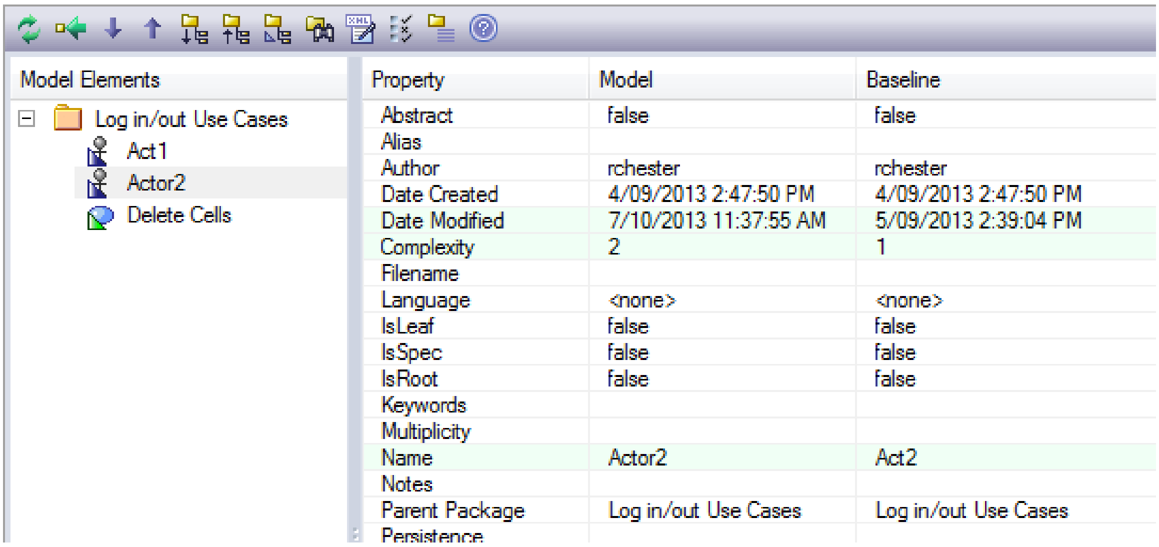
Notes

- Package Baseline facilities are available in the Corporate, Unified and Ultimate Editions of Enterprise Architect

Example Comparison

This diagram 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 'Comparison Utility' view.



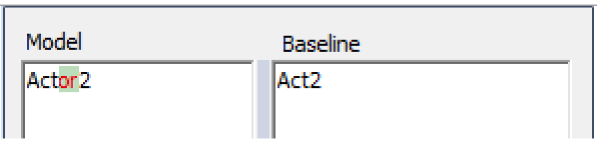
Model Elements

- Log in/out Use Cases
  - Act1
  - Actor2
  - Delete Cells

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

Aspect	Description
Interpretation	A hierarchy of model elements is displayed in the left-hand pane. From the triangle-based icons and the highlighted lines on the report, it is clearly visible which items in the hierarchy have, since the Baseline was captured, been: • Changed • Deleted from the model (in the Baseline only) • Added to the model (in the Model only) or • Switched to different Packages (changes in the Parent Package property) 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. For each property where there is a difference between the model and the Baseline, the row is highlighted. The 'Compare Utility' view enables you to perform operations (such as merging or rolling back changes) on the reported information, using the toolbar, context menu and keyboard.
Increase Level of Detail	The right panel of the 'Compare Utility' view might, for some fields, display only part of the value. It might also not be immediately obvious what a change is. In either case, you can double-click on the property to display full details and to highlight the exact differences; this example shows the highlighted changes to the 'Name' property.

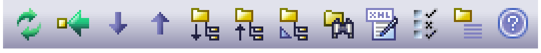
	 Model Actor2 Baseline Act2

Notes

- The Compare utility is available in the Professional, Corporate, Unified and Ultimate Editions of Enterprise Architect

Compare Utility View Options

The 'Comparison Utility' view 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 'Comparison Utility' view
- 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
Refresh	Re-run the comparison to refresh the current display.
Merge To Model	Merge the values of the currently-selected item in the Baseline back into the model.
Next Change	Highlight the next changed item (this skips Moved items).
Previous Change	Highlight the previously-changed item.
Expand All	Fully expand the selected item.
Collapse All	Collapse the changed items in the selected item.
Expand To Changed Items	Expand the selected item to show changed items only (in the event that you have selected to also show unchanged items in the comparison).
Find in Project Browser	Highlight the item in the **Browser window**.
Log To XML	Log the changes to an XML file. A browser displays, on which you specify the file name and location.
Compare Options	Display the 'Baseline Compare Options' dialog.
Manage Package Baselines	Re-display the 'Package Baselines' dialog or '**Reusable Asset Service** - Baseline' view, as appropriate.
Help	Display the Help topic *Package Baselines*.

Context Menu Options

Option	Action
Merge from Baseline Add from Baseline	Restore the item in the model to the Baseline state, or restore a deleted item from the Baseline.
Delete from Model	Remove a recently-created item from the model.
Merge From Baseline (with Options)	(For the root node of the hierarchy on the 'Compare Utility' view.) Display the 'Merge' dialog, which you can use to specify options for rolling back the whole model branch to the Baseline state.
Refresh	(Object-level items.) Re-run the comparison to refresh the current display.
Find in Project Browser	Locate and highlight the item in the **Browser window**.
Open Baseline Diagram Compare	(For a diagram listed in the comparison.) Display the **Baseline Diagram Compare window**, showing differences in diagram content and layout.
Expand All	Fully expand the selected item.
Expand To Changed Items	Expand the selected item to show changed items only.
Collapse All	Collapse the changed items in the selected item.
Log To XML	Log the changes to an XML file. A browser displays, on which you specify the file name and location.
Baseline Compare Options	Display the 'Baseline Compare Options' dialog.

Merge Dialog Options

Option	Action
Changed	Restore all changed items in the model branch to the Baseline state.
In Baseline Only	Restore all deleted items to the model branch from the Baseline.
In Model Only	Remove all recently-created items from the model branch.
Moved	Restore all moved items to their original locations, as identified in the Baseline.
Full Restore from XMI	Completely restore the model branch to the version held in the Baseline XMI 1.1 file, (using the 'XMI 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)

Visual Diagram Changes

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 individual changes if needed to their previous state from the Baseline.

The changes are identified on the 'Baseline Diagram Compare' dialog and on the diagram itself.

Access

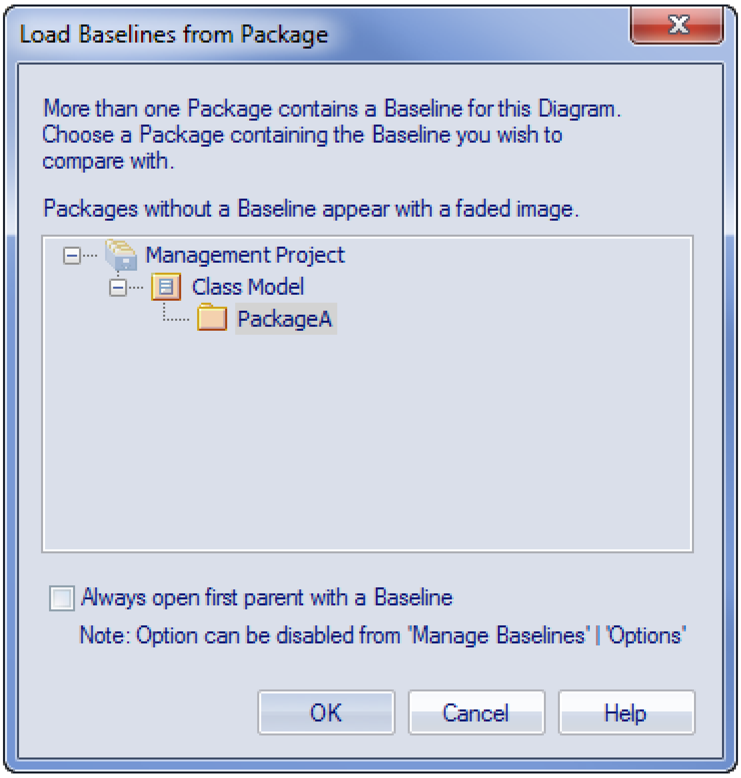
Select the diagram to be checked, then apply one of these access methods:

Ribbon	Layout > Diagram > Options > Compare to Baseline (on an open diagram) Design > Diagram > Manage > Compare to Baseline (on an open diagram)
Other	1. Perform a comparison of a Package and Baseline (see *The Compare Utility (Diff)* Help topic). 2. Locate the diagram in the results on the 'Baseline Comparison' view. 3. Right-click on the diagram name and select the 'Compare to Baseline' option to display the '**Baseline Diagram Compare**' dialog and to open the diagram (refer to the *Results* section).

Multiple Owner Packages

When you create a Baseline, it can be for a Package that contains one or more levels of child Package, and you might create **Baselines** for the Package(s) at each level. If the diagram you are checking is at a lower level in the hierarchy, there might therefore be a number of Baselines that contain information on the diagram, perhaps taken at different times and capturing different changes to the diagram.

When you open the diagram before performing the check, and select one of the direct ribbon access paths, if the diagram is referenced in multiple Baselines you might be required to select the Package from which to use a Baseline, on the 'Load Baselines from Package' dialog.



Load Baselines from Package

More than one Package contains a Baseline for this Diagram. Choose a Package containing the Baseline you wish to compare with.

Packages without a Baseline appear with a faded image.

□ Always open first parent with a Baseline

Note: Option can be disabled from 'Manage Baselines' | 'Options'

The 'Load Baselines from Package' dialog provides the facility to compare the diagram with one of a broader range of Baselines than just those from the diagram's immediate parent.

This dialog displays if you have NOT selected the 'Always open first parent with a Baseline' checkbox on either:

- The 'Load Baselines from Package' dialog itself or
- The 'Baseline Compare Options' dialog

Selecting or deselecting the option in one location also selects or deselects it in the other.

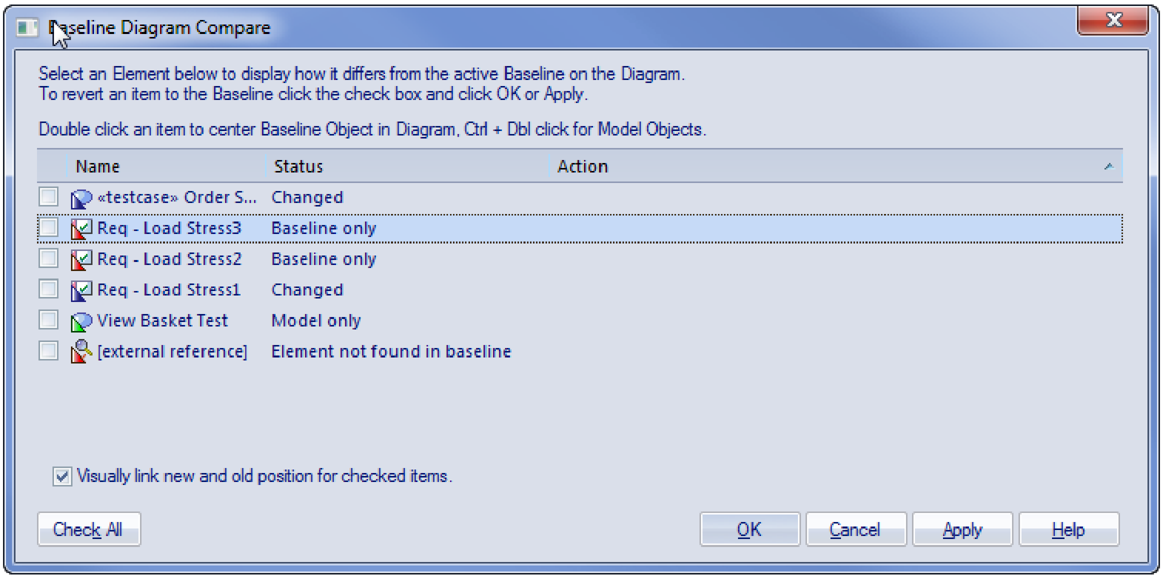
Click on the required Package, and click on the **OK button**. In this case, **or** if the dialog did not display at all (the checkboxes were selected), the 'Baselines dialog displays.

Processing

Click on the required Baseline and on the **Show Differences button**. The '**Baseline Diagram Compare**' dialog displays. Refer to the *Results* section and the *Options* table.

Results

The '**Baseline Diagram Compare**' dialog shows the elements that have been changed on the diagram, and what kind of change was made (examine the 'Status' field).



Baseline Diagram Compare

Select an Element below to display how it differs from the active Baseline on the Diagram.
To revert an item to the Baseline click the check box and click OK or Apply.

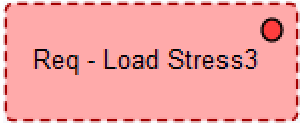
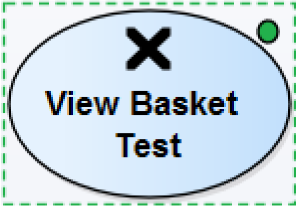
Double click an item to center Baseline Object in Diagram, Ctrl + Dbl click for Model Objects.

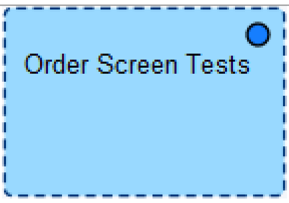
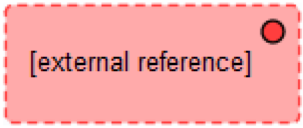
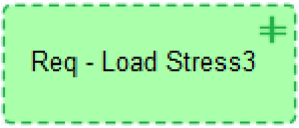
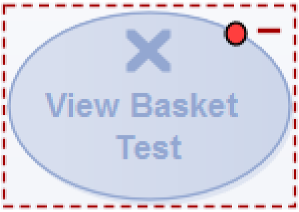
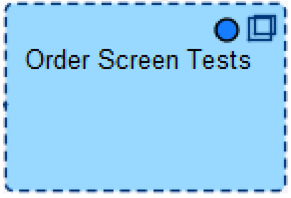
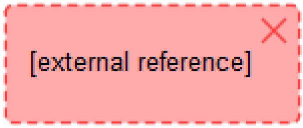
	Name	Status	Action
□	«testcase» Order S...	Changed	
□	Req - Load Stress3	Baseline only	
□	Req - Load Stress2	Baseline only	
□	Req - Load Stress1	Changed	
□	View Basket Test	Model only	
□	[external reference]	Element not found in baseline	

☑ Visually link new and old position for checked items.

As you select elements on the dialog, images are shown on the diagram itself to indicate where the changed element was and what kind of change it underwent.

Options

Option	Detail
Select (click on) name of element	The 'Status' column indicates whether the element has been: - Moved or re-sized (Changed) - Deleted from the diagram (Baseline only) - Added to the diagram since the Baseline was captured (Model only), or - Deleted from its parent external Package, and there is no record in the current Baseline (because the Baseline is only for the current diagram's parent Package) This diagram-linked element has been deleted from the model. The element might be found in a different baseline either in a parent Package baseline or a different Package baseline outside of the current Package. If the external referenced element is restored to the model, the visual comparison will be able to resolve the missing diagram object in the current baseline. When an item is selected, the corresponding element on the diagram is highlighted as shown: - Deleted from the diagram  - Added to the diagram  - Resized or moved to a new position

	 Order Screen Tests - A deleted external element on the diagram  [external reference] The highlighted element on the diagram is marked with a colored dot, as shown, to indicate that it is in focus.
Position the diagram to show the selected element	To scroll the diagram so that you can see the original (Baseline) position of an element, double-click on the item in the list. To scroll the diagram so that you can see the current (model) position of the element, press and hold **Ctrl** while you double-click on the item.
Leave the changes in the item as they are	Ensure that the checkbox against the item is not selected. Click on the **OK button**.
Roll the changes back to the Baseline position	Click on the checkbox against each required item (or click on the **Check All button** to select every item). The 'Action' column displays the action required to roll each element's relationship to the diagram back to the Baseline relationship, and on the diagram the selected elements are represented as shown: - This deleted element will be restored  Req - Load Stress3 - This added element will be removed  View Basket Test - This resized/repositioned element will be put back in its original position  Order Screen Tests - This element from another Package, deleted from the diagram, cannot be restored from this Baseline  [external reference] The comparison automatically shows a blue direction arrow for each reposition or resize that has been checked. For a heavily edited diagram this might be confusing.

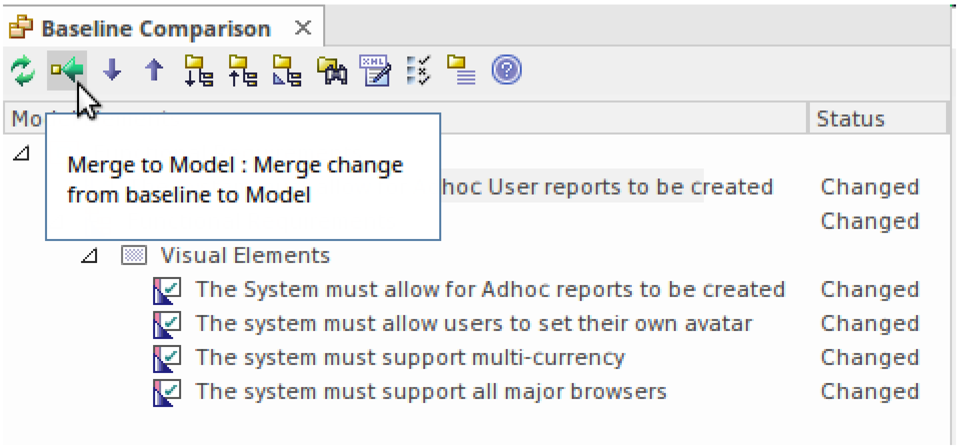
	However, you can hide the arrow for all elements except the one currently in focus; to do this: • Deselect the 'Visually link new and old position for checked items' checkbox To roll back the changes for all items for which a checkbox is selected: • Click on the **Apply button**

Notes

- Diagram Baseline facilities are available in the Corporate, Unified and Ultimate Editions of Enterprise Architect

Revert Model to a Baseline

You can return any part of a model to the content stored in the Baseline. This is a simple process of selecting the item in the Baseline comparison tool and using the 'Merge to Model' toolbar icon.



Baseline Comparison

Merge to Model : Merge change from baseline to Model

Mo	Status
hoc User reports to be created	Changed
	Changed
Visual Elements	
The System must allow for Adhoc reports to be created	Changed
The system must allow users to set their own avatar	Changed
The system must support multi-currency	Changed
The system must support all major browsers	Changed

The 'Merge to Model' option for a selected Requirement element.

The 'Merge to Model' option is useful when you or your team identify unwanted changes that have been made to the model since the Baseline was created. There could be items such as elements, Features, Tagged Values, connectors and diagram objects that have been:

- Deleted
- Modified, or
- Created

You will be warned that this option is not reversible, and if changes are merged from the Baseline to the model the changes to the model will be lost. Note that it is not possible to merge model changes to a Baseline, as the Baseline is a snapshot and cannot be altered.

Manage Baselines

Baselines are snapshots of a repository package taken at a point in time. These snapshots are stored in a format called xmi which is an xml format and these files must be stored so that can be recalled and used for comparisons at a future time. There are two primary places to store baselines:

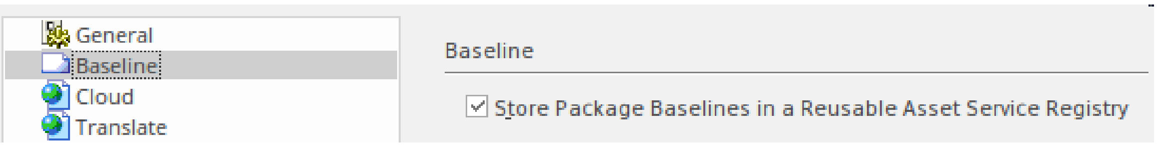
- In the current repository - this is the default location
- In a **Reusable Asset Service** - a reusable asset service needs to be nominated as the location

Baselines are by default stored in the repository, so when for example you create a Baseline of a Requirements Package at a given milestone, the 'snapshot' would by default be stored in the same model as the Requirements. This is expedient and sufficient for small, isolated project teams, but for larger teams and endeavors Enterprise Architect provides some powerful mechanisms to store the Baselines as reusable and universally accessible snapshots available to any project or organizational team, anywhere in the world. This is particularly useful in this age of innovation and agility, where geographically independent teams need to share information and reuse existing assets at Internet speed. The facility uses the Reusable Asset Server (RAS), which is Cloud-based (including on the home premises) and provides sophisticated mechanisms to manage these valuable corporate assets.

A librarian or administrator would typically make the decision about which of these options is most appropriated for a given repository and team. Factors such as whether the baselines need to be made available to a distributed audience and whether dependency analysis is required would be considerations in this decision. The management of the baselines differs depending on the option selected.

Specify Reusable Assert Server for Baselines

To



☑ Store Package Baselines in a Reusable Asset Service Registry

Access

Keyboard Shortcuts	
Ribbon	Configure > Model > Options > Baseline

Manage Baselines in Model

When you manage baselines in the model all of the Baseline features are available through the Manage **Baselines** window. As discussed in a prior topic, this is the default way to work with baselines and the files themselves are stored in compressed format inside the repository. This ensures these baseline snapshots are available to all model users who have the requisite permissions to manage the baselines where security has been enabled.

Storing the baselines in the repository is much more immediate and convenient and is suitable for small teams and situations where baselines are used in a cohesive team environment. If these baselines are important assets that need to be shared or managed by wider communities of users using the Reusable Asset Server may be used to provide access to distributed users.

Access

Ribbon	Design > Model > Manage > Manage **Baselines**
Keyboard Shortcuts	**Ctrl+Alt+B**

Baseline Management

Create, select and process **Baselines** using the 'Package Baselines' dialog.

Option	Action
Current Baselines For Package: <Name>	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.
Show Differences	Run the Compare utility on the selected Baseline and the current model branch or diagram, to display the differences between the two.
Restore to Baseline	Completely restore the model branch from the selected Baseline.
New Baseline	Create a new Baseline.
Delete Selected	Delete the selected Baseline.
Load Other Baselines	Display a drop-down menu that enables you to load **Baselines** from another model, in either a project file or a DBMS repository. • For project files, a browser displays; locate the required project file • For DBMS repositories, the Windows 'Data Link Properties' dialog displays; select the data provider and click on the **OK button** to display the 'Select Data Source' dialog, from which you select the required project In either case, the *Connected To:* message at the bottom of the 'Package Baselines' dialog changes to the name of the alternative model. To return the dialog to the original project, select the third option on the drop down list: 'Load From Selected Package'.

Import File	Import an XML 1.1 file from the file system as a new Baseline for this current model branch.
Export File	Export the selected Baseline to an XML file.
Compare Model to File	Compare the selected model branch with an XML 1.1 file in the file management system; a browser displays, which you use to locate the file.
Options	Set filters to make the comparison more specific.

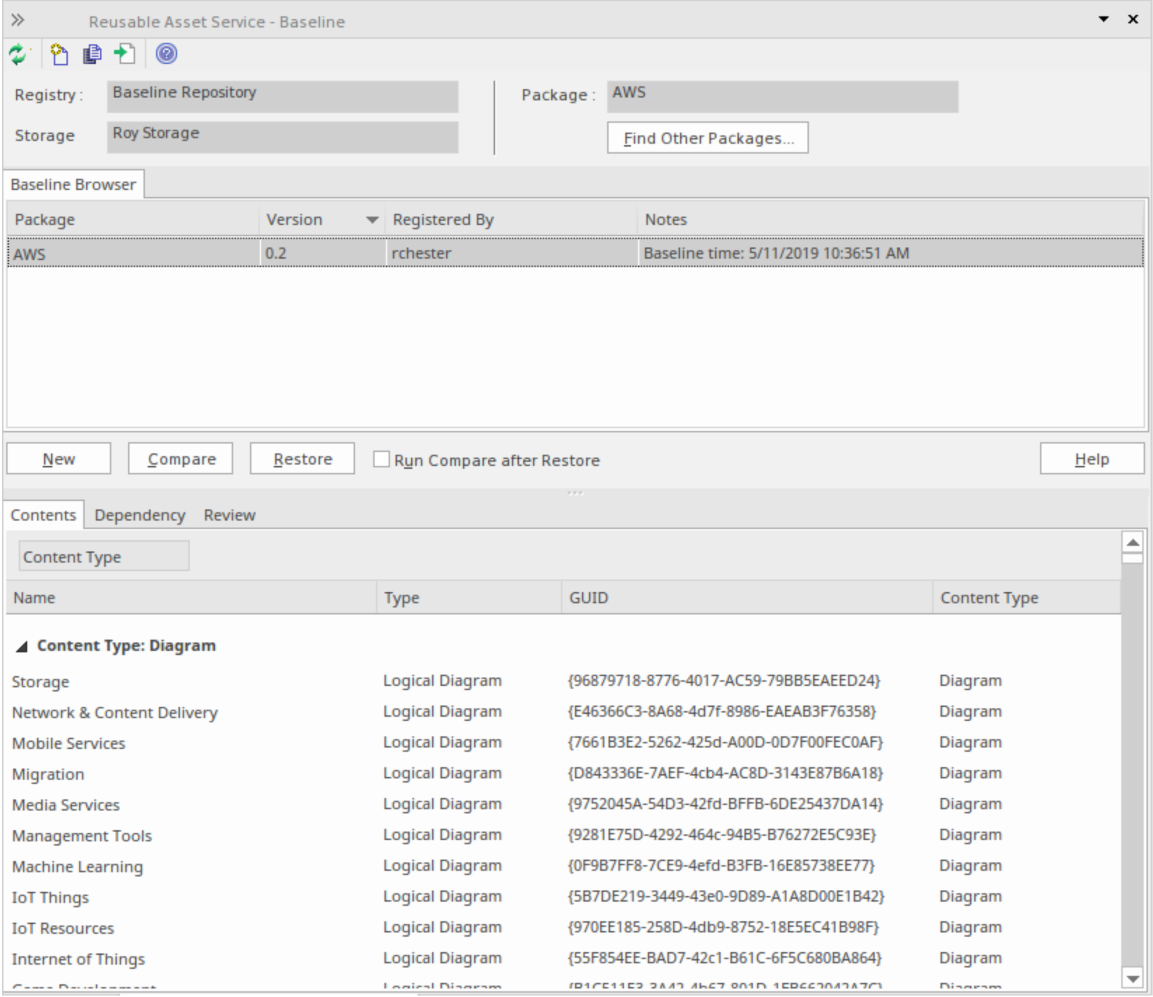
Notes

- Package Baseline facilities are available in the Corporate, Unified and Ultimate Editions of Enterprise Architect

Manage Baselines in Reusable Asset Service

When you manage baselines through the Reusable Asset Server (RAS) all of the Baseline features are available but in addition there are a range of other features only available when using RAS baselines. As discussed in a prior topic the baselines are available to a much wider audience, including other teams and departments, customers, standards authorities, partners, consultants, testers and more, all protected by robust industry standard security. **Baselines** stored in the RAS are more transparent and searchable, making it easier to understand and work with the content they contain.

Enterprise Architect provides a range of facilities for working with and managing Baselines stored in a **Reusable Asset Service** Registry, using the 'Reusable Asset Service - Baseline' view. When you open this view, the 'Baseline Browser' tab immediately displays a list of the Baselines available for the Package that is currently selected in the **Browser window**. You can review and use these Baselines using the context menu and buttons on the tab.



Registry : Baseline Repository

Storage Roy Storage

Package : AWS

Find Other Packages...

Package	Version	Registered By	Notes
AWS	0.2	rchester	Baseline time: 5/11/2019 10:36:51 AM

☐ Run Compare after Restore

Name	Type	GUID	Content Type
Content Type: Diagram			
Storage	Logical Diagram	{96879718-8776-4017-AC59-79BB5EAEED24}	Diagram
Network & Content Delivery	Logical Diagram	{E46366C3-8A68-4d7f-8986-EAEAB3F76358}	Diagram
Mobile Services	Logical Diagram	{7661B3E2-5262-425d-A00D-0D7F00FEC0AF}	Diagram
Migration	Logical Diagram	{D843336E-7AEF-4cb4-AC8D-3143E87B6A18}	Diagram
Media Services	Logical Diagram	{9752045A-54D3-42fd-BFFB-6DE25437DA14}	Diagram
Management Tools	Logical Diagram	{9281E75D-4292-464c-94B5-B76272E5C93E}	Diagram
Machine Learning	Logical Diagram	{0F9B7FF8-7CE9-4efd-B3FB-16E85738EE77}	Diagram
IoT Things	Logical Diagram	{5B7DE219-3449-43e0-9D89-A1A8D00E1B42}	Diagram
IoT Resources	Logical Diagram	{970EE185-258D-4db9-8752-18E5EC41B98F}	Diagram
Internet of Things	Logical Diagram	{55F854EE-BAD7-42c1-B61C-6F5C680BA864}	Diagram

You can review the contents of the selected Baseline using the tabs in the lower half of the 'Reusable Asset Service - Baseline' view, underneath the 'Baseline Browser' tab.

If you have left the 'Baseline Browser' open for a while, there is a possibility that the contents of the Storage in Registry might have changed. Click on the icon in the 'Reusable Asset Service - Baseline' toolbar to refresh the Browser with the latest contents of the Storage.

Note: To store Baselines in a Reusable Asset Service Registry, select the 'Baseline' page in the 'Manage Project Options' dialog and configure the Cloud connection to the Registry, then select or create the Storage in which to store all the Baselines for the Packages in the model.

Access

Ribbon	Design > Model > Manage > Manage **Baselines**
Keyboard Shortcuts	**Ctrl+Alt+B**

Baseline Management

Create, select and process **Baselines** using the '**Reusable Asset Service** - Baseline' view.

Option	Action
Registry	Displays the Registry name, as specified on the 'Baseline' page of the 'Manage Project Options' dialog.
Storage	Displays the Storage name, as defined on the 'Baseline' page of the 'Manage Project Options' dialog.
Package	Displays the name of the Package currently selected in the **Browser window**.
Find Other Packages	Click on this button to open the 'Reusable Asset Services' view, displaying the Storage and its full contents; that is, all Packages with **Baselines**, and any other Packages held in the Storage.
Baseline Browser	Lists the **Baselines**, if any, for the current Package. The Baselines are listed by version - with the highest alphabetical or numerical value at the top. When you select a Baseline in the Browser, details of the Baseline are displayed in the tabs beneath the 'Baseline Browser' tab. **Note**: You can right-click on the column headings and select the 'Toggle **Filter** Bar' option to show or hide the Filter Bar on the display. If you show the Filter Bar, type in the appropriate characters to list only those Baselines that have that string of characters in the value in the corresponding column.
New	Click on this button to create a new Baseline. **Note**: You can also click on the icon in the '**Reusable Asset Service** - Baseline' toolbar to create a new Baseline.
Compare	Click on this button to run the Compare utility on the selected Baseline and the current Package, to display any differences between the two. The results of the comparison are displayed on the 'Comparison Utility' view. **Note**: You can also run the Compare utility by: • Clicking on the icon in the '**Reusable Asset Service** - Baseline' toolbar or • Right-clicking on the Baseline name and selecting the 'Compare' context menu option
Restore	Click on this button to completely restore the Package in the model from the selected Baseline. **Note**:

	You can also restore the Package from the selected Baseline by: • Clicking on the icon in the '**Reusable Asset Service** - Baseline' toolbar • Right-clicking on the Baseline name and selecting the 'Restore' context menu option
Run Compare after Restore	Check this option to automatically run the Compare utility once the Package is completely restored from the selected Baseline.

Notes

- The Package Baseline facilities are available in the Corporate, Unified and Ultimate Editions of Enterprise Architect
- **Reusable Asset Service** Registry details must be configured in the 'Baseline' page of the 'Manage Project Options' dialog
- If a 'Read-Only Access' password is entered for the Storage in the 'Baseline' page of the 'Manage Project Options' dialog, you will not be able to create new **Baselines** in the 'Reusable Asset Service - Baseline' view as the 'New' button will be disabled
- If security is enabled, you must have 'Baselines - Restore model' permission to restore a Package from the selected Baseline

Baseline Contents

In the 'Reusable Asset Service - Baseline' view, when you select a Baseline in the 'Baseline Browser' tab, the three tabs in the lower half of the view are updated with information on the Baseline. The 'Contents' tab lists the diagrams and elements (including any child Packages) held in the selected Baseline, 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

Open the '**Reusable Asset Service** - Baseline' view using one of the methods outlined here.

Select a Baseline in the 'Baseline Browser' tab, then click on the 'Contents' tab to display the contents of that Baseline.

Ribbon	Design > Model > Manage > Manage **Baselines**
Keyboard Shortcuts	**Ctrl+Alt+B**

Review Baseline Contents

Option	Detail
Content Type	Click on this button to toggle between listing the Baseline diagrams first and listing the Baseline elements first.
Toggle Filter Bar	Right-click on the column headings and select this option to show or hide the **Filter** Bar on the display.
(filter bar fields)	Type in the appropriate characters to list only elements and diagrams that have that string of characters in the values in the corresponding column.
Content Type: Diagram Content Type: Element	These are the headers for the two types of object listed in this tab. Click on the appropriate white or black arrow head to hide or show the list of diagrams or elements under the heading.
Name	Displays the name of the element or diagram available in the Baseline.
Type	Displays the type of the element or diagram, such as Use Case or Use Case diagram.
GUID	Displays the Global Unique Identifier of the element or diagram.
Content Type	Displays the object type of the item - Element or Diagram.
Find in Project Browser	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 **Browser window**.
	Right-click on a diagram name and select this option to display the diagram within

View Diagram	a labeled frame, as an image. Alternatively, double-click on the diagram name.

Baseline Dependencies

When you select a Baseline on the 'Baseline Browser' tab, the three tabs in the lower half of the '**Reusable Asset Service** - Baseline' view are updated with information from the Baseline. A Package might contain elements and diagrams that have relationships with objects in other Packages. When you generate a Baseline for that Package, the name and Global Unique Identifier (GUID) of each Package containing these 'external' objects will be stored along with the Baseline and displayed in the 'Dependency' tab.

Note that Package A depends on Package B if any of these constructs (or their Tagged Values) in Package A reference elements in Package B:

- Elements
- Attributes
- Operations
- Operation Parameters
- Diagrams
- Connectors

Access

Open the '**Reusable Asset Service** - Baseline' view using one of the methods outlined here.

Select a Baseline on the 'Baseline Browser' tab, then click on the 'Dependency' tab to display a list of dependencies for that Baseline.

Ribbon	Design > Model > Manage > Manage **Baselines**
Keyboard Shortcuts	**Ctrl+Alt+B**

Check Baseline Dependencies

Option	Detail
Package	Displays the name of the related Package.
GUID	Displays the Global Unique Identifier of the related Package.
Find in Registry	Right-click on the 'Package' line and select this option to see if the Package exists in the Registry. Selecting this option will : • Open the '**Reusable Asset Service**' view • Connect to the Registry and load the Storage that is currently selected in the 'Reusable Asset Service - Baseline' view • Open the 'Search Registry' dialog • Search for the selected Package using its GUID and display the results of the search, if any
Find in Project Browser	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 **Browser window**.
Copy Package Name to Clipboard	Right-click on the 'Package' line and select this option to copy the Package name to the clipboard.
Copy Package GUID to Clipboard	Right-click on the 'Package' line and select this option to copy the Package GUID to the clipboard.

Notes

- The Name and Global Unique Identifier (GUID) of a Package containing 'external' objects will not be stored with the **Baselines** if the option 'Check Package dependency when creating Baseline' is disabled on the 'Baseline' page of the 'Manage Project Options' dialog

Add Review Comments

When you select a Baseline on the 'Baseline Browser' tab, the three tabs in the lower half of the '**Reusable Asset Service** - Baseline' view are updated with information from the Baseline. For any Baseline, you can add individual review comments on any aspect of the Baseline. These comments provide a permanent record on the Baseline - once they have been saved they cannot be edited or deleted. Each comment is attributed to the ID of the user who entered it, and is date stamped.

Access

Open the '**Reusable Asset Service** - Baseline' view using one of the methods outlined here.

Select a Baseline on the 'Baseline Browser' tab, then click on the 'Review' tab to display the review comments of that Baseline.

Ribbon	Design > Model > Manage > Manage **Baselines**
Keyboard Shortcuts	**Ctrl+Alt+B**

Actions

Action	Detail
Review Existing Comments	If there appears to be a longer comment than can be shown in the 'Review' tab, click on it. The full text displays in the 'Comments' field in the lower panel.
Create Comments	Click on the **New button** and start typing your comment on the Package in the 'Comments' field. You can delete and edit text as you type. When you have finished writing your comment, click on the **Save button**. Your comment, preceded by your user ID and the current date, is displayed in the 'Review' panel. Once you have saved your comment, you cannot edit or delete it.

More Information

Edition Information

- Package Baseline facilities are available in the Corporate, Unified and Ultimate Editions of Enterprise Architect
- The Compare utility (used to compare an exported Package with a model Package) is available in the Professional Edition of Enterprise Architect, as well as in the Corporate, Unified and Ultimate Editions
- The Enterprise Architect Corporate, Unified 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

Considerations

Baselines are based on the Global Unique Identifier (GUID) or unique ID of a particular Package:

- Enterprise Architect checks for that ID as the root element within the XML document being used as a Baseline
- 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
- When you save information in a **Version Control** system, the current Version Controlled Package is again the root Package of the document
- It is not useful to create a Baseline by importing an XMI Package file created by Version Controlling a Package that itself contains Version Controlled child Packages; that type of XMI Package file contains stubs for the child Packages, not full information on the child Packages and elements
- 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

XML files must be in the same format used by the 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.

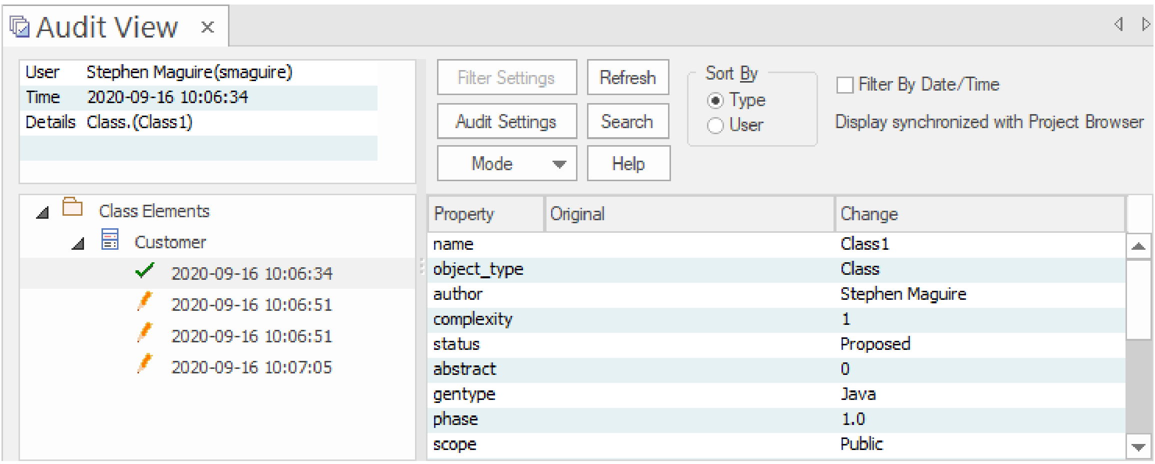
Notes

- 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 workflow 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

Auditing

Enable a Transcript of Model Changes Over Time

Auditing records model changes including when the change occurred and who made it. It records changes to Packages, Elements, Features, Connectors and Diagrams, detailing the creation, modification and deletion of these items. A repository administrator or librarian typically enables auditing during selected project phases or continuously during the lifetime of the repository. An Enterprise Architect repository contains important organization information and ensuring that changes to the models are consistent with an organization's intention and governance principles is critical to the success of this important information asset.



User Stephen Maguire(smaguire)
Time 2020-09-16 10:06:34
Details Class.(Class1)

Class Elements
Customer
✓ 2020-09-16 10:06:34
2020-09-16 10:06:51
2020-09-16 10:06:51
2020-09-16 10:07:05

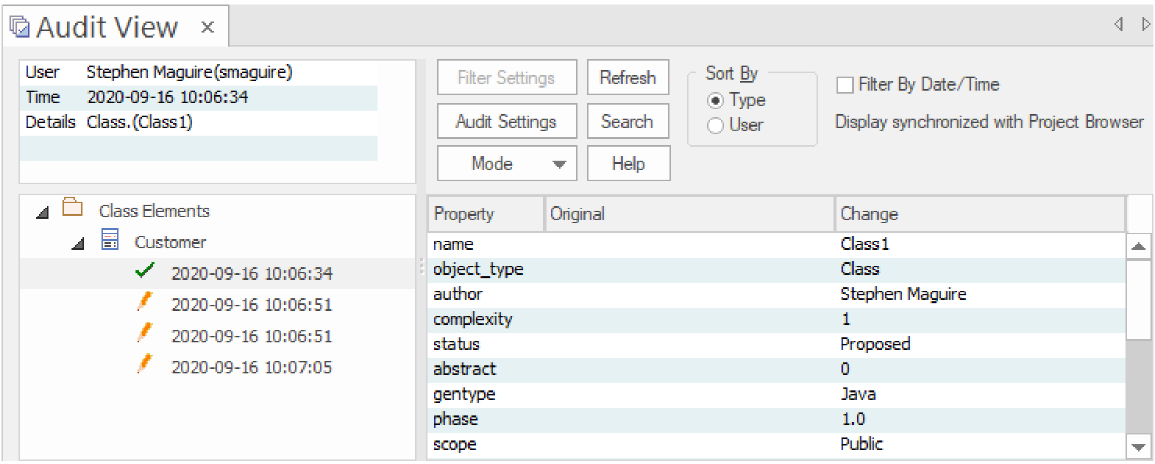
Property	Original	Change
name		Class1
object_type		Class
author		Stephen Maguire
complexity		1
status		Proposed
abstract		0
gentype		Java
phase		1.0
scope		Public

Audit View window showing the creation and change of a Customer Class and its properties.

Audits are a powerful model management mechanism ensuring that changes to models are well understood and governed. In this topic You will learn how to enable auditing, and using the Settings window define the level of detail to record in the audit, the elements to be audited and how batch imports are handled. You can simply enable auditing and it will silently record the changes the users make to the model. You will also learn how to manage audit logs to ensure there are minimal impacts to performance as logs are stored. Auditing can be used for a variety purposes beyond governance for example arresting sub-standard modeling practices by identifying the user and providing them with training and guidance of how to model in a given context.

Brief Introduction

The **Auditing** facility in Enterprise Architect allows changes to the repository to be recorded to an Audit log. This powerful governance tool can be profoundly useful in finding out who changed a model and when it was changed. It is also useful to see the complete list of changes made by a given user or the list of changes made to a given element. The **Audit View** is a dashboard that allows you to display the before and after values of item properties and to drill down from Package level through elements, features and connectors to diagram objects.



User	Stephen Maguire(smaguire)
Time	2020-09-16 10:06:34
Details	Class.(Class1)

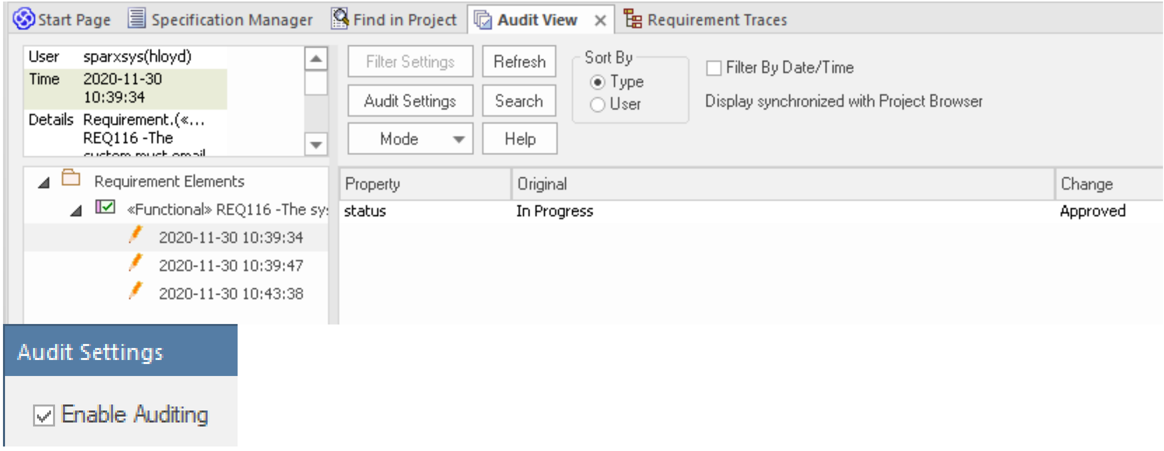
Class Elements
- Customer
  - ✓ 2020-09-16 10:06:34
  - 2020-09-16 10:06:51
  - 2020-09-16 10:06:51
  - 2020-09-16 10:07:05

Property	Original	Change
name		Class1
object_type		Class
author		Stephen Maguire
complexity		1
status		Proposed
abstract		0
gentype		Java
phase		1.0
scope		Public

Audit View showing the creation of, and three subsequent changes to, the Customer Class.

The basic steps to record Audit logs of the changes to a repository are:

1. Enable Auditing - this initiates the process of changes being written to the Audit log.
2. Configure Audit Settings - set the Audit Level and the Audit Options that define which elements will be audited, and set options for importing and reverse engineering.
3. Allow models to be changed - users perform their normal modeling, creating, modifying, and deleting elements, features and diagrams.
4. View the Audit log - to visualize the changes that have been made since the last log clearance.
5. Save and Clear logs - logs can be cleared to increase performance that degrades as the logs fill up.
6. Disable Auditing - optionally Auditing can be disabled if no longer required, and re-enabled when required.



User	sparxsys(hloyd)
Time	2020-11-30 10:39:34
Details	Requirement.(«... REQ116 -The custom must email

Requirement Elements
- «Functional» REQ116 -The sy:
  - 2020-11-30 10:39:34
  - 2020-11-30 10:39:47
  - 2020-11-30 10:43:38

Property	Original	Change
status	In Progress	Approved

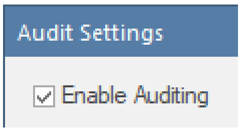
Audit View showing changes made to a Requirement's Status property, from **Validate** to Implemented.

It is important to point out a fundamental difference between Auditing and **Baselines** and **Version Control**. While all three of these facilities are concerned with changes to a repository over time, with Auditing it is not possible to revert the model to a previous state, which is possible with Version Control and Baselines. In contrast with a Baseline Compare, which only displays the original value and the current values of an item property, the Auditing View can display all the incremental changes over time, including when they were made and who made them.

Setting up Auditing is a simple process, and once set up it will begin recording information about what is changed in the repository based on the settings that you have specified. The next few sections provide the basic steps so you and your teams can get started with Auditing and controlling changes to your repository.

Enabling Auditing

You can, as a Librarian or Administrator, enable **Auditing**, which initiates the process of model changes being written to the Audit log. You might choose to only enable Auditing for a period of time such as when a new modeler joins the team, or when contractors have access to the model, or in the final stages of a sprint or iteration.



Audit Settings

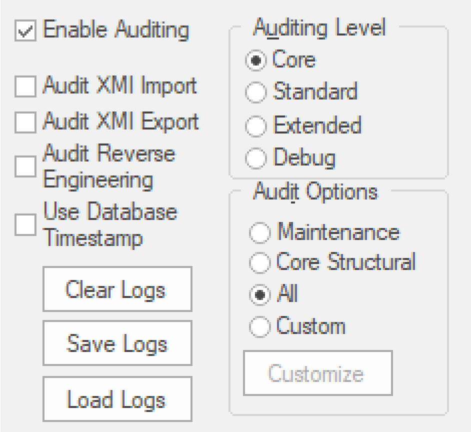
☑ Enable Auditing

Audit Settings window, allowing Auditing to be enabled or disabled.

Auditing can be disabled at any time and enabled again at a later time. The logs, if not cleared, will simply be added to once Auditing is re-enabled.

Configuring Settings

The configuration of the Audit facility is an important step in setting up the tool, to ensure that the changes that you want to be able to visualize will be recorded in the Audit log and will be available when the time comes to view them. There are three main parts to the configuration:



☑ Enable Auditing

□ Audit XMI Import

□ Audit XMI Export

□ Audit Reverse Engineering

□ Use Database Timestamp

Auditing Level: ● Core ○ Standard ○ Extended ○ Debug

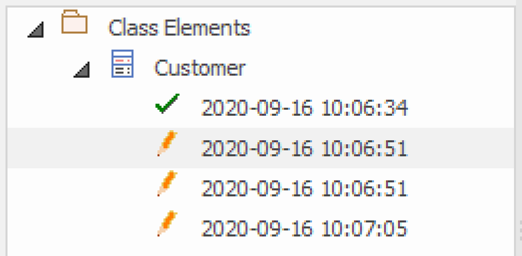
Audit Options: ○ Maintenance ○ Core Structural ● All ○ Custom

Audit Settings window, showing Audit Level, Option, Batch import and Log settings.

- Import and Reverse Engineering options - allow you to specify whether these bulk items should be recorded in the Audit log
- **Auditing** Level - You can specify the extent of the information recorded
- Auditing Options - You can specify which elements will be audited

Viewing Audits

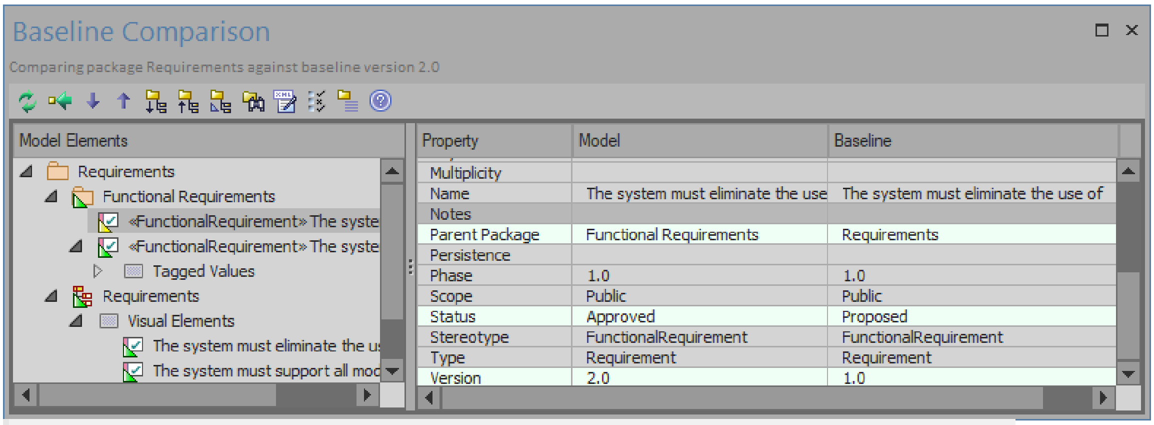
As a Librarian or Administrator, you can view the content of the Audit either through the View Audit window or in the **System Output** window. The **Audit View** provides a number of options for changing the display and provides a convenient Tree View control for navigating through the changes.



- Class Elements
  - Customer
    - ✓ 2020-09-16 10:06:34
    - 2020-09-16 10:06:51
    - 2020-09-16 10:06:51
    - 2020-09-16 10:07:05

Audit View window showing the tree of items and their changes.

Changes are listed and ordered in time order with an icon that indicates if the change was a creation, modification or deletion. A panel on the right displays the change itself, while the header section at the top describes who made the change, when it was made and the nature of the changes.

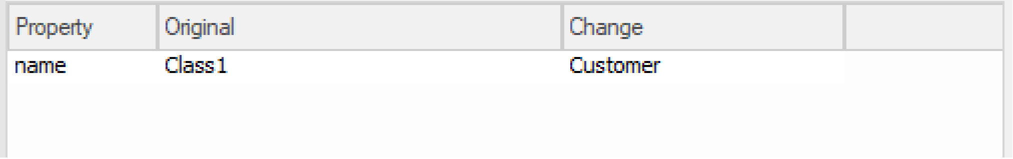


Baseline Comparison

Comparing package Requirements against baseline version 2.0

- Requirements
  - Functional Requirements
    - «FunctionalRequirement» The syste
    - «FunctionalRequirement» The syste
      - Tagged Values
  - Requirements
    - Visual Elements
      - The system must eliminate the u
      - The system must support all moc

Property	Model	Baseline
Multiplicity		
Name	The system must eliminate the use	The system must eliminate the use of
Notes		
Parent Package	Functional Requirements	Requirements
Persistence		
Phase	1.0	1.0
Scope	Public	Public
Status	Approved	Proposed
Stereotype	FunctionalRequirement	FunctionalRequirement
Type	Requirement	Requirement
Version	2.0	1.0

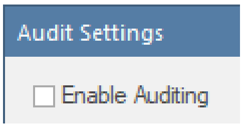


Property	Original	Change	
name	Class1	Customer	

Audit View window showing changes to a Requirement's properties, parent Package, Status and Version

Disabling Auditing

As a librarian or administrator your can disable auditing, which terminates the process of model changes being written to the audit log. You can choose to only enable



Audit Settings

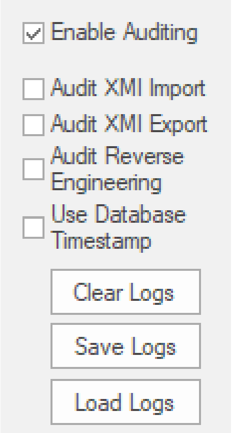
□ Enable Auditing

***Audit Settings window** allowing **Auditing** to be enabled or disabled.*

Auditing can be enabled again at any point in time when required. The logs, if not cleared, will simply be added to once auditing is re-enabled.

Managing Logs

When **Auditing** has been enabled changes to the elements specified in the audit options are recorded in an audit log. The management of logs is critical to the performance of the Auditing facility. When the logs fill up the time it takes to write a change will increase, subtly degrading the performance experienced by modelers as they work in the repository. The management of logs can ensure that users don't notice any perceivable change in the their save times. As the administrator or librarian you can save the logs to a network file and then clear the logs. It is recommended that this process is managed to ensure an optimal user experience. The logs can be managed from the left hand panel of the **Audit Settings window**.


☑ Enable Auditing

□ Audit XMI Import

□ Audit XMI Export

□ Audit Reverse Engineering

□ Use Database Timestamp

Audit Settings options to Clear, Save and Load Logs.

Audit Settings

The **Audit Settings window** allows you to enable or disable auditing but in addition there are a number of options that you can use to manage what is recorded by the **Auditing** facility. These can be altered during different sessions of auditing, for example you may only want to record limited information about requirements during a phase of the project. Subsequently you may required detailed information about all element types and you could make adjustment to the settings to specify this. The important options are summarized below:

- Define whether bulk operations should be included in the audit
- Administer your audit records such as saving and clearing logs
- Specify the detail of the recorded audit by specifying the audit level
- Specify the type of objects you wish to be included in the audit

Access

Ribbon	Configure > Model > Audit : Audit Settings

Configure Settings

Option	Action
Enable Auditing	Select this checkbox to turn the **Auditing** facility on.
Audit XMI Import	Select this checkbox to record changes arising from XMI importing in the audit. As **Version Control** uses XMI, you must select this option to record changes from checking out Packages.
Audit XMI Export	Select this checkbox to record changes arising from XMI exporting in the audit. As **Version Control** uses XMI, you must select this option to record changes from checking out Packages.
Audit Reverse Engineering	Select this checkbox to record changes arising from reverse engineering in the audit.
Use Database Timestamp	Select this checkbox to use the database server's timestamp instead of each user's local timestamp; this improves security. This option is not available for project files.
Clear Logs	Click on this button to permanently delete all log data from the current project; use the **Save Logs** button first, to save the audit records outside the project. The system prompts you to specify whether to clear items logged over a specific period of time. • Click on the **No button** to clear all log items currently held in the database • Click on the **Yes button** to display the 'Time **Filter**' dialog, on which you select a standard time period or define your own This function can be accessed through the **Automation Interface**.

Save Logs	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 **Clear Logs button** to remove them. The system prompts you to specify whether to save items logged over a specific period of time. - Click on the **No button** to save all log items currently held in the database - Click on the **Yes button** to display the 'Time **Filter**' dialog, on which you select a standard time period or define your own This function can be accessed through the **Automation Interface**.
Load Logs	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.
Core	Select this radio button to record changes to elements (including attributes and operations), Packages, connectors and some model-level information.
Standard	Select this radio button to record the same changes as for the Core option, plus changes to some of the 'housekeeping' data for diagrams - Name, Author, Version, Stereotype, Notes and Date Modified. You can check for changes to diagram content and structure using the Baseline facility for reviewing visual changes to diagrams.
Extended	Select this radio button to record the same changes as for the 'Standard' option, plus changes to project security.
Debug	Select this radio button to record the same changes as for the 'Extended' option plus changes to any database tables. Some of the auditing information will not be visible unless the **Audit View** mode is set to 'Raw'.
Maintenance	Select this radio button to audit changes to maintenance elements only; that is: - Package (element) - Requirement - Feature - Use Case - Actor - Note - Issue and - Change
Core Structural	Select this radio button to audit changes to maintenance elements plus certain structural elements; that is: - Package (structure) - Class - Interface - Signal - Node - Component - Artifact

	- Part - Port and - Device
All	Select this radio button to audit changes to all types of element.
Custom	Select this radio button to audit changes to element types that you specify. The **Customize 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 **Select All** button to select every element type). Click on the **OK 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

View the Audits

As an Administrator or Librarian you can view the changes that have been made to the items in the repository. The changes that are recorded depend on three main settings:

- Import and Reverse Engineering options - allow you to specify whether these bulk items should be recorded in the audit log
- **Auditing** Level - You can specify the extent of the information recorded
- Auditing Options - You can specify which elements will be audited

There are two different views of the changes that are available to you as follows:

- The **Audit View** - a fully featured window that allows you to navigate through the changes and change the display
- The **System Output** window - a textual display that changes as the element context changes.

Audit View

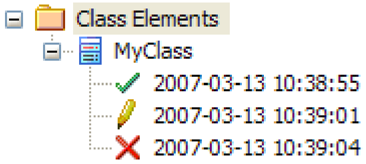
The **Audit View** provides a convenient dashboard of information allowing an administrator or librarian to view and manage the changes to the information that has been recorded by auditing. A tree view control provides a quick way to navigate through a branch of the repository showing all the items that have been changed. Colored Icons provide a quick visual queue as to whether a particular item has been created, changed or deleted. When a change is selected in the tree view the panel to the right displays the properties that have been created, modified or deleted and the top panel shows the responsible user, the time and date of the change and a summary.

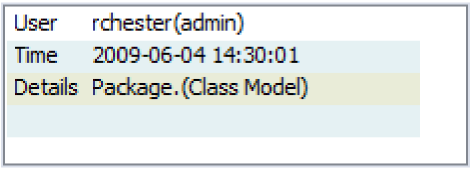
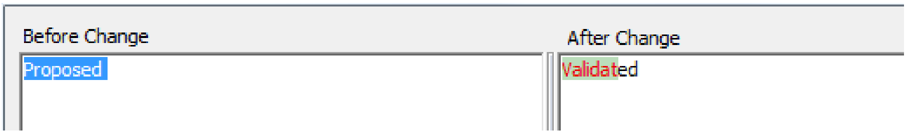
Access

Ribbon	Configure > Model > Audit

Sections of the Audit View

The **Audit View** is divided into three main areas:

Section	Description
View controls	The view controls provide a variety of settings for controlling auditing and the display of audit records.
Audit tree	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: • Sorting • **Filter** (by time) • Mode • **Auditing** settings (what was actually recorded) If synchronizing with the **Browser window**, it is also affected by the Package, diagram or element you have selected.  Class Elements MyClass ✓ 2007-03-13 10:38:55 2007-03-13 10:39:01 ✗ 2007-03-13 10:39:04 In the audit tree: • The green tick indicates a creation • The yellow pencil indicates an edit • The red cross indicates a deletion Right-clicking an element in the audit tree (such as MyClass) displays a context menu that enables you to locate the selected element in: • The Browser window • Any diagrams in which it exists
Record display	The record display is in two parts: the identity of the selected change, and the actual change made.

	The data in the record display is determined by the view controls and mode and, if synchronizing with the **Browser window**, by the Package, diagram or element you have selected. **Identity**  User rchester(admin) Time 2009-06-04 14:30:01 Details Package.(Class Model) The identity of a change consists of: • The Windows username of the user that made the change; if security is enabled, the name is of the format WindowsUsername(SecurityUser) • When the change was made • The path of the change; for example: *Class* Class1 - *Attribute* Att1 - *Attribute Constraint* Constraint **Changes** Changes are displayed in a table format, showing the: • Property (or data item) name • Its original value before the change and • Its value after the change If you double-click on an item in the list of changes (or right-click and select the 'Show Differences' option) the **Difference window** displays:  Before Change: Proposed After Change: Validated This shows the specific changes that have been made, highlighting the edited, created, deleted or formatted characters. Changes to the format 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.

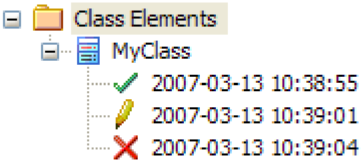
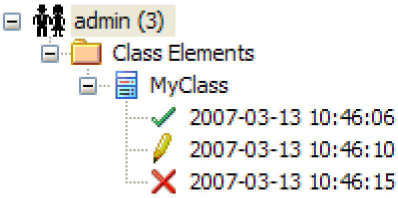
Notes

- If security is enabled, you must have **Audit View** permission to display data in the Audit View

Audit View Controls

The **Audit View** controls provide a variety of settings for controlling auditing and the display of audit records.

Options

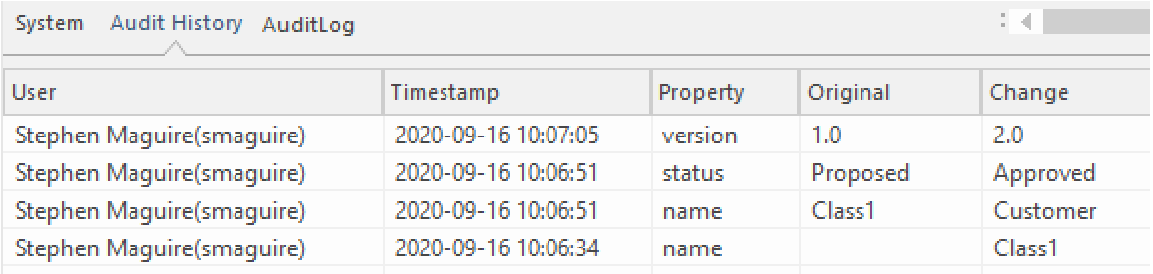
Option	Action
Load or Refresh	Click to reload the Audit Tree, updated with any new audit results.
Search	Click to search through log items for a particular area - you can search by Name, Type or GUID. The items are loaded and filtered with the current Sort By, Time **Filter** and Mode settings. If you refresh the **Audit View**, you must run the search again.
Audit Settings	Click to open the 'Audit Settings' dialog.
Sort-by	Select the appropriate radio button for the required display setting: - Type - changes are grouped under element type (such as Class or Requirement), and then grouped under the changed element  Class Elements MyClass ✓ 2007-03-13 10:38:55 2007-03-13 10:39:01 ✗ 2007-03-13 10:39:04 - User - 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 sort  admin (3) Class Elements MyClass ✓ 2007-03-13 10:46:06 2007-03-13 10:46:10 ✗ 2007-03-13 10:46:15
Filter By Date/Time	Select to enable the **Filter Settings button**, to filter the audit results by time period. Changes that occur outside the selected filter period are not shown in the **Audit View.**
Filter Settings	Click to display the 'Time **Filter**' dialog, to set the filter time period: - Today - to display changes occurring today - Previous Hour - to display changes occurring in the last 60 minutes - Previous 24 Hours - to display changes occurring in the last 24 hours - Previous Week - to display changes occurring in the last 7 days - Previous 30 Days - to display changes occurring in the last 30 days - Previous Year - to display changes occurring in the last 365 days

	• Custom - to define your own time period, using the From and To fields The six pre-configured time periods automatically update when you click on the **Refresh button**; custom periods are static and do not automatically update. If you have set a filter period, and you deselect the 'Filter By Date/Time' 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.
Status Text	Read to see which: • Mode has been selected and • Time filter is being applied to the data
Mode	Click to display a menu of options to change the mode of the **Audit View**. Select: • 'Standard' - to automatically synchronize with the **Browser window**; where changes have been made, the Audit View reflects your selection from the Browser window - if you click on: - An element, the Audit View displays the history for that element - A Package, the Audit View displays the history for that Package and its immediate children (but not the contents of nested Packages) - 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) • 'Advanced' - to load large sets of log items independent of the Browser window; in this mode, a special Audit Settings group can be displayed in the Audit Tree, which records: - When **Auditing** has been enabled and disabled - Who made the change - The date and time of the change - Changes to the Audit Settings - When Audit Operations are executed - Security changes (which can be browsed in the same way as other changes) • 'Deleted' - to display only deleted records, but otherwise data is shown as in 'Advanced' mode; records can be sorted by element type or by user as required • 'Raw' - 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 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 Additional database information is displayed; this additional information might be insignificant or only in machine-readable format

Audit History Tab

When **Auditing** is turned on, an 'Audit History' tab is enabled and can be viewed in the **System Output** window. The tab shows a history of changes to whichever element or connector you have selected in the:

- Current diagram
- **Browser window**
- **Audit View**, or
- **Package Browser**



User	Timestamp	Property	Original	Change
Stephen Maguire(smaguire)	2020-09-16 10:07:05	version	1.0	2.0
Stephen Maguire(smaguire)	2020-09-16 10:06:51	status	Proposed	Approved
Stephen Maguire(smaguire)	2020-09-16 10:06:51	name	Class1	Customer
Stephen Maguire(smaguire)	2020-09-16 10:06:34	name		Class1

Output window Audit History Tab showing changes to the elements in the repository.

As you select different elements or connectors, the 'Audit History' tab automatically updates to reflect your current selection. For each change made to the element or connector, the tab shows:

- Who made the change
- When the change was made
- Where the change was made
- The value of the characteristic before the change
- The value of the characteristic after the change

Access

Ribbon	Start > Desktop > Design > **System Output** > Audit History

Notes

- To see this tab, you must have the **Audit View** open; the tab continues to display if you subsequently close the Audit View
- If security is enabled, you must have Audit View permission to display data on the 'Audit History' tab

Performance Considerations

Auditing is a powerful tool for determining who made changes to the model and when the changes were made. When auditing has been enabled every time a change is made to the model the Enterprise Architect auditing engine records the change. These changes will naturally, like any transaction system, take some to be written to the audit log. When the logs are small this write time will be imperceivable but when the logs grow in size the time taken to write to the log will increase and could delay the saving of changes. The tool provides a number of ways to reduce theses delays in the saving of changes. The delays can be experienced in two different parts of the auditing process.

- **When making changes to the repository.** As described above the saving of model changes may be delayed effecting you or any user who is updating an aspect of the model that is being audited. For example a change to an element's name, changing the position of an object on a diagram or if configured the creation of tables or code classes resulting from reverse engineering a database or code package will all result in a write to the audit logs.
- **When accessing the Audit View.** The audit view that is typically accessed by a librarian or administrator can become slow dues to the volume of audit records that need to be sorted and displayed.

The performance of both these aspects of the auditing process will be covered in the following topics

Auditing Performance

Enabling auditing on a project increases the time taken for most actions such as insertions, changes and deletions of objects and changes to properties of elements, connectors and diagrams including moving objects in diagrams. When auditing is initially enabled the time required to write to the audit logs will be almost unperceivable but as the logs grow in size the log write time will become longer and may cause some frustration to you or you colleagues making whose man focus is creating useful models. Fortunately there are some simple housekeeping and administration procedures that can reduce this time typically performed by a librarian or administrator..

Operation Delays

Operation	Detail
Large Deletions	Deleting large Packages or Package structures, or large numbers of elements, takes significantly longer with auditing on. You might disable auditing before performing such a deletion.
XMI Imports	Importing XMI takes longer with auditing enabled. A project-level option is provided for disabling auditing of XMI Imports.
Reverse Engineering	Reverse engineering code takes longer with auditing enabled. A project-level option is provided for disabling auditing of reverse engineering.
Replication	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.

Audit View Performance

As the logs become larger the **Audit View** can exhibit slow performance, simply because of the volume of records it needs to process to create the visualizations that are used by the librarian or administrator. Fortunately there are ways to reduce the delays to the presentation of changed items and these only require a small amount of house keeping.

Considerations and Responses

Consideration	Detail
Navigating the Browser window Within Auditing is Slow	Try setting the time filter to a period in the immediate past, such as 'Today', 'Previous 24 Hours' or 'Previous Week'; this time period updates each time you open or refresh the **Audit View**. Save log items outside the project with the **Save Logs** 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 **Load Logs button**.
The Audit View is Slow in Loading and Changing Modes	Try setting the time filter to a period in the immediate past, such as 'Today', 'Previous 24 Hours' or 'Previous Week'; this time period updates each time you open or refresh the **Audit View**. Save log items outside the project with the **Save Logs** 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 **Load Logs button**.
Navigating the Audit Tree is Slow	Close the 'Audit History' tab in the **System Output** window

More Information

Edition Information

- The **Auditing** facility is available in the Corporate, Unified and Ultimate Editions
- Warning - If your site runs separate Editions of Enterprise Architect, when Auditing is turned on in a project any Professional Edition users are locked out of the project; to restore access, turn Auditing off in the project from the Corporate, Unified or Ultimate Edition instance of Enterprise Architect

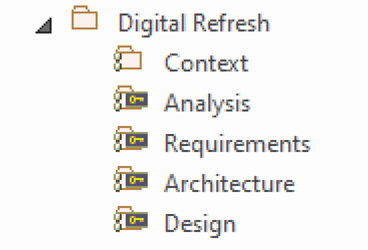
Related Tools

- Version Control
- Baselines
- The Compare Utility (Diff)

Version Control

Stores Versions of parts of a Repository for Comparison and Rollback

Version Control provides a mechanism to store previous versions of any section of a model and later compared to a current state. Version Control is a powerful team feature that allows models to be developed in a coordinated way within a team environment.   Sections of the model can be placed under Version Control and you can manage changes to and revisions of your model content by placing individual model Packages, view nodes or root nodes under Version Control. The information stored within the repository are valuable corporate assets and need to be managed as such.



Browser window showing one Checked-Out Package (Context) and four Checked-In Packages under Version Control.

Version Control provides a safeguard against losing previous model changes. In this topic you as the Librarian will learn how to configure a Version Control product for use within Enterprise Architect and to apply the configuration to control Packages, individually or in groups. You will also learn how to use Version Control to Check-Out and Check-In Packages, how to view previous versions, and compare a current version to a previous version.

Brief Introduction

The **Version Control** feature allows you to check-out a Package, which then **locks** it so that no other user can modify it until you have checked it back in again. You can then modify the Package, adding new elements and diagrams, changing others and deleting still others. When you have completed your changes and are ready for other modelers to see your work, you can check-in the Package. Other modelers would then need to do a Get Latest to pull the new changes down from the Version Control Server.

Enterprise Architect does not version the Packages internally; instead it integrates with industry standard Version Control servers, which an administrator must install and configure for use by Enterprise Architect. The files that Enterprise Architect uses are industry standard XMI files, which are best seen as 'binary' from the point of view of Version Control. This means that it is not possible for two modelers to be working on the same file, as it would not be possible to merge the files as is possible with source code or text files.

The basic steps to use Version Control are:

1. Select and set up a Version Control Server and Client side Software.
2. Place the required Packages under Version Control.
3. Check Out a Package.
4. Make modifications to the contents of the Checked Out Package.
5. Check In the Package, entering comments that describe the changes.
6. Compare the Repository Package with the Version Controlled Package.
7. Retrieve (rollback to) a previous version.

Enterprise Architect's Version Control integration provides several key facilities, including:

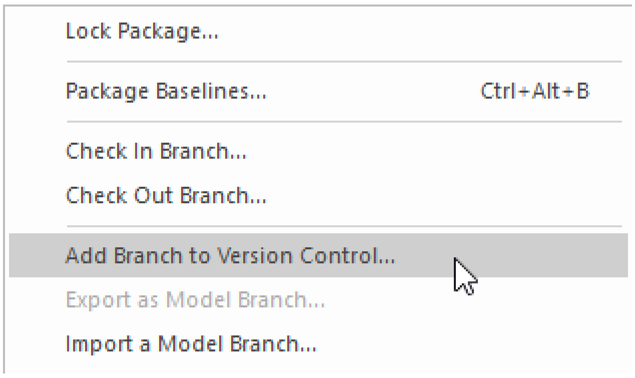
- Saving a history of changes made to your model's Packages
- Retrieving previous revisions of Packages
- Propagating model updates between team members
- Coordinating the sharing of Packages between team members

Commissioning a Version Control System

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 or a librarian must define a Version Control Configuration within your project. You can then use Version Control to manage changes to the Packages of your model.

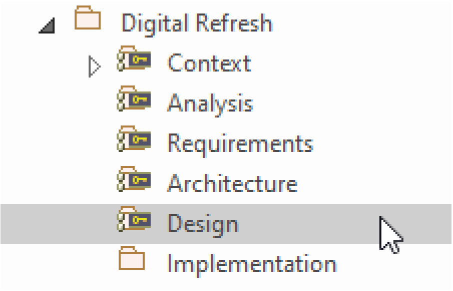
Configuring Packages

Before you can Check Out a package and begin making changes you or the librarian must first configure the package for **Version Control**. You simply select the Add Branch to Version Control option and choose an existing configuration. This to adds the package to the Version Control System and places the package in a Checked-In state.


Lock Package...
Package Baselines... Ctrl+Alt+B
Check In Branch...
Check Out Branch...
Add Branch to Version Control...
Export as Model Branch...
Import a Model Branch...

Context menu showing the Add Branch to Version Control option.

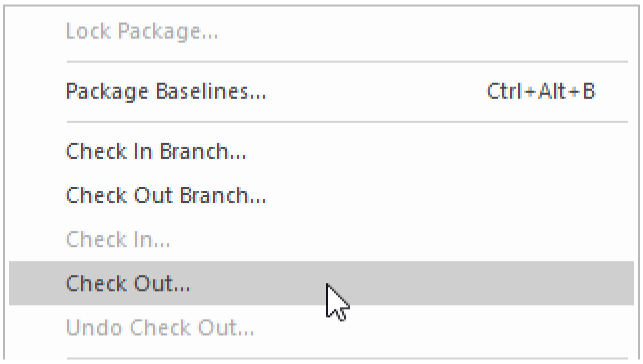
Packages under Version Control are identified in the **Browser window** by icons that indicate the current status of the Package. When a package is initially added to the Version Control System it will have a locked icon as shown in the diagram below.



Browser Window showing the Design Package with a Lock icon after Check-In.

Checking Out Packages

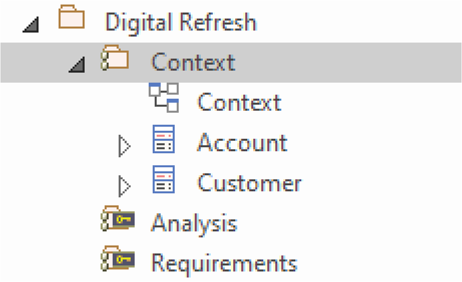
Once **Version Control** has been set up and configured you can start checking out Packages. When you want to work on a Package that is Version Controlled, you need first to Check Out that Package. This will change the Package icon in the **Browser window** from locked to unlocked. While you have the Package checked-out no other user will be able to work on the Package and they will in turn see a locked Package icon.



Context menu showing the Check Out option.

Make Modifications to Package Content

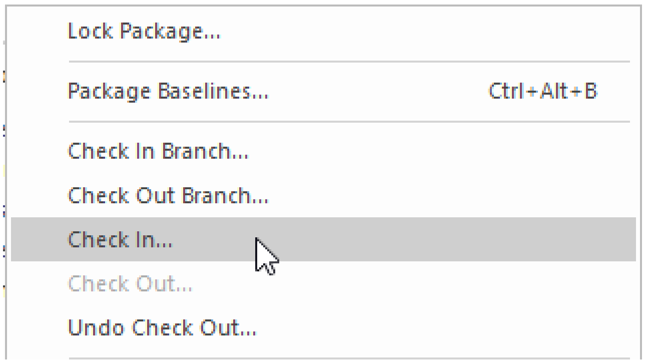
Once a Package has been checked out you are free to make modification to the content, just as you would in a repository that doesn't have any **Version Control** implemented. You will be the only one able to edit the content, as other users that attempt this will receive a message saying the Package is checked out.



Browser window showing the Context Package that has been checked out, showing the Checked-Out Package icon

Checking In Packages

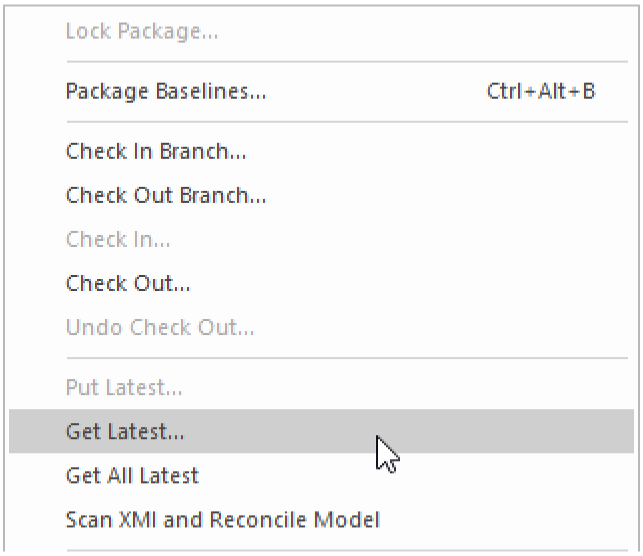
Once you have finished making changes to the contents of a Checked-Out Package, you can Check In that Package. This will change the Package icon in the **Browser window** from unlocked to locked. As soon as you Check-In the Package it will display a locked Package icon and the **Version Control** system will be updated. You can also enter a comment that describes the changes that you made while the Package was under your control, which will be useful when looking back at version changes. Once the Check-In is complete, other users will be able to Check-Out the Package.



Context menu showing the Check In option.

Getting Latest Changes

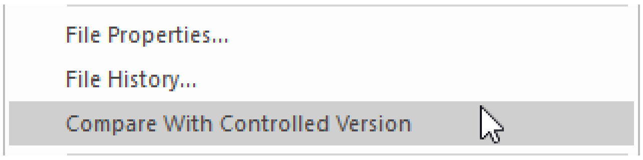
If you are working in a single shared model any changes that your colleagues have made will be available by simply reloading the package. If however you are working in private models then you will need to periodically get the latest changes that your colleagues have check in to the **Version Control** system. You can Get Latest for a single package or for the entire model depending on whether you are interest are in just one or all of the changes.



Context menu showing the Get Latest Changes option.

Comparing to Previous Version

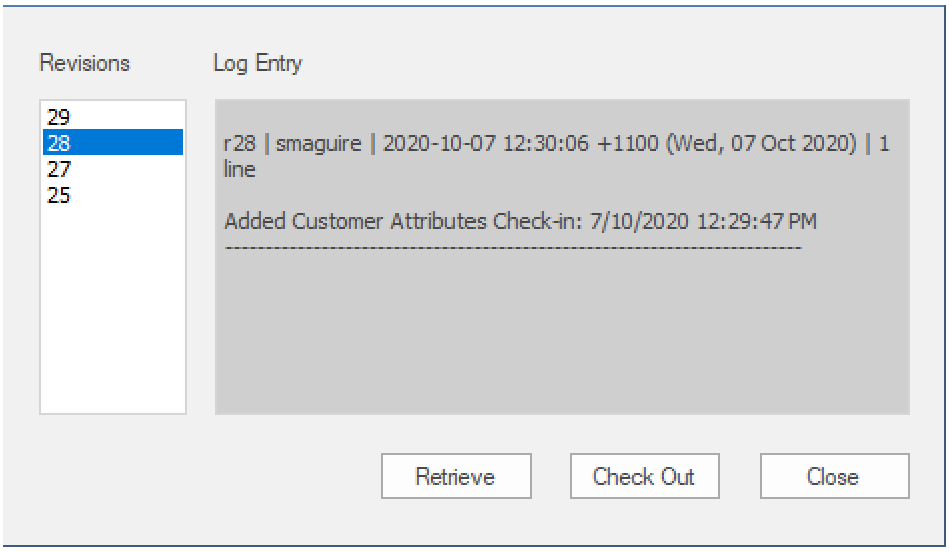
At any stage of a model's evolution you can compare the current version of a Package to the version stored in the **Version Control** system. This option launches the Comparison window that displays the differences between the model's current state and the latest revision held in the Version Control system.



Context menu showing the 'Compare with Controlled Version' option.

Reverting to a Prior Version

As your model evolves you might want to revert to a previous version of a Package. By using the 'File History' dialog you can select any prior version. By viewing the Check-In comments and the Date and Time of the Check-In the required version can be selected. The Retrieve option will restore the current model to the selected version without the need to check-out the Package.



```
r28 | smaguire | 2020-10-07 12:30:06 +1100 (Wed, 07 Oct 2020) | 1
line

Added Customer Attributes Check-in: 7/10/2020 12:29:47 PM
------------------------------------------------------------------------
```

*File Version **History window** showing a number of versions (revisions) with Comments.*

Setup Options

There are a number of options that can be chosen when setting up **Version Control** for one or more teams in your organization. There are also some fundamentals about the way that Enterprise Architect integrates with the Version Control software with respect to locking that you should review before setting up Version Control.

Version Control Locking Overview

Enterprise Architect implements **Version Control** of your model by exporting Package data from the project database to XMI Package files, which are placed under Version Control in the source-code control application. The XMI 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 here.

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.

Repository Options

The **Version Control** facility can be used in many different ways, although these roughly fall into one of four types of use as discussed in this table.

Usage

Type	Details
Single Shared model	Users share a model stored in a central 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: • Archive successive versions of your work to date • Maintain Package revision history • Recover from unwanted changes or accidental deletions, through an 'Undo' facility • Regulate access to Packages
Multiple Private models	One model is created by a single user who configures it for **Version Control**. The model file is then distributed to other users, with each user storing their own private copy of the model. Version Control is used to: • Propagate changes to the model, throughout the team • Archive successive versions of your work to date • Maintain Package revision history • Recover from unwanted changes or accidental deletions, through an 'undo' facility • Regulate access to Packages
Shared Packages	Individual users create separate models but share one or more Packages: • Users share Packages through **Version Control**
Standard Packages	A company might have a standard set of read-only Packages that are broadly shared: • Individual users retrieve Packages with the Get Package menu option

Factors to consider

Factor	Detail
System Requirements and Configuration	You apply **Version Control** through a third-party source-code control application that manages access to and stores revisions of the controlled Packages. Typically the configuration consists of: • A server component that manages a Version Control repository, and

	- Client components on the workstations, that manage local working copies of controlled files 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.
Version Control Usage	There are two main ways in which projects can be deployed: - Centralized Shared Model - Distributed Private Models **Version Control** is employed in the same way for both scenarios; however, when using Private Model deployment you have the additional facility of propagating model updates throughout the team. Version Control can also be used to share standard Packages between different projects.
Team Deployment	Consider the process of setting up a **Version Control** environment and applying Version Control to a project to be accessed by a number of users.
Version Control Basics	Enterprise Architect enforces serialized editing of Version Controlled Packages, using the lock-modify-unlock mode of operation.
Applying Version Control to Models	Using **Version Control** consists of placing individual model Packages under Version Control, rather than Version Controlling the project as a whole.
Version Control and Project Reference Data	To share changes in reference data between users in a version-controlled project deployed as multiple private models, you periodically export the reference data from the model where the changes were made, and import it into the other models maintained by the team.
Offline Version Control	You can prevent the system from attempting to make any **Version Control** connections by choosing to Work Offline 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.

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 in that:

- 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; if the model is read-only, the system displays an error message and fails to load the model

Version Controlling Packages in your Model

To overcome these limitations, 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 XMI files being exported, as the parent Packages' XMI 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.

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
Checking Out	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.
Get All Latest	If you select the 'Get All Latest' option from the **Version Control** menu, any new stubs are populated from the Version Control system.
Importing Models	You can populate a large and complex model, by 'getting' only the root Packages, then using 'Get All Latest' to recursively iterate through the attached and nested Packages. This is a powerful and efficient means of managing your project and simplifies handling very large models, even in a distributed environment. The 'Import a Model Branch' option combines the procedure steps into a single operation.

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)

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 XMI 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 XMI file and apply Version Control to that file using your Version Control software external to Enterprise Architect.

Version Control and Teams

This is a summary of the process of setting up a **Version Control** environment and applying Version Control to a project to be accessed by a number of users.

Version Control - Process Overview

Step	Action
1	Install your **Version Control** product.
2	Create a **Version Control** repository.
3	Create a **Version Control** project to be used with your Enterprise Architect project.
4	Check-out a working copy of the **Version Control** project (a module, project or folder within the Version Control system) into a local folder. 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.
5	Before attempting to access the new **Version Control** environment from within Enterprise Architect, you should first verify that it functions correctly when used outside of Enterprise Architect.
6	Within Enterprise Architect, define a **Version Control** configuration to provide access to the working copy files. 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. 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. • In Shared Model deployment, all users connect to a single instance of the model database, so the model is shared automatically • 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 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 perhaps choosing the Version Control project associated with this Enterprise Architect project. Once this has been done, the Version Controlled Packages that already exist in the model are ready for use.
7	Configure Packages within the Enterprise Architect model for **Version Control**. That is, apply Version Control to individual Packages.
8	Check-out and check-in Packages as required.

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

Offline Version Control

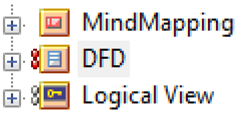
When loading a model that uses **Version Control**, Enterprise Architect normally initializes a connection to the Version Control system for each Version Control configuration defined in the model. If Enterprise Architect is unable to connect a Version Control configuration for any reason, it displays 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 before loading your model.

Access

Ribbon	Configure > **Version Control** > Work Offline

Working Offline

Concept	Discussion
Choosing to Work Offline	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 disconnected 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. You can switch between working offline and working online at any time, either before or after a model is loaded, by toggling the 'Work Offline' menu option. Enterprise Architect disconnects or reconnects Version Control (depending on the connection availability) according to your selection.
Using Version Control Whilst Disconnected From the Version Control Server	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. 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.
Offline Check Out	You can 'check-out' and edit a Version Controlled Package even when your machine is disconnected from the **Version Control** server. In this example, the colored 'figure 8' icon for DFD indicates that you have checked it out whilst off line.  MindMapping DFD Logical View (The gray 'figure 8' icon shown against Logical View indicates that you have

	checked out a Version Controlled Package on line.) 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 merge changes to an XMI 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 first-saved set of changes is lost.
Checking In a Package That Was Checked Out Offline	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 remains unchanged from when you last updated your local copy, Enterprise Architect exports and checks in your Package without further prompting. On the other hand, if the repository now contains a file that has changed since you last updated your local copy, checking in your Package would overwrite 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 'Undo Check Out' 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. You can use the 'File Properties' 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.
Update Before You Disconnect	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. 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 before you disconnect. The 'Get All Latest' option makes this a simple task.

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.

Contacts

User Type	Contact via
Registered users	https://sparxsystems.com/registered/reg_support.html
Trial users	support@sparxsystems.com

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
Version Control Server	The server component maintains the controlled files in their many revisions, in a central repository. The server component is usually located on a server machine that is accessible to all team members who are using **Version Control**.
Server Configuration	The steps for configuring a **Version Control** server are, broadly: • Install the software • Create a repository • Create Version Control projects (or modules or folders for use with specific projects) • Configure user IDs and passwords For details on configuring any particular Version Control server, consult the appropriate documentation provided with the Version Control product.
Version Control Client	The client component manages the working copies of the controlled files, submitting files to or retrieving files from the server as required. A **Version Control** client must be installed on every machine on which you run Enterprise Architect and want to access your Version Control system.
Client Configuration	The steps for configuring a **Version Control** client are, broadly: • Install the software • Create a new directory for use as a local working copy folder • Log in to the Version Control server • Associate the working copy folder with its corresponding server repository folder 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.

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
Subversion	Subversion is free, open source software. 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, but requires Subversion's Windows-based command line client for integration. There are many graphical user interface clients available for use with Subversion, such as TortoiseSVN; this type of client cannot be used directly for integration with Enterprise Architect, but can be very useful in preparing a working Subversion environment for use by Enterprise Architect. Binary Packages are available for download from: • http://subversion.apache.org/packages.html Subversion documentation is available from: • http://svnbook.red-bean.com/nightly/en/index.html
Concurrent Versions System (CVS)	CVS is free, open source software. 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, but requires CVS's Windows-based command line client for integration. There are many graphical user interface clients available for use with CVS, such as TortoiseCVS; this type of client cannot be used directly for integration with Enterprise Architect, but can be very useful in preparing a working CVS environment for use by Enterprise Architect. The software is available for download from: • http://www.nongnu.org/cvs/ CVS documentation is available from: • http://cvsbook.red-bean.com/cvsbook.html
Microsoft Team Foundation Server	Enterprise Architect is able to use either the: • Command line client for TFS, or • MS TFS-SCC client Your choice of client affects how you specify your **Version Control** Configuration. MS TFS-SCC clients are available for download from Microsoft's web pages. (Run an Internet search for 'TFS MSSCCI'.)
Common Source Code Control (SCC)-compatible products	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. These products are SCC-compatible and are known to successfully integrate with

	Enterprise Architect: • Accurev • ClearCase • MS Visual Source Safe • MS TFS-SCC • MKS Source Integrity • Perforce • Source Offsite • Snapshot CM Products that do not appear in this list should still integrate successfully with Enterprise Architect, if there is a client available for that product that complies with the MS SCC API specification.

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
Install server components	Executable files for Subversion can be obtained from the Apache Software Foundation. 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. VisualSVN is a package that can greatly simplify the installation, configuration and management of your Subversion server.
Create a repository	Please consult the official Subversion documentation.
Create Subversion users	Please consult the official Subversion documentation.
Create a new repository sub-tree	It is good practice to create a new repository sub-tree in Subversion for each 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. TortoiseSVN can greatly simplify the process of creating new repository sub-trees.
Install client components	Executable files for Subversion can be obtained from the Apache Software Foundation.
Create a working copy folder	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 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. TortoiseSVN can greatly simplify the initial check out of a working copy folder.
Setting up Subversion under Wine/CrossOver	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.

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

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
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	Open a Windows command prompt, navigate to *tempDir* and issue the Subversion command *import*. For example: C:\Documents and Settings\user> cd \tempDir C:\tempDir> svn import . https://host.example.com:8443/repos/ --message "Repository Initialization"
3	On your PC, delete the temporary directory structure (tempDir) 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 can also be performed using TortoiseSVN's Repository Browser, which provides commands for simply creating new folders directly in the repository

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
1	Choose or create a suitable directory on your system in which to create your Subversion working copy.

2	Open a command prompt window and navigate to the directory you have selected. For example:
	C:\> mkdir mySVNWorkSpace
	C:\> mkdir mySVNWorkSpace/myEAModelName
	C:\> cd mySVNWorkSpace/myEAModelName

3	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: C:\> svn checkout "https://myserver:8443/svn/repos_folder "C:\mySVNWorkSpace/myEAModelName" --username myUserName --password myPassword (By specifying your Subversion username and password, you ensure that they are correctly cached by Subversion and available for use by Enterprise Architect.) 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 the **P key** to permanently accept the certificate. 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.
4	Verify that your Subversion environment functions correctly.

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 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

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
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: C:\> cd mySVNWorkSpace
3	Create a test file, such as *Test.txt*, containing the text *Subversion Test*. For example: C:\> echo Subversion Test > Test.txt
4	Execute each of these Subversion commands: • svn add Test.txt • svn commit -m"Commit comment" Test.txt • svn update Test.txt • svn lock Test.txt • Use your preferred editor to modify the file and save the changes • svn commit -m"Second commit comment" Test.txt The commands should execute without any errors and without prompting the user for any extra input.

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

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 macOS 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.

Preparing a Subversion Environment Under Wine

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
1	Start Enterprise Architect. You do not have to load a project at this point.
2	From the 'Start' ribbon, select 'Desktop > Preferences > Toolbars... The 'Customize' dialog is opened. Select the 'Tools' tab, then click on the 🗋 icon. A new, blank entry is opened on the 'Tools' tab of the 'Customize' dialog.
3	Define the new menu item entry: • In the newly-opened 'Menu contents' field, type the name 'Wine Console' • In the 'Command' field, type 'wineconsole' • In the 'Arguments' field, type 'cmd' • Leave the 'Initial directory' field blank
4	Click on the **Close button**. The 'Customize' dialog closes.
5	Select the ribbon option: 'Configure > User Tools > Wine Console'. A Wine console window opens.
6	Type 'C:' and press the **Enter key**. The Wine console switches to the C: drive.
7	Issue the command to install your Subversion client. For example: C:\>/Installers/Subversion-client-1.9.5.win32.exe To install from a .msi file, use Wine's msiexec utility. For example: C:\>msiexec /i "Installers\Slik-Subversion-1.9.7-win32.msi" Installation of the Subversion command line client begins.
8	Create a folder to serve as the working copy folder to be used by Enterprise Architect. For example: C:\>mkdir C:\VC_workspaces\SVN_workcopy
9	Issue the command to perform the initial checkout from the Subversion repository, specifying the repository folder, working copy folder, username and password. For example: C:\>svn checkout "https://myServer:8443/svn/repos_folder" "C:\VC_workspaces\SVN_workcopy "

	--username myUserName --password myPassword (After specifying your Subversion username and password, they are correctly cached by Subversion and are available for use by Enterprise Architect.) 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 the **P key** to permanently accept the certificate. 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.
10	Type 'Exit' and press the **Enter key**. The Wine console window closes. You are now ready to load a project in Enterprise Architect and apply **Version Control** to it, using the normal Windows-based procedures.

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

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

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
Obtain and install TFS	Enterprise Architect uses the TFS command line client to integrate TFS **Version Control**. The TFS command line client is normally available as part of your Visual Studio installation.
Choose a TFS project	It is good practice to create a new TFS project, or least a new Source Control Folder 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 model. For further information, please consult your TFS product documentation.
Create a TFS workspace	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. TFS 2012 and VS 2012 (and later versions) feature a new type of workspace called 'local' workspaces. Do not attempt to use TFS 'local' workspaces with Enterprise Architect. You must use only 'server' workspaces for Enterprise Architect Version Control, as 'local' workspaces do not support the application of checkout **locks** to files. Enterprise Architect relies on the presence of checkout locks to ensure that Packages can only be checked out exclusively and that a given Package is not already checked-out in some other project (for instance, in a Private Model deployment). This is necessary because it is not practical to merge the XMI Package files that Enterprise Architect uses for Version Control. A single TFS workspace can 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 Package files in a folder that is separate from other Artifacts, such as source code files, creating a separate work space to use just for your Package files, or creating and mapping a separate folder for Package files within an existing workspace.
Configure exclusive check-outs	The XMI format files used for the **Version Control** of Enterprise Architect's Packages can not be merged as if they were 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 XML files.
Verify the TFS Workspace	Enterprise Architect uses the TFS command line client to checkin and checkout files from the TFS repository. After creating the TFS workspace, it is important to verify that the command line client can be used to add, check-in and check-out files residing in the working copy folder that is mapped through this workspace.

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

TFS Workspaces

In order to use TFS to provide **Version Control** of the Packages in a model, you 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 have TFS 2013 open and a TFS Team Project already exists for you to use in Version Control of the Packages of your Enterprise Architect project. You can create the TFS workspace through MS Visual Studio.

TFS 2013 and later versions support the use of Local workspaces. However, Local workspaces do not support the application of checkout **locks**. For this reason, Sparx Systems strongly advises against the use of Local workspaces when Version Controlling your Enterprise Architect Package files. Using Local workspaces carries a high risk of merge conflicts when checking in, which would almost certainly result in loss of data or a corrupted model database.

Map a local folder to a TFS Source Control Folder

Step	Action
1	Connect to your TFS server. From the TFS main menu, choose 'View | Team Explorer'. In the Team Explorer toolbar, click on the **Connect to Team Projects button** (fourth from left, with a power plug icon). The 'Team Explorer Connect' page displays.
2	Click on the link 'Select Team Projects...' The 'Connect to Team Foundation Server' dialog displays.
3	Click on the appropriate Team Foundation Server, Team Project Collection and Team Project, then click on the Connect button.
4	Click on the Home page of the Team Explorer, and then on the **Source Control Explorer** button. The Source Control Explorer view displays.
5	In the **Source Control Explorer**, click on the **drop-down arrow** in the 'Workspace' field in the toolbar, then click on 'Workspaces' at the bottom of the list. The 'Manage Workspaces' dialog displays.
6	Click on the **Add button**. The 'Add Workspace' dialog displays.
7	Click on the **Advanced button**. A number of new fields display on the dialog.
8	Type in an appropriate name for the new workspace and, if required, type in a comment.
9	Set the 'Location' field to 'Server'. This setting is important. This setting is not available in TFS 2010 and earlier releases, where all workspaces are Server based.
10	Click in the Source Control Folder column, then click on the **Browse button** to select a Source Control Folder. Select the Source Control folder (in the Team Project) to use for controlling Enterprise Architect

	Packages.
11	Click on the **Browse button** in the 'Local Folder' 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 **OK button**. The new workspace is created and saved. The 'Add Workspace' dialog closes.
13	Click on the **OK 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

TFS Exclusive Check Outs

The XMI 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
1	Using Visual Studio, from the main menu select 'View | Team Explorer'.
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 'Team Foundation Server Settings | Source Control File Types'.
3	Select the entry for XML files (or create an entry if necessary) then click on the **Edit button**.
4	Deselect the 'Enable file merging and multiple check out' checkbox.

Verify the TFS Workspace

Enterprise Architect will be using the TFS command line client to commit file updates from the local working copy folder into the TFS repository. After creating the TFS workspace that maps your local working copy folder to a TFS repository folder, it is good practice to verify that the TFS workspace functions correctly before attempting to use it with Enterprise Architect. You need to be able to add files to TFS, place a check-out lock on the files and check-in changes to those files.

Verify correct operation of a TFS Workspace

Step	Action
1	Use Windows to open a command prompt window.
2	Select the directory you specified as the local working copy folder when creating the TFS workspace. For example: C:\> cd myTFSWorkingCopyFolder
3	Create a test file, such as *Test.xml*, containing the text '*my TFS Test*'. For example: C:\> echo my TFS Test > Test.xml (The filename extension '.xml' is used because the TFS server should have been configured to allow exclusive checkouts for XML files.)
4	Using the command line client, execute each of these TFS commands: • tf.exe add Test.xml • tf.exe checkin Test.xml /override:"Manual check in." /comment:"a message" • tf.exe checkout /lock:checkout Test.xml Use your preferred editor to modify the file and save the changes. • tf.exe checkin Test.xml /override:"Manual check in." /comment:"a message" • tf.exe get . /recursive

Notes

- All of the commands in step 4 should execute without any errors; in particular, you should be able to apply a checkout lock when checking out the file
- If the checkout command reports any errors, you might have to reconfigure the TFS workspace as a server based workspace, or you might have to configure the TFS Server to use 'exclusive checkouts' for XML files

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
Install server components	Executable files for CVS can be obtained from March Hare Software. 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.
Create a repository	Please consult the official CVS documentation.
Create CVS users	Please consult the official CVS documentation.
Create a new repository module	It is recommended good practice to create a new repository module in CVS for each 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. A repository module represents a project, or a set of related files in the repository. TortoiseCVS can greatly simplify the process of creating new repository sub-trees.
Install client components	Executable files for CVS can be obtained from March Hare Software. Enterprise Architect is a Windows based application - it requires a Windows based CVS command line client for integration.
Create a working copy folder	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 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 folder and its contents, to create your local working copy. TortoiseCVS can greatly simplify the initial check out of a working copy folder.
Setting up CVS under Wine/CrossOver	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

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
1	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. Your administrator must create a username and password for you before you can make a connection.
2	Select or create a suitable directory to use as your CVS working copy directory.

3	Open a command prompt window and navigate to your CVS working copy directory. For example:
	C:\> mkdir myCVSWorkSpace
	C:\> cd myCVSWorkSpace

4	Log in to the remote CVS repository. For example: C:\myCVSWorkSpace> cvs -d:pserver:myUserID@ServerName:/reposPath login Replace *myUserID* with your CVS username, replace *ServerName* with the name of your CVS server and replace *reposPath* with the path to the repository on the server. A prompt for a password displays.
5	Enter your password. You are logged in to the CVS server.
6	Perform the initial checkout of the CVS repository module, into the local working copy directory. For example: C:\myCVSWorkSpace> cvs -d:pserver:myUserID@ServerName:/reposPath checkout moduleName (Replace *moduleName* with the name of the repository module that you want to check out.) A subdirectory is created in your current working directory, with the same name as the module being checked out. Any files already existing in the repository module are downloaded to the working copy folder as working copy files.
7	Verify that your CVS environment functions correctly.

Notes

- Much of the process can also be performed (more simply) using the TortoiseCVS command 'Make New Module'

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
1	Use Windows to open a command prompt window.
2	Select the directory you specified as the working copy in the cvs checkout command, when preparing the CVS workspace. For example: C:\> cd myCVSWorkSpace
3	Create a test file, such as *Test.txt*, containing the text *CVS Test*. For example: C:\> echo CVS Test > Test.txt
4	Execute these CVS commands: • cvs add Test.txt • cvs commit -m"Commit comment" Test.txt • cvs update Test.txt • cvs edit Test.txt • cvs editors Test.txt The commands should execute without any errors and without prompting the user for any extra input. The editors command should produce output that resembles this: Test1.txt myUserID Tue Aug 9 10:08:43 2009 GMT myComputer C:\myCVSWorkSpace\moduleName
5	Take note of the userID that follows the filename. Enterprise Architect must find and use this user ID when you create your **Version Control** configuration.

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

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

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
Install and configure your chosen Version Control product	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. After installing the Version Control software, the administrator should also create Version Control user IDs for all users who require access to Version Control. For further information, consult the product documentation for your particular Version Control product.
Create a new SCC project	It is good practice to create a new SCC **Version Control** project, or least a new folder within a project, for each Enterprise Architect project being added to Version Control with SCC. 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 model. For further information, consult the product documentation for your particular Version Control product.
Configure your SCC project to support exclusive check-outs for .XML files	The XMI-format files used for the **Version Control** of Enterprise Architect Packages can not be merged in the same way as 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.
Create a local working copy folder	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. Your Version Control product provides some means by which you associate a working copy folder with a repository folder. For further information, consult the documentation for your particular Version Control product.

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

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
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 'Unique ID' drop-down list.
3	The existing SCC configuration is the initial value, represented by SCC-XXXXX; 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 'Unique ID' drop-down list (if one exists).
5	After you have assigned the unique ID, click on the **OK button** to load the model.

Version Control Configuration

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

Ribbon	Configure > **Version Control** > Project-VC
Context Menu	Right-click on Package > **Package Control** > **Version Control** Settings

Define Version Control Configuration

Step	Action
1	On the '**Version Control** Settings' dialog, click on the **New button**.
2	In the 'Unique ID' field, type a suitable configuration name.
3	Select the Type of **Version Control** product you are connecting to, by clicking on the corresponding radio button.
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 **Save button**. The new configuration is added to the Defined Configurations 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 **Close 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

Version Control Settings

Defining **Version Control** settings within Enterprise Architect should only be performed after you have successfully created the Version Control environment that you intend to use to control your Package files. If you have not yet done so, please refer to the Help topic *Applying Version Control in a Team Environment* for a summary of what is required.

To set up a Version Control configuration on your model, or update 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

Ribbon	Configure > **Version Control** > Project-VC
Context Menu	Right-click on Package > **Package Control** > **Version Control** Settings

Configuration Options

Option	Action
This model is private	Select to specify that this model database is to be accessed by just a single user (Private Model). Leave unselected (the default) or deselect to specify that the database is to be accessed by multiple concurrent users (Shared Model). If in doubt, use the default setting.
Save nested Version Controlled Packages to stubs only	Select to specify that the exported XMI 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 XMI file will contain the full content of nested Version Controlled child Packages.
For all Packages, create placeholders for external references	Select to force all XMI 1.1 imports across the model to exclude incoming relationships and instead create external references. If the 'Create placeholders for missing External References during XMI 1.1/2.1 Import' checkbox is not selected in the XML Specifications options for a user, this field overrides that setting. The *XML Specifications* Help topic provides an example of the benefits of selecting this option.
Unique ID	Specify a name that uniquely identifies the configuration. Either: • Type a name to identify a new configuration, or • Click on the **drop-down arrow** and select the name of a configuration previously defined in a different project (if any exist)
Type	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 SCC for: • MS Visual Source Safe • Rational Clear Case • Perforce • AccuRev • Any other SCC-compatible clients For any other product that you are using, select the type that matches the product - CVS, Subversion or TFS.
New	Click on this button to clear the fields and create a new **Version Control** configuration.
Save	Click on this button to save the details of a new or updated configuration.
Delete	Click on an entry in the Defined Configurations list and click this button to remove the definition of the selected configuration from this model.
Defined Configurations	Review a list of configurations that are in use in the current model.
In future, do not prompt for incomplete configurations	Select to specify that the user is not prompted to complete the definition of configurations that are not fully specified (the default). Deselect to prompt the user to complete configurations that are not fully defined.
Close	Close the '**Version Control** Settings' dialog.
Help	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 have 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

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

Ribbon	Configure > **Version Control** > Project-VC : Type: SCC
Context Menu	Right-click on Package > **Package Control** > **Version Control** Settings: Type: SCC

Settings

Option	Action
Local Project Path	Displays the full path of the folder that contains the local (working) copies of the XMI Package files. This field is read-only; its value can only be set through use of the **Select Path button**.
Select Path	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. • 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 **OK button** • 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 • 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 **Save button** on the 'Version Control Settings' dialog
Current User	Displays the user name used to log on to the **Version Control** system that is

	accessed through this configuration. This field is read-only; the value it displays is retrieved from the SCC client.
SCC Provider	Displays the name of the SCC provider. This field is read-only; the value it displays is retrieved from the SCC client.
SCC Project	Displays the name of the SCC Project that this configuration attaches to. This field is read-only; the value it displays is retrieved from the SCC client.

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

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

Ribbon	Configure > **Version Control** > Project-VC : Type: CVS
Context Menu	Right-click on Package > **Package Control** > **Version Control** Settings: Type: CVS

Settings

Option	Action
Working Copy Path	Displays the full path of the folder that contains the local (working) copies of the XMI Package files. This field is read-only; its value can only be set through use of the **Select Path button**.
Select Path	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.
Current User	This field is read-only; its value is retrieved from a file named CVS\Root, located in the folder selected using the **Select Path button**.
CVS Exe Path	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 **Select Path button**.
Select Path	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.
VC Time-Out Value	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 Time-out 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

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

Ribbon	Configure > **Version Control** > Project-VC : Type: Subversion
Context Menu	Right-click on Package > **Package Control** > **Version Control** Settings: Type: Subversion

Settings

Option	Action
Working Copy Path	Displays the full path of the folder that contains the local (working) copies of the XMI Package files. This field is read-only; its value can only be set through use of the **Select Path button**.
Select Path	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.
Subversion Exe Path	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 **Select Path button**.
Select Path	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.
VC Time-Out Value	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 Time-out 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

- 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

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

Ribbon	Configure > **Version Control** > Project-VC: Type: TFS
Context Menu	Right-click on Package > **Package Control** > **Version Control** Settings: Type: TFS

Settings

Option	Action
Working Copy Path	Displays the full path of the folder that contains the local (working) copies of the XMI Package files. This field is read-only; its value can only be set through use of the associated **Select Path button**.
Select Path	Click on this button to choose the Working Copy Path. The file browser dialog opens, through which you navigate through the file system to the appropriate folder.
Server Name	Displays the name of the TFS Server that is associated with the working copy folder specified in the 'Working Copy Path' field. This field is read-only; Enterprise Architect retrieves the value it displays by querying the TFS client.
Workspace Name	Displays the name of the TFS Workspace that is associated with the working copy folder specified in the 'Working Copy Path' field. This field is read-only; Enterprise Architect retrieves the value it displays by querying the TFS client.
User Name	Type in the user name with which to log into the TFS Server.
Password	Type in the password with which to log into the TFS Server.
Display Name	TFS 2013 and later releases use a Display Name to report on who owns a checkout lock on a file. TFS 2010 uses AccountID when reporting lock owners and so does not require a Display Name. If using TFS 2013, type in your TFS Display Name as shown in the User column of

	the Visual Studio **Source Control Explorer** when you checkout a file.
Checkout Locks Required	This checkbox defaults to selected. TFS 2013 supports the use of Local workspaces, but these do not support checkout **locks**. If you want to use TFS Local workspaces (not recommended), you must deselect this checkbox. Otherwise, leave the checkbox selected. It is recommended that all workspaces used for Version Controlling Enterprise Architect Package files are set as 'Server' workspaces.
TFS Exe Path	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 **Select Path button**.
Select Path	Click on this button to specify the path to the TFS command line client. The file browser dialog opens, through which you navigate through the file system to the appropriate folder.
VC Time-Out Value	Type in the number of seconds that Enterprise Architect should wait for a TFS command to complete; if a command does not complete within this allowed time, the system displays a Time-out 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 a path 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 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

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

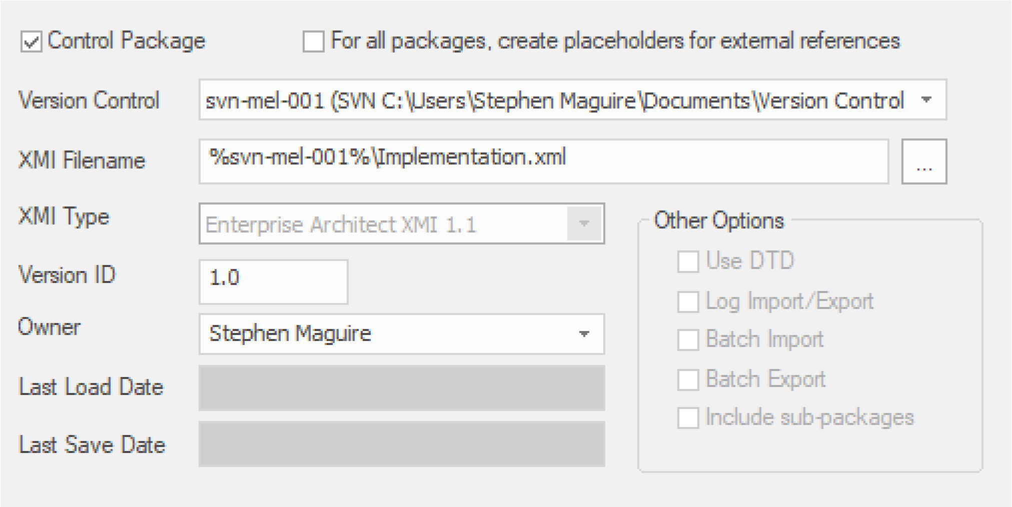
Ribbon	Configure > **Version Control** > Project-VC
Context Menu	**Browser window** | right-click on Package | **Package Control** | **Version Control** Settings

Select existing configuration

Step	Action
1	On the '**Version Control** Settings' dialog, click on the **New button**. The various fields on the dialog are cleared, ready for data entry.
2	In the 'Unique ID' 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.
3	Click on the **Save button**. The configuration details are saved and are ready for use in the current project.

Applying to Packages

Version Control in Enterprise Architect operates at a Package level. Any Package in the repository can be put under Version Control and assigned to any Version Control Configuration. This means that different Packages can be controlled by different Version Control systems.



☑ Control Package

□ For all packages, create placeholders for external references

Version Control: svn-mel-001 (SVN C:\Users\Stephen Maguire\Documents\Version Control

XMI Filename: %svn-mel-001%\Implementation.xml

XMI Type: Enterprise Architect XMI 1.1

Version ID: 1.0

Owner: Stephen Maguire

Last Load Date:

Last Save Date:

Other Options

□ Use DTD

□ Log Import/Export

□ Batch Import

□ Batch Export

□ Include sub-packages

The **Package Control** Options window showing the setting for the selected Package.

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 XMI 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

Ribbon	Configure > **Version Control** > Package-VC
Keyboard Shortcuts	**Ctrl+Alt+P**

Apply Version Control to a single Package

Step	Action
1	Click on the 'Control Package' checkbox to select it, indicating that this Package is to be controlled.
2	Click on the '**Version Control**' **drop-down arrow** and select the Version Control configuration to be used to control this Package.
3	The 'XMI Filename' field displays a default filename for the Package export file, based on the Package name. Optionally, modify the filename. Either: • Type a new name, or • Click on the ... button to open a file selection dialog The target file must be located within the working copy folder of the selected **Version Control** configuration, or one of its sub-folders.
4	The 'Version ID' field defaults to '1.0'. Optionally, change this to a different version number.
5	The 'Owner' field defaults to your user name. Optionally, type or select the name of the user who actually owns the Package.
6	Click on the **OK button**. The 'Add Package to **Version Control**' dialog displays.
7	Optionally, clear the 'Keep checked out' checkbox.

	After applying **Version Control**, the Package either remains checked-out for editing, or is checked-in and locked against editing, depending on this setting.
8	Click on the **OK button**. The 'Add Comment' dialog displays.
9	Optionally, add any further comments to the default comment. Enterprise Architect provides a default comment that includes the current date & time.
10	Click on the **OK button**. The current Package is exported to the nominated XMI file, which is then committed to **Version Control**. The Package icon in the **Browser window** is updated to reflect the Package's Version Control status.

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**

Browser Window Indicators

Packages under **Version Control** are identified in the **Browser window** by icons that indicate the current status of the Package.

Indicators

Icon	Indicates that
	This Package is Version Controlled and not checked out to you. You cannot edit the Package (until you check out the Package yourself).
	This Package is Version Controlled and checked out to you. You can edit the Package.
	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.
	This Package is controlled and is represented by an XMI file in the file management system, but it is not under **Version Control**. You can edit this Package.

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 **Browser window**, and all of the Packages contained within it.

Access

Context Menu	**Browser window** | right-click on Package | **Package Control** | Add Branch to **Version Control**

Apply Version Control to all Packages within a selected model branch

Step	Action
1	On the 'Apply VC to Branch' dialog, click on the **drop-down arrow** in the '**Version Control** Configuration' field and select the configuration to use.
2	Optionally, tick the checkbox 'Export as Model Branch'. Once the **Version Control** operation is complete a Model Branch file (.EAB file) is created for this branch.
3	Click on the **OK button**. 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 XMI files, based on the Package GUIDs.

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

Fundamental Usage

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. The fundamental Version Control functions are outlined in this table.

Fundamental Functions

Facility	Detail
Define Version Control Settings	**Version Control** Configurations are used by Enterprise Architect to communicate with your Version Control system. You define one or more Version Control configurations in your project and then use those configurations to control the Packages in your project.
Configure a Package	To put a Package under **Version Control** you mark the Package as a controlled Package, specify the Version Control configuration to control it, and associate an XMI file with the Package.
Check Out a Package	Checks out the Version Controlled Package currently selected in the **Browser window**, so that you can update modeling objects within it.
Undo Check Out of a Package	Reverses the check-out of a Package, discarding any modifications that have been made by restoring the Package content to the latest revision held in **Version Control**.
Check In a Package	Checks in the Package currently selected in the **Browser window**. Checking-in updates the reference version of a Package or group of Packages in the model.
Check In a Model Branch	Checks in all Packages involved in a particular unit of work, as a single operation. Checking-in updates the reference version of a Package or group of Packages in the model.
Check Out a Model Branch	Checks out all Packages within a selected model branch as a single operation, so that you can update modeling objects within them.
Import a Package From Version Control	Retrieves Packages from **Version Control** that have been created by other users, or by you in another model, and imports them into your current model.
Apply Version Control to a Model Branch	Applies **Version Control** to all Packages within the selected model branch, in a single operation. In this context, a model branch is a Package in the **Browser window**, and all of the Packages contained within it.
View Package Revision History	Displays the change history of Version Controlled Packages. You can also check out a prior revision of the Package for editing, effectively rolling-back to a prior revision of the Package.

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

Context Menu	Right-click on Version Controlled Package > **Package Control**

Options

Option	Action
Check In Branch	Check-in Packages contained in the currently selected model branch (that is, the selected Package and all of its child Packages). 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. You can also choose to keep the Packages checked-out after committing a new revision to **Version Control**.
Check Out Branch	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.
Check In	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.
Check Out	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).
Undo Check Out	Restore the selected Package to the latest revision in the **Version Control**

	repository and lock the Package against further editing.
Put Latest	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.
Get Latest	Synchronize the currently selected Package with the latest revision from the **Version Control** repository. Available only for Packages that are checked in.
Get All Latest	Updates 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 fixes any missing cross-references by comparing the Package with its XMI 1.1 file. If the cross-reference information in the XMI does not match the model, the system updates the model with the information from the XMI and records this update in the **System Output** window. You can roll back such updates by selecting the entry in the System Output window and using the context menu option 'Rollback Update' (or 'Rollback Selected Updates' if multiple entries are selected). • Closing the model clears the entries in the System Output window • An entry in the System Output window is also cleared as and when you roll-back the update for it
Scan XMI and Reconcile Model	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. This function is useful in team environments where each user maintains their own private copy of the model database (that is, 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. 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 **System Output** window. You can roll back such updates by right-clicking on the entry in the System Output window and selecting the 'Rollback Update' option (or 'Rollback Selected Updates' if multiple entries are selected). Closing the model clears the entries in the System Output window; an entry in the window is also cleared as and when you roll-back the update for it. This functionality is invoked automatically as part of the 'Get All Latest' operation. 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 'Scan XMI and Reconcile Model' from time to time, following the re-importation of controlled Packages - for example, after using 'Get Latest' to update a number of Packages - or after performing a number of Package check-outs. As a general rule, avoid running 'Scan XMI and Reconcile Model' while you have

	uncommitted changes in your model; generally, you: • Check-out a number of Packages • Invoke 'Scan XMI and Reconcile Model' • Make your modifications • Commit any outstanding changes before you check-out more Packages and run 'Scan XMI and Reconcile Model' again
File Properties	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.
File History	Display change history information for the currently selected Package. Revert to or check-out a prior revision of the Package.
Compare with Controlled Version	Compare the currently selected Package with the latest revision of its associated XMI file retrieved from **Version Control**.
Add Branch to Version Control	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 **Browser window**, and all of the Packages contained within it.
Export as Model Branch	Export a newly created model branch from your own private copy of a model.
Import a Model Branch	Retrieve a model branch and import it into either the source model or another model.
Get Package	Access Packages in the **Version Control** repository that are not currently available in your model.
Re-synch Status With VC Provider	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 XMI 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.
Version Control Settings	Display the '**Version Control** Settings' dialog.

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

Check Out a Package

When you need to work on a Version Controlled Package, you check it out. The local XMI 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

Context Menu	Right-click on Package > **Package Control** > Check Out
Keyboard Shortcuts	**Ctrl+Alt+L**

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.

These 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 **Browser window** 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' context menu option; the dialog displays and you can change the settings

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

Context Menu	Right-click on 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 **Browser window** is updated to reflect the change in the Package's Version Control status.

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

Context Menu	Right-click on Package > **Package Control** > Check In
Keyboard Shortcuts	**Ctrl+Alt+S**

Check in a single Package

Step	Action
1	The selected Package is exported and the 'Add Comment' dialog displays. A default comment is provided that contains the current date and time. Optionally, modify the default check-in comment
2	Click OK. The Package file is checked-in to **Version Control** and the Package icon is updated to reflect the change in Version Control status.

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

Context Menu	Right-click on Package > **Package Control** > Check Out Branch

Check out a sub-tree of model Packages

Step	Action
1	The selected root-node Package and all of its contained sub-Packages are recursively checked out. Any Packages that cannot be checked-out are listed in a message box, with a brief description of the problem; for example: *The Package is already checked out by user 'Fred'*.
2	When Project Security is enabled in Lock to Edit mode, Enterprise Architect prompts you to apply a User Lock throughout the selected model branch before proceeding.

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

Context Menu	Right-click on Package > **Package Control** > Check In Branch

Check in Packages from within a model branch

Step	Action
1	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. Optionally: • Click an individual Package to select just that Package • **Ctrl+click** on an individual Package to add or remove it from the selection • **Shift+click** on a range of Packages to select them • Click on the **All button** to select all listed Packages • Click on the **None button** to clear the selection
2	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 'Keep packages checked-out after committing new revision' checkbox.
3	Click on the **OK button**. 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.
4	Optionally, modify the default check-in comment.
5	Click on the **OK button**. 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.

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 **Browser window**.

Access

Context Menu	Right-click on Package > **Package Control** > Get Latest

Update Package to latest revision

The local XMI file associated with the Package is updated to the latest revision from **Version Control**. The XMI 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.

These 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 **Browser window** 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' context menu option; the dialog displays and you can change the settings

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

Context Menu	Right-click on Package > **Package Control** > Get All Latest

Update all Packages in project to latest revision retrieved from Version Control

All the local XMI 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 these 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 XMI 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 information in the model database is always the same as, or ahead of, what is committed into Version Control; in this situation, the Get All Latest command will simply refresh your view of the model database, by reloading diagrams or reloading Package content in the **Browser window**

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

Context Menu	Right-click on Package > **Package Control** > File History

Review change history of a version-controlled Package

Step	Action
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 'Revisions' 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.
2	On the 'File Version History' dialog, you can optionally click on either: • Check Out - 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 or • Retrieve - 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

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

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

Context Menu	Right-click on Package > **Package Control** > File History

Review change history of a version-controlled Package (SCC client)

Step	Action
1	The change history mechanism offered by the third party SCC provider displays. To import a prior revision of the Package into your model, use the 'SCC History' dialog to retrieve the revision, then close the dialog.
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	Optionally, click on either: • Yes, 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 OR • No, 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

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'

Retrieve Prior Revision - SCC Client

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

Context Menu	Right-click on Package > **Package Control** > File History

Example Procedure - retrieve prior revision, TFS-SCC client

Step	Action
1	Display the 'TFS File History' dialog.
2	Click on the **Get button**. The TFS-SCC client displays the 'Resolve Conflicts' dialog. This dialog offers the 'Automerge All XML Package files' option. DO NOT select this option. It is important to prevent any merging of Enterprise Architect's XML Package files.
3	Click on the **Resolve button**. The TFS-SCC client displays the 'Resolve writeable file conflict' dialog.
4	Select the 'Overwrite local file/folder' option. The existing working copy of the Package file is overwritten by the prior revision retrieved from **Version Control**.
5	Click the **OK button**. The TFS-SCC client redisplays the 'Resolve writeable file conflict' dialog; it should now show no conflicts.
6	Click on the **Close button**. The TFS-SCC client redisplays the 'File History' dialog.
7	Click on the **Close button**. Enterprise Architect displays a prompt, asking whether to check-out the prior revision.
8	Click on the: • **Yes button** to check-out the prior revision • **No button** to retrieve a read-only version of the Package, that is NOT checked-out and is NOT editable

Advanced Usage

Once you get familiar with using the fundamental functions of **Version Control** there are a number of more advanced functions that you might want to use in working with your models. The advanced Version Control functions are outlined in this table.

Advanced Functions

Export a Version Controlled Model Branch	Exports **Version Control** information about the root Package of a model Branch, that is used to simplify the process of exporting and importing a hierarchy of Packages from one model to another.
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.
Import a Model Branch From Version Control	Uses Enterprise Architect's Model Branch files, 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.
Add Connectors To Locked Elements	Generally, when working in a diagram containing locked elements, you cannot add a connector to a locked element. There are scenarios in which a connector can be created on a locked element depending on the locking status of the parent Packages.
Validate Package Configurations	You can test the validity of the **Version Control** settings associated with each Version Controlled Package within your current model.
Resynchronize the Status of Version Controlled Packages	Re-synchronizes the **Version Control** status of Packages within your project with the status reported by your Version Control provider.

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

Context Menu	Right-click on Package > **Package Control** > Get Package

Import Packages from Version Control into current model

Step	Action
1	On the 'Get Shared File' dialog, click on the **drop-down arrow** of the '**Version Control** Configuration' 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 **OK 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
- XMI Package files associated with Packages that are already part of your project, are NOT included in the list of files available for import

Export Controlled Model Branch

Applying **Version Control** to a model can result in many XMI files being 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

Context Menu	Right-click on Package > **Package Control** > Export as Model Branch

Create a Model Branch File to represent a Package hierarchy stored in Version Control

Step	Action
1	On the 'Export as Model Branch' dialog, in the 'EAB Filename' field, type a name for your Model Branch File. Alternatively, click on the ... button and browse for the file location. Note that the Package name is supplied as a default.
2	Click on the **OK 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

Import Controlled Model Branch

Applying **Version Control** to a model can result in many XMI files being 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 XMI 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

Context Menu	Right-click on target Package for import > **Package Control** > Import a Model Branch

Import a Model Branch

Step	Action
1	On the 'Import VC Model Branch' dialog, either: • Use the lower portion of the dialog to select a model branch file (this is the simpler option if the associated **Version Control** configuration has already been saved in the current model; continue to step 2) OR • Click on the **Find a Model Branch (.EAB) file button** (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 *Manually Locating Model Branch Files* topic)
2	Click on the **drop-down arrow** in the 'Select a **Version Control** Configuration' field and select a configuration. A list of .eab files controlled by that configuration is displayed in the 'Select a Model Branch (.EAB) file' list.
3	Select the Model Branch File you need, then click on the **OK button**.

	The system imports the root Package specified in the Model Branch File and recursively imports and populates all the sub-Packages contained in the root Package.

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

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

Context Menu	Right-click on target Package for import > **Package Control** > Import a Model Branch : Find a Model Branch (.EAB) file

Locate the Model Branch File

Step	Action
1	Access the 'Import VC Model Branch' dialog, then browse for and select the Model Branch File that represents the model branch to import.
2	Click on the **Open button**. If the **Version Control** configuration referenced by the file is fully defined within the current model, the import commences at this point. Otherwise, Enterprise Architect displays a dialog prompting you to complete the required configuration.
3	Click 'Yes' to proceed with completing the definition of the **Version Control** configuration. The 'Version Control Settings' dialog is displayed.
4	Complete the definition of the configuration. (Typically this involves simply specifying the working copy folder.)
5	Click on the **Save 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

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
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

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

Ribbon	Configure > **Version Control** > Check Configuration

Validate Version Control settings

Step	Action
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.
2	Open the '**Version Control** Settings' dialog to complete the definition of any invalid or missing Version Control configurations.
3	Click on an error message in the **System Output** window to highlight the corresponding Package in the **Browser window**.
4	Right-click a Package node and select '**Package Control** | Configure Package' 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 XMI file.

Resynchronize the Status of Version Controlled Packages

It is possible to update the **Version Control** status of 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

Ribbon	Configure > **Version Control** > Re-Synch Status (applies to all Packages in model)
Context Menu	Right-click on Package > **Package Control** > Re-synch Status With VC Provider (applies to single Package only)

Resynchronize Version Control status

Step	Action
1	The results of the re-synchronization process are sent to the **System Output** window.
2	Double-click on any result message to select, in the **Browser window**, 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

More Information

Edition Information

- **Version Control** facilities are available in the Professional, Corporate, Unified and Ultimate Editions of Enterprise Architect.
- A team needs to provide their own Version Control Server and Client that are compatible versions.

Notes

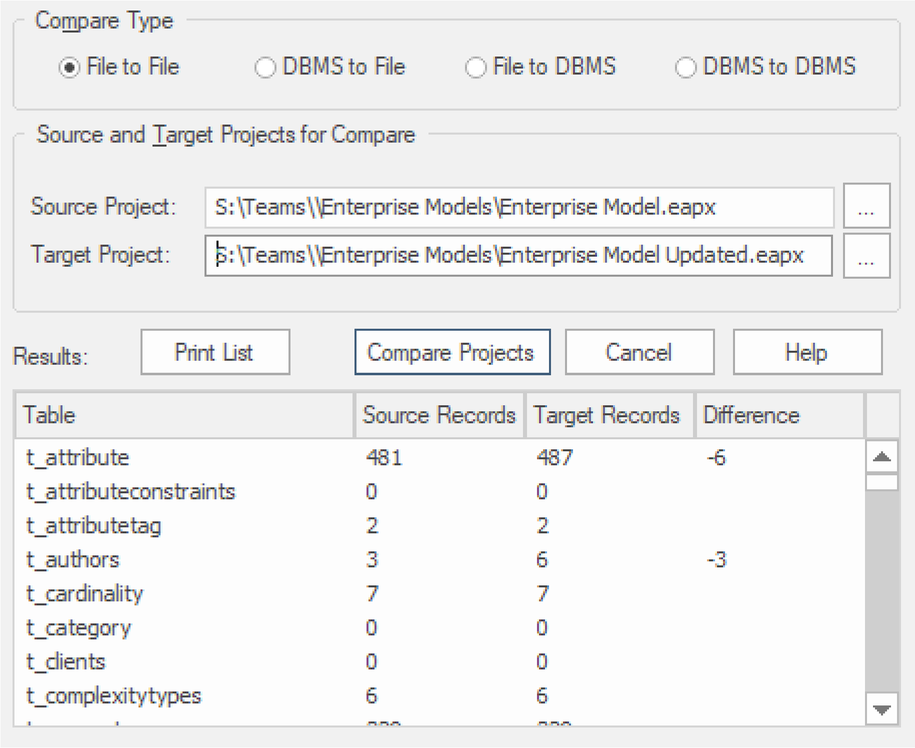
- Sparx Systems strongly urge you not to manipulate Version Controlled Package files outside of Enterprise Architect; it is possible to leave the Package files in a state that Enterprise Architect cannot recognize
- Database replication should not be combined with Version Controlled Packages
- If the 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 'Preferences' 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

Compare Projects

Compares a Source and Target Repository for Differences

The Compare Project facility provides a summary of the changes to a repository by providing a comparison of the number of rows in each system table in the source and the target repositories. The modeling information contained in an Enterprise Architect repository is stored in a set of tables in a relational database. These are system tables that allow you to compare the number of rows in each of the tables, which will provide valuable insights into the differences between two repositories. The tables (with rare exceptions) have intuitive names as indicated by these examples:

- t_attribute - stores element attributes
- t_diagram - stores diagrams
- t_object - stores elements
- t_package - stores Packages (folders)



Compare Type: ● File to File ○ DBMS to File ○ File to DBMS ○ DBMS to DBMS

Source Project: S:\Teams\\Enterprise Models\Enterprise Model.eapx

Target Project: S:\Teams\\Enterprise Models\Enterprise Model Updated.eapx

Table	Source Records	Target Records	Difference
t_attribute	481	487	-6
t_attributeconstraints	0	0	
t_attributetag	2	2	
t_authors	3	6	-3
t_cardinality	7	7	
t_category	0	0	
t_clients	0	0	
t_complexitytypes	6	6	

Apart from the deliberate changes that you and other modelers make when working on your repository, a number of operations can also make changes to your project that you either want to monitor carefully or that you do not want to retain. Such events include:

- Recovering from a database problem
- Restoring a backup
- Performing a Project Data Transfer
- Importing from XMI, 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' that the two repositories are equivalent. The repositories can be on different platforms and the comparison can be made between files and server based repositories as described here:.

- Compare a project file to another project file (e.g. a *eapx file and a *.feap)
- Compare a project file to a DBMS-based repository (e.g. a *eapx file and an Oracle db)

- Compare two DBMS repositories (e.g. a MySQL db and an Oracle db)

The comparison examines the number of rows in each database table, producing a report indicating the total records in each and the difference in record count between the two. If discrepancies are found, you would need to investigate further manually. The comparison does not examine the actual data in the tables but rather provides the summary in the form of table row counts

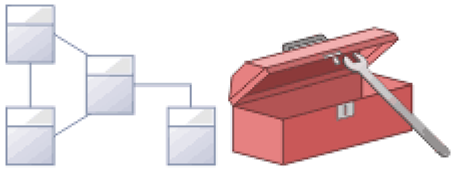
Access

Ribbon	Configure > Model > Integrity > **Project Compare**

Compare two projects

Step	Action
1	On the '**Project Compare**' dialog, select the radio button for the database types of the two projects you want to compare: • File to File • DBMS to File • File to DBMS • DBMS to DBMS
2	In the 'Source Project' and 'Target Project' fields, type the name or connection string for the source and target projects to compare.
3	Click on the **Compare Projects 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 **Print List button**.

Project Maintenance



The repository is an important data store that must be maintained to ensure that it is running efficiently and that the information it contains has integrity. Enterprise Architect has built-in features that allow you to keep the repository in good health. These include tools to maintain the database itself, such as Repair and Compact functions (for file based repositories), and a series of tools to manage the data and models, such as model validation and data integrity checks. Reference data can be imported and exported from the repository, and can be shared between repositories, ensuring consistency. The **Browser window** allows repositories to be refactored easily by dragging and dropping Packages, elements, features and diagrams to new locations. The Discussion and Library features allow the model to be critically analyzed and reviewed.

Facilities

Facility	Detail
Check the integrity of the data in a project	If you have a failed XMI 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.
Reset Auto Increment Columns in Tables	XMI Import/Export can affect the table auto increment column values and push them towards their maximum allowed value; you can re-sequence the columns to avoid this problem.
Upgrade an old project to enable use of new features	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.
Rename a project	If you want to rename an Enterprise Architect file-based project, you can do so through Windows Explorer.
Compact a project	As with many relational database products .EAP/.EAPX files do not physically release allocated space after data is deleted, therefore from time to time it can be beneficial to perform a 'compact', to reduce the physical size of the project file.
Repair a project if it did not close properly	If a .EAP/.EAPX project has not been closed properly, such as during system or network outages, it is possible that the project will reports inconsistencies errors the next time it is opened. To resolve such issues a 'Repair' can be performed to correct any potential problems.

Notes

- You can only rename, compact and repair models created as .EAP/.EAPX files with Enterprise Architect. While most DBMS products provide similar functionality, they typically require an elevated level of permissions and therefore usually performed by your Database Administrator (DBA).

Brief Introduction

An Enterprise Architect repository is continuously evolving as you add new items or change or delete existing items. During this evolution events can occur that have some minor internal impact on the integrity of the records stored in the database. New product enhancements are also made as a result of the company's strong commitment to innovation and from feedback from customers. These enhancements need to be incorporated into existing product installations so that customers continue to derive the benefit and value from the tool. In this topic you will learn about some of the remedial functions that can be used to keep you repository in good health and operating at optimal speed as listed below:

- Check Project Data Integrity - perform a health check
- Reset Auto Increments - reset the auto-increment column on system tables
- Upgrade a Project - augment the files to support new features or enhancements
- Run SQL Patches - run a SQL script to correct or enhance a database
- Rename a Project - change the name of a project file
- Compact a Project - decrease the size of a file based database
- Repair a Project - correct a problem due to file corruption from an unexpected network shutdown

Check Project Data Integrity

Over time as the models in your repository evolve small aberrations can occur in the database typically as a result of badly formed XMI imported from third-party systems or network crashes or other unforeseen events that disrupt the integrity of information in the repository. To ensure your repository is in good-health 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 or other modeling language conformance, only the data relationships and repository structure. So it will check that all attributes have identifiers but it will NOT check that you have used the correct relationship between a Use Case and an Actor.

project-integrity-check-report

You can conveniently run the integrity checker first in Report mode to discover and analyze problems located by the Integrity Checker. Once you have looked through the list of Problems that the check has found and viewed the proposed actions you can and then run it again in 'Recover/Clean' mode.

Other than for quite small repositories It is recommended that you don't run all the available checks at the same time as this can take some time to complete. Either run them individually or in small sets.

When Enterprise Architect 'Cleans' the model, it attempts to recover any lost Packages and Elements, and generates a new Package called _Recovered_. Check through any elements that are found and, if required, drag them back to where they belong in the **Browser window**. If they are not required, you can choose to delete them or export them as an XMI export to serve as a backup.

Access

Ribbon	Configure > Model > Integrity > Project Integrity

Check the data integrity of your project

Step	Action
1	Open the project.
2	Select the 'Project Integrity Check' menu option. The 'Project Integrity Check' dialog displays.
3	Select the checkbox for each check to run: • Package Structure • Object Structure • Object Features • All GUIDs • Cross References

	• Connectors • UML 2.0 Migration
4	Select either: • The 'Report Only' option to just view a report on the state of your model, or • The 'Recover/Clean' option to recover and clean your project; before selecting this option, back up your project file first
5	To write a log of the integrity check, click on the Save Results button and select a log file.
6	Click on the **Go button** to run the check. If you want to display the resulting information in a more readable layout, you can resize the dialog and its columns.

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, Unified and Ultimate Editions of Enterprise Architect, if security is enabled you must have 'Check Data Integrity' permission to perform a data integrity check

Reset Auto Increments

Enterprise Architects models are natively stored in tables in a relational database and the tool uses the auto-increment feature of these database system to when inserting new rows in its system tables. There are a number of operations such as bulk imports and exports from XML files that can cause a table's auto-increment column to approach its largest allowed value. This topic explains how to correct this issues and to re-sequence the values in any affected tables .

Impacts

Impact Due To	Impact
XML Export/Import	XML Export/Import can cause gaps in the numbering sequence of auto increment columns. Each XML Import deletes rows from several tables; the import then adds rows starting from the largest previous value of the auto increment column. 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 int datatype has a maximum value of 2,147,483,647.
Replication	Large auto increment values can also arise where the project originated as an EAP replica or design master. The Jet engine assigns random values for auto increment columns with each XML Import into the project. 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

Ribbon	Configure > Model > Integrity > Reset IDs

Re-sequence auto increment columns

Step	Action
1	Open the project.
2	Select the 'Reset IDs' menu option. A dialog displays listing all non-empty tables that contain an auto increment or identity column. • The 'Rows' column shows the number of rows in the table • The 'Maximum ID' column shows the current maximum value of the auto increment column • The 'Action' column shows either 'No Action' or 'Reset', depending on how close the column value is to the datatype's maximum

	Tables requiring a reset are automatically selected in the list.
3	Click on the **Go button** to reset the auto increment column values.

Notes

- Connection to the project must be via a **direct** connection; IDs cannot be reset while the connection to the repository is via Pro Cloud Server
- Project **Auditing** must be disabled before IDs can be reset
- MySQL repositories will require re-starting of the MySQL server so that the reset auto increment columns start from the reset value
- While strict not necessary, Sparx System would recommend that this task is performed when all users are not actively using the Project
- Sparx System recommend that a backup of the Project should be done both before and after performing a Reset of Table Auto Increment columns
- In the Corporate, Unified and Ultimate Editions of Enterprise Architect, if security is enabled you must have 'Check Data Integrity' permission to reset IDs

Upgrade a Project

The structure of Enterprise Architect project files is occasionally changed to support new features. In such cases, the project might have to be upgraded. Upgrading to the new file structure is a simple and quick process that brings your project to the current level to:

- Ensure correct operation and
- Support all the latest Enterprise Architect features

Initial Check

When you load a project that was created in an earlier 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.

The Upgrade Project Wizard

The Upgrade Project Wizard takes you through the upgrade process and:

- Advises you of the necessity to upgrade
- Advises you to back up the current project; it is essential to back up before any changes are made
- Checks which upgrade path is required
- Guides you through the steps to perform the upgrade
- Opens the newly converted project

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

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

Run SQL Patches

Enterprise Architect is a robust platform and there are many internal checks that are applied to ensure that the integrity of the information in the repository is maintained. Occasionally an issue might arise and Sparx Systems may choose to release a patch to that to rectify the problem.

The patch generally checks how many records are to be updated, and reports on what is to be done.

Access

Ribbon	Configure > Model > Integrity > Run Patch

Run SQL Patch

1	Open Enterprise Architect and load the Project that requires the patch applied.
2	Select the 'Run Patch' option. The 'Run Patch' dialog displays.
3	Click the **Browse button** and select the supplied patch file (*.XML) in the file browser. Upon loading the select file, the Description will then be loaded with the explanation of what action the patch will perform.
4	Click the **Run button** to apply the patch to the current Project.

Rename a Project

Occasionally the purpose of your repository may change, or you may for any number of reason want to change its name. While you can change the name of the root package in the repository to change the name of the repository itself you will need to rename the file 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.

Compact a Project

After a period of time, through general use, 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 the unused space.

Access

Ribbon	Configure > Model > Integrity > Manage .EAPX/.EAP File > Compact .EAPX/.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.
3	Select the 'Compact .EAPX/.EAP File...' 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, Unified and Ultimate Editions of Enterprise Architect, if security is enabled you must have 'Administer Database' permission to compact a project

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 unrecognized 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

Ribbon	Configure > Model > Integrity > Manage .EAPX/.EAP File > Repair .EAPX/.EAP File

Repair a project that was not closed properly

Step	Action
1	Ensure that all users are logged off the project you are attempting to repair.
2	Copy the project file to a local drive on your workstation.
3	In Enterprise Architect, select 'Start > Desktop > Preferences > Preferences' to display the 'Preferences' dialog. On the 'General' page, deselect the 'Use Jet 4.0 - requires restart' checkbox.
4	Close and restart Enterprise Architect - you do not need to open any model, including the one you are repairing.
5	Select the 'Repair .EAPX/.EAP File...' 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.

Notes

- Always repair projects on a local drive, never on a network drive
- The 'Compact a Project' option can also repair corrupt .eap files
- In the Corporate, Unified and Ultimate Editions of Enterprise Architect, if security is enabled you must have 'Administer Database' permission to repair a project

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.

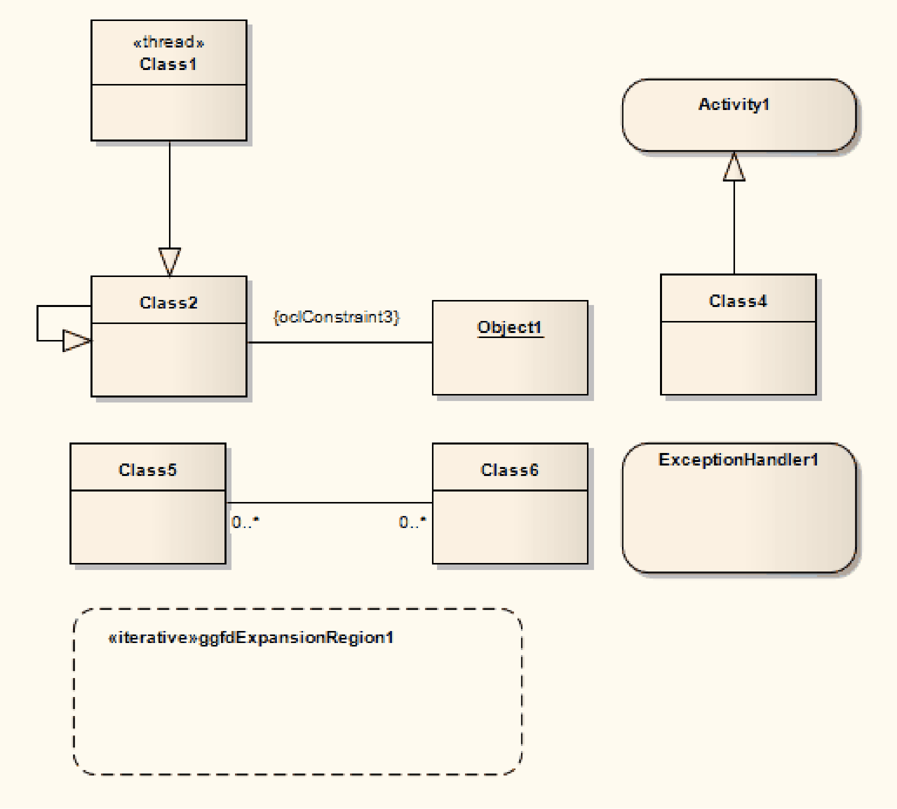
You can add to the system-provided validation rules by using an **Add-In** or an MDG Technology (such as BPMN) that contains its own rules.

Access

Ribbon	Design > Model > Manage > **Validate** > Validate Current Package

Validation Concepts

Concept	Detail
Validating Objects	Validating a UML: - Element validates the element and its children, its features (attributes and operations) and its relationships (connectors) - Diagram validates the diagram itself (for correctness) as well as any elements and connectors within the diagram - Package validates the Package and all sub-Packages, elements, connectors and diagrams within it
Example - Model Violation	This UML diagram contains several basic violations of the UML rules:

	 «thread» Class1 Activity1 Class2 {oclConstraint3} Object1 Class4 Class5 0..* 0..* Class6 ExceptionHandler1 «iterative»ggfdExpansionRegion1 If you run **Model Validation** on this diagram, these violations are identified in the **System Output** window: • A UML ExpansionRegion (ExpansionRegion1) is missing its child input ExpansionNode • An invalid self-generalization exists on Class2 (UML elements cannot be self-generalized) • An OCL violation exists for the anonymous Association (between Class2 and Object1) • A UML ExceptionHandler (ExceptionHandler1) is missing its child input ObjectNode

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

Ribbon	Design > Model > Manage > **Validate** > Configure Validation Rules

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.

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 **Browser window** or a diagram. If you do not want to continue a validation you have started, you can cancel it.

Access

Ribbon	Design > Model > Manage > **Validate** > Validate Current Package
Keyboard Shortcuts	**Ctrl+Alt+V**

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 'Design > Model > Manage > **Validate** > Cancel Current Validation'

Notes

- If the **System Output** window does not automatically display, select 'Start > Desktop > Design > System Output'.

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
Well-Formedness	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. As explained in *Error Message Format*, these rules generate messages with the codes: • MVR01... (Element: Well Formedness checkbox selected) • MVR05... (Relationship: Well Formedness checkbox selected) • MVR08... (Feature: Well Formedness checkbox selected) • MVR0B... (Diagram: Well Formedness checkbox selected)
Element Composition	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. These rules generate messages with the code: • MVR02... (Element: Composition checkbox selected)
Property Validity	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. These rules generate messages with the codes: • MVR03... (Element: Property Validity checkbox selected) • MVR06... (Relationship: Property Validity checkbox selected) • MVR09... (Feature: Property Validity checkbox selected)
Custom Properties	The validator applies these rules to check an element, relationship or feature against any defined constraints in OCL. These rules generate messages with the codes: • MVR04... (Element: OCL Conformance checkbox selected) • MVR07... (Relationship: OCL Conformance checkbox selected) • MVR0A... (Feature: OCL Conformance checkbox selected)

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

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
MVR010001	«Element» is not a valid UML Element Error: The element is not a recognized UML 2.5 element.
MVR050001	«Relationship» is not a valid UML Relationship Error: The relationship is not a recognized UML 2.5 relationship.
MVR050002	«Relationship» is not legal for «Start Element» --> «End Element» Error: The relationship between the given start and end elements is not valid for those elements.
MVR050003	«Parent Element»:isLeaf=true and cannot be generalized by «Child Element» Error: The Generalization relationship cannot exist between parent and child elements because the parent element is defined as a leaf element.
MVR050004	«Child Element»:isRoot=true and cannot generalize «Parent Element» Error: The Generalization relationship cannot exist between parent and child elements because the child element is defined as a root element.
MVR050005	«Element» cannot generalize self Error: The element cannot be self-generalized.
MVR0B0001	Statechart violation: «extended information» Error: The State diagram contains a UML violation; «extended information» provides more details on the specific violation.
MVR0C0001	Sequence Violation: «extended information» Error: 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.)
MVR7F0001	«Requirement» Not realized Error: A Requirement element must be at the target end of a Realization, Association (including Aggregation and Composition) or Generalization connector.

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
MVR020001	«Element» is missing required child element «Child Element» Information: The element is missing a child element of type Child Element.
MVR020002	Invalid **UML Package** child Information: 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).
MVR020003	Invalid child «Child Element name» («Child Element Type») Information: The child element is invalid on the tested parent element.

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
MVR030001	«Element»:«Property» property is undefined Information: The element property contains no value.
MVR030002	«Element»:«Property» property has invalid value: "«Value»" Information: The element property contains an invalid value.
MVR030003	«Element»:isLeaf=true and cannot be abstract Information: The element's isLeaf and isAbstract properties are both set to **True**, which is invalid.
MVR060001	«Relationship»:«Property» property is undefined Information: The relationship property contains no value.
MVR060002	«Relationship»:«Property» property has invalid value: "«Value»" Information: The relationship property contains an invalid value.
MVR090001	Attribute/AssociationEnd mismatch, «Attribute»: «Mismatch description», ... Information: The given attribute has an associationEnd of the same name but they differ in the listed details.

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.

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
An Element	Right-click on the element and select 'Properties | Properties Dialog'. Click on the 'Constraints' page; in the 'Type' field click on the **drop-down arrow** and select OCL. To perform an OCL Validation: 1. Select 'Design > Model > Manage > **Validate** > Configure Validation Rules' and select the 'Element: (OCL) Conformance' checkbox. 2. Click on the element and select 'Design > Model > Manage > Validate > Validate Current Package'. Any OCL syntax violations are recorded in the '**Model Validation**' tab of the **System Output** window.
A Relationship	On a diagram, right-click on the connector and select the 'Properties' option; select the 'Constraints' page, and in the 'Type' field click on the **drop-down arrow** and select 'OCL'. To perform an OCL Validation: 1. Select 'Design > Model > Manage > **Validate** > Configure Validation Rules' and select the 'Relationship: (OCL) Conformance' checkbox 2. Click on the connector and select 'Design > Model > Manage > Validate > Validate Current Package' Any OCL syntax violations are recorded in the '**Model Validation**' tab of the **System Output** window.
An Attribute	Right-click on the element and select 'Attributes | Constraints'; in the 'Type' field click on the **drop-down arrow** and select 'OCL'. To perform an OCL Validation: 1. Select 'Design > Model > Manage > **Validate** > Configure Validation Rules' and select the 'Feature: (OCL) Conformance' checkbox. 2. Click on the element and select 'Design > Model > Manage > Validate > Validate Current Package'. Any OCL syntax violations are recorded in the '**Model Validation**' tab of the **System Output** window.

Reported violations

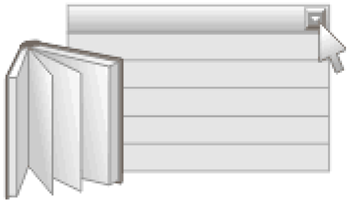
Error ID	Description

MVR040001	OCL violation: «violated OCL» Information: The element's OCL constraint uses incorrect syntax.
MVR070001	OCL violation: «violated OCL» Information: The relationship's OCL constraint uses incorrect syntax.
MVR0A0001	OCL violation: «violated OCL» Information: The attribute's OCL constraint uses incorrect syntax.

Notes

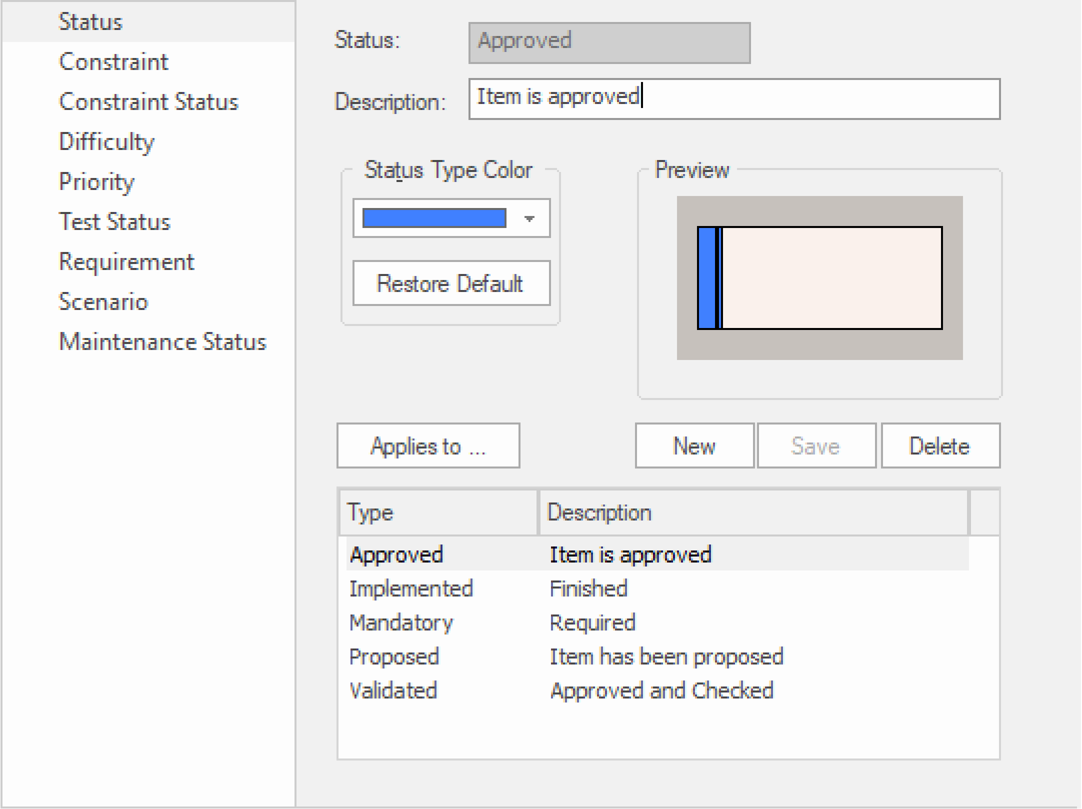
- To have a valid OCL constraint, the syntax must be correctly formed; if the expression is not correct, a message displays stating that the OCL constraint is not valid

Reference Data



Reference Data Defines Meta Data for items in the Repository

Reference data is used universally in Enterprise Architect as meta-information for properties and base values for repository items including elements, features and connectors and tool features. It is often seen in drop-down lists and selection boxes, and includes such items as Status Codes, Requirement Types, Model Authors and more.



- Status
- Constraint
- Constraint Status
- Difficulty
- Priority
- Test Status
- Requirement
- Scenario
- Maintenance Status

Status: Approved

Description: Item is approved

Type	Description
Approved	Item is approved
Implemented	Finished
Mandatory	Required
Proposed	Item has been proposed
Validated	Approved and Checked

General Types window showing Status Types including color settings for each type.

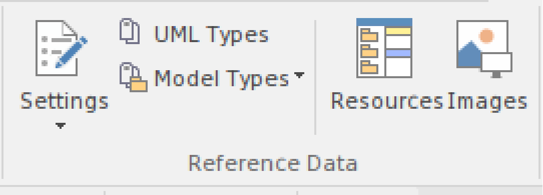
For convenience, Enterprise Architect comes pre-configured with useful reference data, but it is best practice to review this existing data when setting up a project and either replace or edit it to ensure it is appropriate for your organization's projects.

Brief Introduction

Reference data is an important part of a repository and, apart from ensuring integrity and consistency, can drive the visualizations of reports and dashboard widgets. You can, for example, visualize the Requirements grouped by Status and Difficulty using Charts and graphs. You can define reference data by using the dialogs for each reference data type or by importing it from another repository. It is best practice to set up the metadata before populating the repository with your organization's modeling information. This ensures that when you add elements you will have the right information available, and removes the need to change the elements again.

There are different types of reference data, some of which apply at a repository level - such as Status Codes and Authors - and some that are more technical in nature - such as Code Engineering and Database Datatypes - that apply to particular element or tool types.

You can transfer reference data between repositories by exporting it from one repository and importing it into another; this mechanism can save time and ensure consistency between repositories.



Configure ribbon Reference Data panel.

- UML Types - Stereotypes, **Tagged Value Types** and Cardinality Values
- People - Authors
- General Types - Status Types, Constraints, Requirement Types
- Project Indicators - Risk, Metric, Effort
- Estimation Factors - Technical and Environmental
- Datatypes and Namespaces
- People
- General Types

UML Types

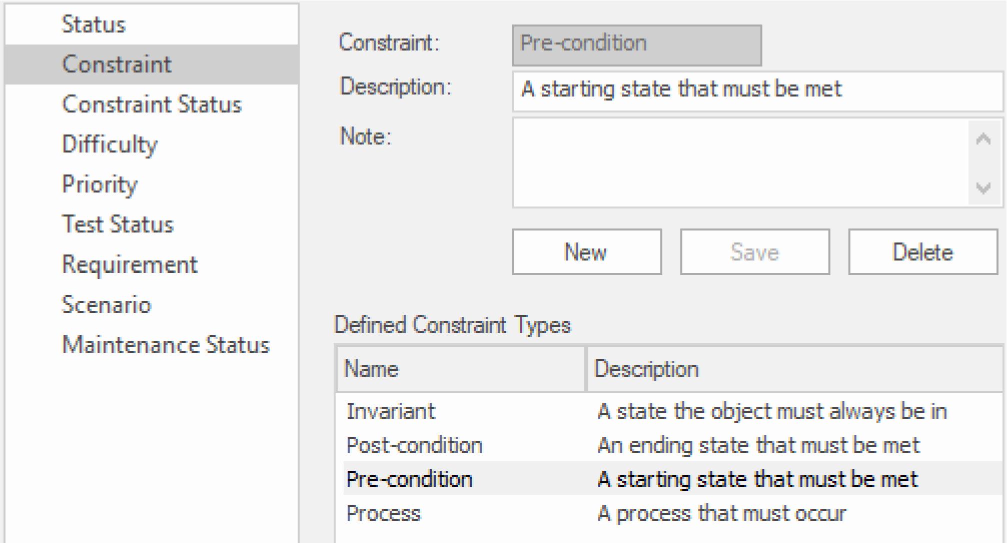
These are part of the UML Grammar and allow you to extend the core language to make it more suitable for your organization's projects. You use them to create new types (Stereotypes) that are relevant to the industry you work in or to add specific properties (Tagged Values) that help to specify the important aspects of elements of our domain. You can also specify a set of number pairs that specify how many (Cardinalities) of one item relates to another item.

People

People are the stakeholders who participate in some way in the modeling of the enterprise and its projects. They include the people who create and maintain the models (Authors), the parts they play in the modeling effort (Roles), the people who carry out project related tasks (Resources) and the people who ultimately requisitioned the models and system (Clients).

General Types

These are collections of meta data values that you use in drop-down lists when defining element properties. They add rigor to your models and are also used extensively in creating visualizations.



Constraint: Pre-condition

Description: A starting state that must be met

Note:

Defined Constraint Types

Name	Description
Invariant	A state the object must always be in
Post-condition	An ending state that must be met
Pre-condition	A starting state that must be met
Process	A process that must occur

Constraints Type node of the **General Types window** showing the Pre Condition type.

Project Indicators

You can add a new type to the global set of risk types.

Maintenance

You can define Risk type and Test types which can be used when creating project based Risks and Test Respectively. There are a set of predefined types but you can add new types or modify or delete existing ones.

Metrics and Estimation

You can estimate the size of a software centric system by using Use Case points which utilize Environment and Technical Complexity factors and an hourly rate to calculate the effort required.

Auto Names and Counters

You can set a predefined name prefix including a counter for all element types which is used whenever a new element is created incrementing the counter each time.

Data Types and Namespaces

These define the set of data types used by both programming languages and database systems. These are preset for each of the languages and systems but you can define new types as required. You can also add completely new products that are not part of the predefined set and add types to these. Namespaces can also be defined.

Sharing Reference Data

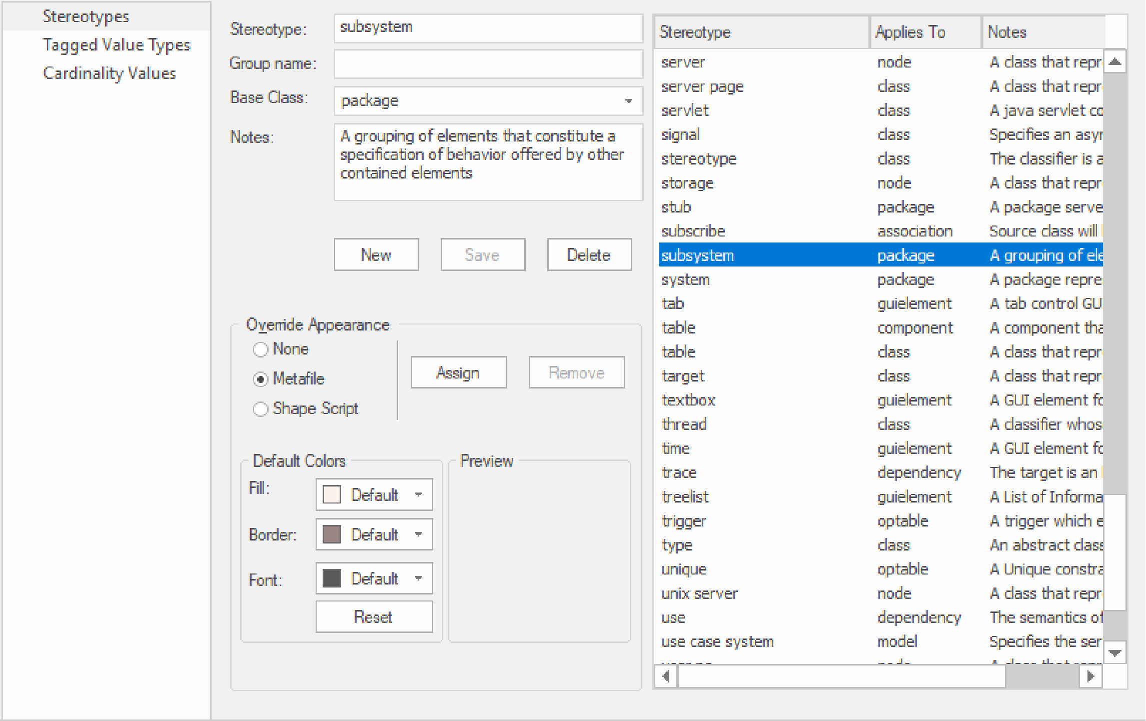
Enterprise Architect pre-populates reference data as default data, which you can then edit as required. Another productive way of getting data into a repository is to import it from an existing Enterprise Architect repository. This is a simple process of exporting the data from one repository and importing it into another.

UML Types

When the UML was devised its 'grammar' was designed to allow for extensions much in the same way that natural language grammars such as English and French are extended over time. Two of the most important of these extension mechanisms are Stereotypes and Tagged Values.

Using the 'UML Types' dialog you can configure stereotypes, **Tagged Value Types** and Cardinality (Multiplicity) Values for your project. In more detail, you can:

- Add, modify and delete stereotypes
- Customize stereotypes' appearance
- Update Shape Scripts and metafiles associated with stereotypes
- Create and modify Tagged Values
- Define cardinality rules



Stereotype: subsystem

Group name:

Base Class: package

Notes: A grouping of elements that constitute a specification of behavior offered by other contained elements

Override Appearance

○ None

● Metafile

○ Shape Script

Default Colors

Fill: Default

Border: Default

Font: Default

Stereotype	Applies To	Notes
server	node	A class that repr
server page	class	A class that repr
servlet	class	A java servlet co
signal	class	Specifies an asyr
stereotype	class	The classifier is a
storage	node	A class that repr
stub	package	A package serve
subscribe	association	Source class will
subsystem	package	A grouping of ele
system	package	A package repre
tab	guielement	A tab control GU
table	component	A component tha
table	class	A class that repr
target	class	A class that repr
textbox	guielement	A GUI element fo
thread	class	A classifier whos
time	guielement	A GUI element fo
trace	dependency	The target is an
treelist	guielement	A List of Informa
trigger	optable	A trigger which e
type	class	An abstract class
unique	optable	A Unique constra
unix server	node	A class that repr
use	dependency	The semantics of
use case system	model	Specifies the ser

Stereotypes window showing the details for the Subsystem stereotype.

Access

Ribbon	Configure > Reference Data > UML Types

Stereotype Settings

Enterprise Architect has an extensive set of Standard 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 the *Custom Stereotypes* Help Topic.

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 object shapes to determine the shape and dimensions of the object.

Access

Ribbon	Configure > Reference Data > UML Types

Manage Stereotypes

Option	Action
Stereotype	Type in the name of the new stereotype, or click on an existing stereotype in the list in the right-hand panel.
Group name	For a stereotype that applies to an element feature (an attribute or operation) type in a name under which to group the features that have the stereotype. This is shown on diagrams in the attributes or operations compartment of the element. If you do not provide a group name, the label in the feature compartment defaults to the stereotype name.
Base Class	Click on the **drop-down arrow** and select an object type, so that the stereotyped object will inherit the base characteristics from that existing object type.
Notes	Type in any notes concerning the stereotype (and not the objects to which the stereotype is to be applied).
None	Select this radio button to use the default object appearance.
Metafile	Select this radio button to use an image file for the appearance of the stereotyped object.
Shape Script	Select this radio button to apply a custom shape to the stereotyped object, which you define using the Enterprise Architect Shape Scripting language.
Assign	Click on this button to select or define the associated metafile or Shape Script to be applied to the stereotyped object.
Remove	Click on this button to remove the associated metafile or Shape Script from the stereotype.
Fill	Click on the **drop-down arrow** and select a default color for the background of the

	object. This color will be applied to all occurrences of any object to which the stereotype has been applied; if the color is subsequently changed, the change is immediately applied to all occurrences of any object 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, such as the 'Alternative Appearance' dialog (**F4**) definition.
Border	Click on the **drop-down arrow** and select a default color for the object border.
Font	Click on the **drop-down arrow** and select a default color for the stereotype font.
Reset	Click on this button to reset the stereotyped object appearance definition to the default object appearance.
Preview	Provides a visual representation of the stereotyped object definition.
New	Click on this button to clear the dialog fields so that you can define another stereotype.
Save	Click on this button to save the stereotype definition and add it to the list of current stereotypes.
Delete	Click on this button to immediately remove the selected stereotype from the list of defined stereotypes.

Shape Editor

A Technology Developer can use the **Shape Editor** 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

Ribbon	Configure > Reference Data > UML Types > Stereotypes > [select or specify a stereotype] : Shape Script : Assign

Reference

Option	Action
Format	The format in which the script is written.
Import	Opens a Windows browser dialog, allowing for a script file to be loaded.
Export	Opens a Windows browser dialog, allowing for a script file to be saved.
Edit window	The area in which a script can be written.
Preview of main	A visual representation of the compiled script. Nothing displays until the **Refresh button** is clicked.
Next Shape	If a composite shape is defined within the Edit Window, clicking on the **Next Shape button** pages through the components of the shape.
Refresh	Parses your script and produces a visual example of the shape in the Preview of main window.
OK	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 'Preferences' 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

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 'Tags' tab of the **Properties 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

Ribbon	Configure > Reference Data > UML Types > **Tagged Value Types**

Create a Tagged Value Type

Field	Action
Tag Name	Type the new name of the Tagged Value.
Description	Type a description of the Tagged Value.
Detail	Type any additional information necessary.
New	Click to begin creating a new Tagged Value.
Save	Click to save the details of a Tagged Value.
Delete	Click to delete a Tagged Value from the Defined Tag Types list.
Defined Tag Types	A list of previously defined Tagged Values.

Notes

- You can transport these **Tagged Value Type** definitions between models, using the 'Configure > Model > Transfer > Export Reference Data' and 'Import Reference Data' ribbon options; **Tagged Value Types** are exported as Property Types

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 these 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

Ribbon	Configure > Reference Data > UML Types > Cardinality Values

Reference

Field	Action
Cardinality	Type the new name of the Cardinality Value.
New	Type a description of the Cardinality Value.
Save	Saves the Cardinality value and adds it to the Cardinality List.
Delete	Deletes a Cardinality value from the Cardinality List.
Cardinality List	A list of already defined Cardinality values.

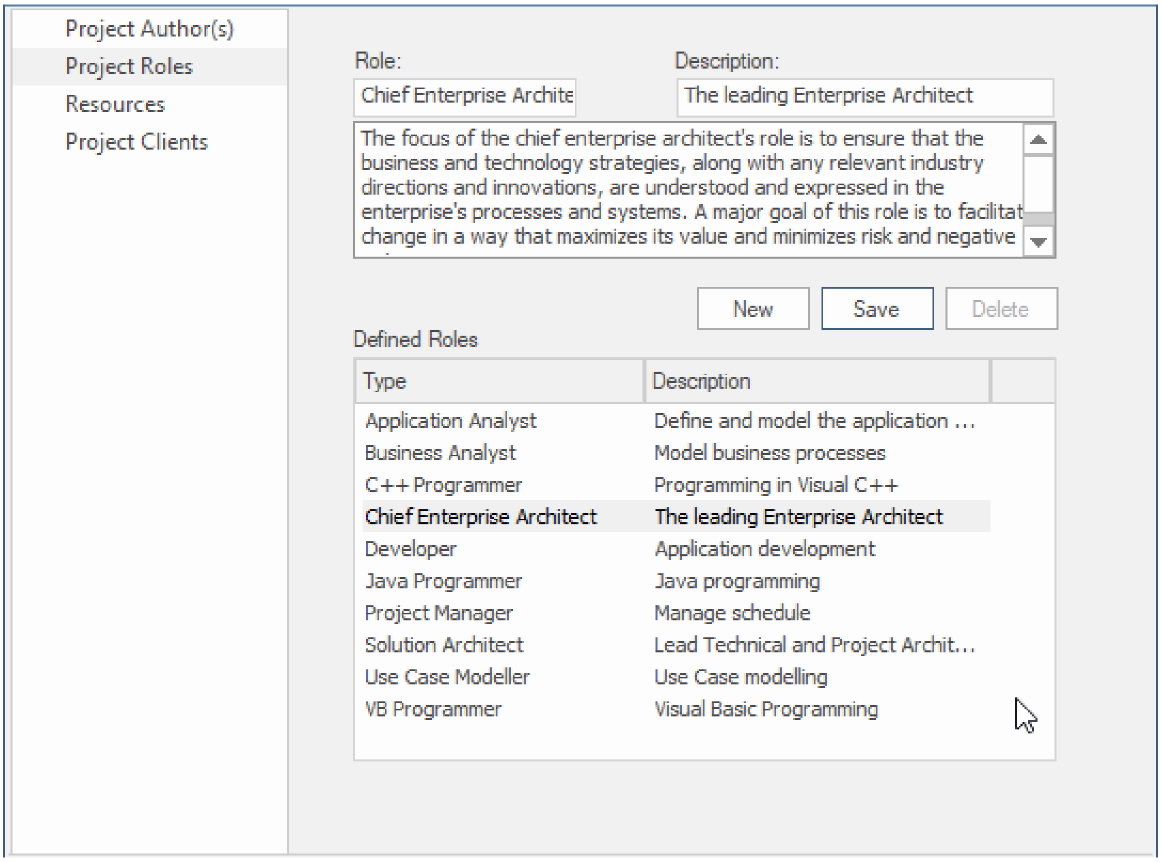
Notes

- You can transport these cardinality values between models, using the 'Configure > Model > Transfer > Export Reference Data' and 'Import Reference Data' ribbon options

People

In your project, you can set up and configure the people and stakeholders involved in and relevant to this model; that is:

- 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



Role: Chief Enterprise Archite

Description: The leading Enterprise Architect

The focus of the chief enterprise architect's role is to ensure that the business and technology strategies, along with any relevant industry directions and innovations, are understood and expressed in the enterprise's processes and systems. A major goal of this role is to facilitat change in a way that maximizes its value and minimizes risk and negative

Defined Roles

Type	Description
Application Analyst	Define and model the application ...
Business Analyst	Model business processes
C++ Programmer	Programming in Visual C++
Chief Enterprise Architect	The leading Enterprise Architect
Developer	Application development
Java Programmer	Java programming
Project Manager	Manage schedule
Solution Architect	Lead Technical and Project Archit...
Use Case Modeller	Use Case modelling
VB Programmer	Visual Basic Programming

Project Roles page of the People window showing the details of the Chief Enterprise Architect role.

Access

Ribbon	Configure > Reference Data > Model Types > People

Project Authors

Project Authors are the individual people (e.g. John Brown) who work to create and maintain the models contained in the repository. When you create a model element your name will be assigned to the element and this name will be usedThey can be assigned as resources that work on model elements and

Access

Ribbon	Configure > Reference Data > Model Types > People > Project Author(s)

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 'Configure > Model > Transfer > Export Reference Data' and 'Import Reference Data' ribbon options

Reference

Field	Action
Name(s)	Type the name of the person registered as a Project Author. If you are using a Windows Active Directory, you can select names from the directory; click on the ... button to display the 'Select Users' dialog. 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. Do not leave any spaces between the names and the semicolons.
Role	(Mandatory) Enter the role the Project Author plays in the project (such as Designer, Analyst, or Architect). You can type a role name or click on the **drop-down arrow** and select a role defined through the 'Project Roles' tab.
Notes	Type any additional notes concerning the Project Author.
New	Add further Authors.
Save	Add the new Author to the Defined Authors list.
Delete	Removes an Author from the Defined Authors list.

Defined Authors	Review the Project Authors already defined.

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

Ribbon	Configure > Reference Data > Model Types > People > Project Author(s) : ...

Reference

Field	Action
Object Types...	Opens the 'Object Types' dialog, which provides a choice of object types that can be used.
Locations...	Defines the root location from which to begin a search.
Check Names	Matches object names listed in the Enter the object names to select section.
Enter the object names to select	Specify object names to search for.
Advanced...	Opens the 'Advanced' dialog to provide further search options. For more information on this dialog, please refer to Windows Help and Support.

Use to

- Select Project Authors from Windows Active Directory

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)

Project Roles

People associated with a project play one or more roles with respect to a model. The roles represent the disciplines associated with the model and system. They are typically defined as part of a system acquisition or development process. The roles could be names like: Chief Enterprise Architect, Business Analyst, Ux Designer, Product Owner. People can be assigned to a role.

Access

Ribbon	Configure > Reference Data > Model Types > People > Project Roles

Reference

Field	Action
Role	Type or select the name of the role.
Description	Type a description of the role.
Notes	Type any additional information related to the role.
New	Add further Roles.
Save	Add the new role to the Defined Roles list.
Delete	Removes a role from the Defined Role list.
Defined Roles	Review the Project Roles already defined. The 'Defined Roles' list is available for selection for any element in the model; for example, you can select roles on the 'Project Authors' page of the 'People' dialog, and the **Resource Allocation window**. You can also specify other roles on these dialogs, but such roles are not added to the 'Defined Roles' list.

Use to

- Define role types that are captured within Enterprise Architect

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 'Configure > Model > Transfer > Export Reference Data' and 'Import Reference Data' ribbon options

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

Ribbon	Configure > Reference Data > Model Types > People > Project Resources

Reference

Field	Action
Name	Type or select the name of the person listed as a resource. The resource name is available for use in Resource Management.
Organization	Type the name of the organization employing the resource.
Role(s)	Type or select the role the resource plays in the project (for example, Designer, Analyst, Architect).
Phone 1, Phone 2, Mobile, Fax	Type the contact telephone numbers for the resource.
Email	Type the email address for the resource.
Notes	Type any additional notes on the resource.
Available Resources	Review resources that have already been defined.
New	Add further resources.
Save	Add a new resource to the Available Resource list.
Delete	Delete a resource from the Available Resources list.

Use to

- Record information on project resources

Notes

- You can transport these resource definitions between models, using the 'Configure > Model > Transfer > Export Reference Data' and 'Import Reference Data' ribbon options

Project Clients

Project clients are the parties (individuals or organizations) that either requisitioned the system or part of the system being modeled or will be the users of the system. Defining these parties helps to keep the focus on the important reasons why the system is built modeled.

Access

Ribbon	Configure > Reference Data > Model Types > People > Project Clients

Reference

Field	Action
Name	Type or select the name of the client.
Organization	Type the name of the organization that employs the client.
Role(s)	Type the role the client plays in the project (for example, Manager, Sponsor).
Phone 1, Phone 2, Mobile, Fax	Type the contact telephone numbers for the client.
Email	Type the email address of the client.
Notes	Type any additional notes on the client.
Defined Clients	Review clients that have already been defined.
New	Add details of further clients.
Save	Add a new client to the Defined Client list.
Delete	Delete a client record from the Defined Client list.

Use to

- Capture client details associated with the current model

Notes

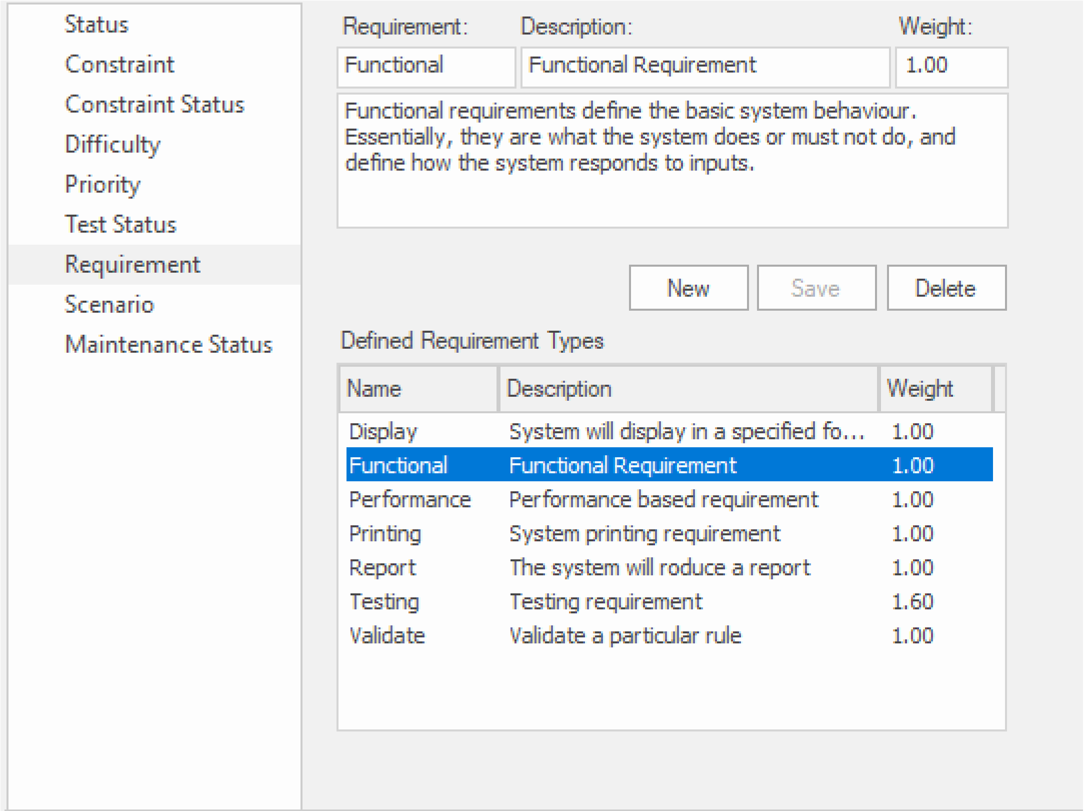
- You can transport these client definitions between models, using the 'Configure > Model > Transfer > Export

Reference Data' and 'Import Reference Data' ribbon options

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
- Maintenance Status



Requirement: Functional

Description: Functional Requirement

Weight: 1.00

Functional requirements define the basic system behaviour. Essentially, they are what the system does or must not do, and define how the system responds to inputs.

Defined Requirement Types

Name	Description	Weight
Display	System will display in a specified fo...	1.00
Functional	Functional Requirement	1.00
Performance	Performance based requirement	1.00
Printing	System printing requirement	1.00
Report	The system will roduce a report	1.00
Testing	Testing requirement	1.60
Validate	Validate a particular rule	1.00

Requirement page of the **General Types window** showing the details for the Functional Requirement Type.

Each of these can be separately managed through the 'General Types' dialog.

Access

Ribbon	Configure > Reference Data > Model Types > General Types

Status Types

The status of an element can define a point in a workflow or a milestone point in project management processes, such as business analysis or product development. 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

Ribbon	Layout > Diagram > Appearance > Configure Status Colors, or Configure > Reference Data > Model Types > General Types > Status

Customize Element Status Types

Option	Description
Status	Type the name of the status.
Description	Type a short description of the status.
Status Type Color	Using the **drop-down arrow**, select a color to be applied to the current status type.
Preview	Displays a visual representation of an element with the selected status type color.
Restore Default	Restore the status type color to its default setting.
Applies to...	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 **Applies to ... button** and, in the 'Applied Status Colors' list, select the checkbox against each element type to which to apply the status colors.
New	Clear the dialog fields to create a new status.
Save	Save the status, and add it to the Type list.
Delete	Remove the currently selected status from the Type list.

Type	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 'Preferences' dialog
- You can transport the status types (and the colors assigned to status types) between models, using the 'Configure > Model > Transfer > Export Reference Data' and 'Import Reference Data' ribbon 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 'Preferences' dialog

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

Ribbon	Configure > Reference Data > Model Types > General Types > Constraint

Customize Constraint Types

Option	Description
Constraint	Type the name of the constraint to define.
Description	Type a brief description of the constraint.
Note	Type any further details concerning the constraint.
New	Clear the fields to define a new constraint.
Save	Save the constraint definition and add it to the Defined Constraint Types list.
Delete	Remove the currently selected constraint from the Defined Constraint Types list.
Defined Constraint Types	Lists the currently-defined constraint types and their descriptions.

Notes

- You can transport customized constraint types between models, using the 'Configure > Model > Transfer > Export Reference Data' and 'Import Reference Data' ribbon options

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

Ribbon	• Configure > Reference Data > Model Types > General Types > Constraint Status Types

Customize Constraint Status Types

Option	Description
Status	Type the name of the constraint status to define.
Make Default	To make the selected constraint status the default for all constraint 'Status' fields, select the checkbox.
New	Clear the fields to define a new constraint status.
Save	Save the constraint status definition and add it to the Type list.
Delete	Remove the currently selected constraint status from the Type list.
Type	Lists the currently available constraint status types.
↑ ↓	Move the selected constraint type higher or lower in the Type list. This defines the ordering used in the Status drop down combo boxes, and in sorting any report or list when the Constraint Status Type is used as the sort-by property.
Sort Alphabetically	Reorganize the entries in the Type list into alphabetical order.
Restore Defaults	Remove all customized constraint status values and restore the system defaults.

Notes

- You can transport customized constraint status types between models, using the 'Configure > Model > Transfer > Export Reference Data' and 'Import Reference Data' ribbon options

Difficulty Types

The elements that you add to a repository can be simple or complex, and as you analyze elements to understand the amount of work required to review, develop or implement them it is expedient to set the Difficulty property that allows you to record this dimension.

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

Ribbon	Configure > Reference Data > Model Types > General Types > Difficulty

Customize Difficulty Types

Option	Description
Difficulty	Type the name of the Difficulty to define.
Make Default	To make the selected Difficulty the default for all 'Difficulty' fields, select the checkbox.
New	Clear the fields to define a new Difficulty type.
Save	Save the Difficulty definition and add it to the Type list.
Delete	Remove the currently selected Difficulty from the Type list.
Type	Lists the currently available Difficulty types.
↑ ↓	Move the selected type higher or lower in the 'Type' list. This defines the ordering used in the 'Difficulty' drop down combo boxes, and in sorting any report or list when the Difficulty type is used as the sort-by property. 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 'Type' field.
Sort Alphabetically	Reorganize the entries in the Type list into alphabetical order.
Restore Defaults	

	Remove all customized Difficulty values and restore High, Medium and Low.

Notes

- You can transport customized Difficulty types between models, using the 'Configure > Model > Transfer > Export Reference Data' and 'Import Reference Data' ribbon options

Priority Types

No all elements are equal, and as you define the elements in a repository you can assign a Priority to indicate the relative importance of the element. You define the element's priority using the 'Priority' field of the elements's 'Properties' dialog. 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

Ribbon	Configure > Reference Data > Model Types > General Types > Priority

Customize Priority Types

Option	Description
Priority	Type the name of the Priority to define.
Make Default	To make the selected Priority the default for all 'Priority' fields, select the checkbox.
New	Clear the fields to define a new Priority.
Save	Save the Priority definition and add it to the Type list.
Delete	Remove the currently selected Priority from the Type list.
Type	Lists the currently available Priority types.
↑ ↓	Move the selected type higher or lower in the 'Type' list. This defines the ordering used in the 'Priority' drop down combo boxes, and in sorting any report or list when the Priority type is used as the sort-by property. 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 'Type' field.
Sort Alphabetically	Reorganize the entries in the Type list into alphabetical order.
Restore Defaults	Remove all customized Priority values and restore High, Medium and Low.

Notes

- You can transport customized Priority types between models, using the 'Configure > Model > Transfer > Export Reference Data' and 'Import Reference Data' ribbon options

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

Ribbon	Configure > Reference Data > Model Types > General Types > Test Status

Customize Test Status Types

Option	Description
Status	Type the name of the Test status to define.
Make Default	To make the selected test status the default for all test 'Status' fields, select the checkbox.
New	Clear the fields to define a new Test status.
Save	Save the Test status definition and add it to the Type list.
Delete	Remove the currently selected Test status from the Type list.
Type	Lists the currently available Test status types.
↑ ↓	Move the selected type higher or lower in the 'Type' list. This defines the ordering used in the test 'Status' drop down combo boxes, and in sorting any report or list when the Test Status Type is used as the sort-by property. 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 'Type' field.
Sort Alphabetically	Reorganize the entries in the Type list into alphabetical order.
Restore Defaults	Remove all customized Test status values and restore the system defaults.

Notes

- You can transport customized Test status types between models, using the 'Configure > Model > Transfer > Export Reference Data' and 'Import Reference Data' ribbon options

Requirement Types

Requirements are critical to the success of a project, and methodologies typically specify a number of requirement types used to make sure the business, technical, architecture and transition requirements are well-defined. You define the type of requirement using 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 from your methodology or from an industry standard method..

Access

Ribbon	Configure > Reference Data > Model Types > General Types > Requirement

Customize Requirement Types

Option	Description
Requirement	Type the name of the requirement.
Description	Type a short description of the requirement.
Weight	Type a weighting to apply to the requirement type.
New	Clears the dialog fields so that you can define a new requirement type.
Save	Saves the requirement type details and adds it to the Defined Requirement Types list.
Delete	Deletes a selected requirement type from the Defined Requirement Types list.
Defined Requirement Types	Lists the currently available requirement types.

Notes

- You can transport these requirement types between models, using the 'Configure > Model > Transfer > Export Reference Data' and 'Import Reference Data' ribbon options

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

Ribbon	Configure > Reference Data > Model Types > General Types > Scenario

Customize Scenario Types

Option	Description
Scenario Type	Type the name of the scenario type.
Description	Type a short description of the scenario type.
Weight	Type a value for the weighting to apply to the scenario type.
Notes	Type any additional information to describe the scenario type.
New	Clears the dialog fields so that you can define a new Scenario type.
Save	Saves the scenario type and adds it to the Defined Scenario Types list.
Delete	Deletes a selected scenario type from the Defined Scenario Types list.
Defined Scenario Types	Lists the currently available scenario types to offer in the drop-down list of a 'Scenario Type' field.

Notes

- You can transport customized scenario types between models, using the 'Configure > Model > Transfer > Export Reference Data' and 'Import Reference Data' ribbon options

Maintenance Status Types

Model Maintenance items such as Changes, Issues, Defects and Tasks pass through a series of stages whereby each item is raised, assessed, approved, worked on, reviewed and completed. As an item passes through each stage, it can be assigned a Status from this system-provided list:

- Active
- Complete
- Deferred
- Merged
- New (the initial default status)
- No Action
- Paused
- Rejected
- Rework
- Verified

The Maintenance Status values are defined on the 'Maintenance Status' page of the 'General Types' dialog. You can add to or replace any of these values with your own custom values.

Access

Ribbon	Configure > Reference Data > Model Types > General Types > Maintenance Status

Customize Maintenance Status Types

Option	Description
New	Click on this button to clear the fields to define a new Maintenance status.
Maintenance	Type the name of the Maintenance status to define.
Make Default	To make the selected status the default for all Maintenance item 'Status' fields, select the checkbox.
Save	Save the Maintenance status item and add it to the 'Type' list.
Delete	Remove the currently selected Maintenance status from the 'Type' list.
Type	Lists the currently available Test status types.
↑ ↓	Move the selected type higher or lower in the 'Type' list. This defines the ordering used in the Maintenance 'Status' drop down combo boxes, and in sorting any report or list when the Maintenance Status Type is used as the sort-by property.

	For example: you have set up a number of Change items on a Package, and you run a report on the Change items; then you sort on the 'Status' column. The report entries are grouped by status, in the order you defined in the 'Type' list.
Sort Alphabetically	Reorganize the entries in the 'Type' list into alphabetical order.
Restore Defaults	Remove all customized Maintenance status values and restore the system defaults.

Notes

- You can transport customized Maintenance status types between models, using the 'Configure > Model > Transfer > Export Reference Data' and 'Import Reference Data' ribbon options

Risk Types

Using Enterprise Architect you can add a risk type to the global list of risk types that can be assigned to any element in the model. The global list of risk types displays in the 'Type' field drop-down list on the 'Risks' page of the **Project Indicators window**.

Access

Open the **Project Indicators window** using one of the methods outlined here, then select the 'Risk' page.

Ribbon	Configure > Reference Data > Model Types > Project Indicators : Risk

Add a new risk type to the global list

Click on the **New button** to define a new Risk

Option	Action
Risk Type	Type the name of the risk type. (Or, to edit an existing risk type, click on the risk name in the 'Defined Risks' panel.)
Description	Type a short description of the risk type.
Weight	Type the default weighting to apply to the risk type.
<note>	Type any additional information on the risk type.
Save	Click on this button to save the changes that you have made to the dialog.
New	Click on this button to clear the data fields ready to define a new risk type.
Delete	Click on an entry in the 'Defined Risks' panel, and click on this button to immediately delete the risk type.
Close	Click on this button to close the 'Project Indicators' dialog. If you have not saved your changes, the system prompts you to save or abort those changes.

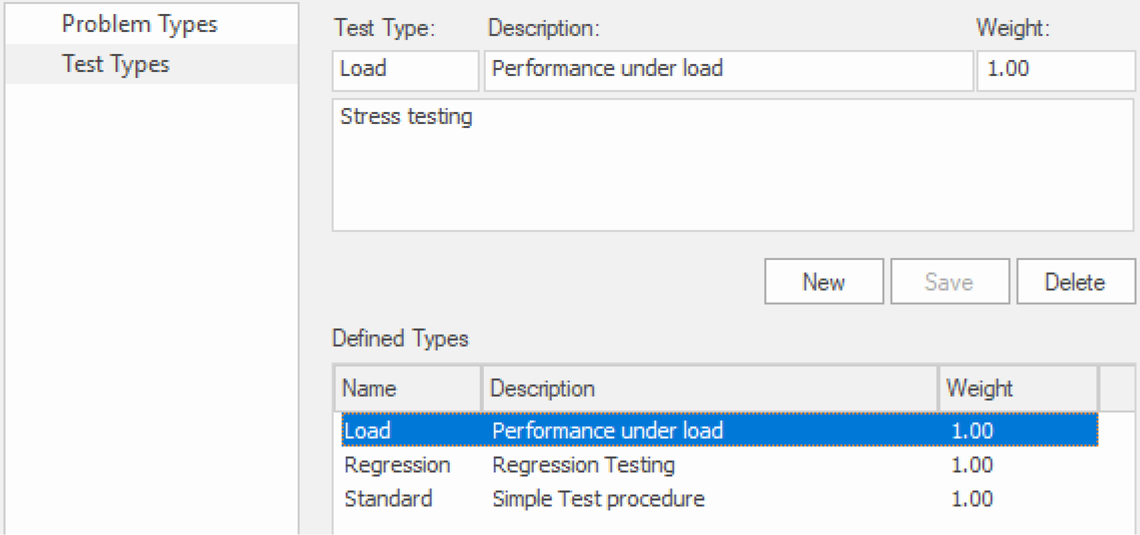
Notes

- 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 the 'Configure > Model > Transfer > Export Reference Data' and 'Import Reference Data' ribbon options

Maintenance

Within your project, you can set up Maintenance types that determine the base Problem Types to be used within the model. These include the following:

- Risk Types
- Test Types



Test Type: Load

Description: Performance under load

Weight: 1.00

Stress testing

Defined Types

Name	Description	Weight
Load	Performance under load	1.00
Regression	Regression Testing	1.00
Standard	Simple Test procedure	1.00

Access

Ribbon	Configure > Reference Data > Model Types > Maintenance

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

Ribbon	Configure > Reference Data > Model Types > Maintenance > Problem Types

Define Problem Types

Field	Action
Problem Type	Type in the name of the problem type.
Description	Type a short description of the problem type.
Weight	Provide a weighting to apply to the problem type.
Note	Type any additional information on the problem type.
Defined Types	Displays all of the pre-defined and saved problem types.
New	Click on this button to clear the dialog fields so that you can add a new problem type.
Save	Click on this button to save the scenario details and add it to the 'Defined Types' list.
Delete	Click on this button to delete a selected scenario from the 'Defined Types' list.

Notes

- You can transport these problem types between models, using the 'Configure > Model > Transfer > Export Reference Data' and 'Import Reference Data' ribbon options
- You transport the problem types together with test types as a Maintenance Types file

Testing Types

You can 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, regression tests and standard tests.

Access

Ribbon	Configure > Reference Data > Model Types > Maintenance > Test Types

Reference

Field	Action
Test Type	Type the name of the test type.
Description	Type a short description of the test type.
Weight	Type a weighting to apply to the test type.
Note	Type any additional information on the test type.
Defined Types	Displays all of the pre-defined and saved test types.
New	Add a new test type.
Save	Saves the scenario details and adds it to the 'Defined Types' list.
Delete	Deletes a selected scenario from the 'Defined Types' list.

Notes

- You can transport these test types between models, using the 'Configure > Model > Transfer > Export Reference Data' and 'Import Reference Data' ribbon 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

Metrics and Estimation

You can calculate the size of a planned software centric project by using a system of measurement called Use Case Points based on the fact that a set of Use Cases will specify the required functionality for a system. The method uses two important factors in the calculation:

- Environmental Complexity Factors (ECF)
- Technical Complexity Factors (TCF)

You specify a weighting for each factor to indicate how important the factor is in the overall measurement. You can also add any number of new factors or change or delete existing ones as required. These factors are specified and maintained in '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 and Project Resources topics, or specifically:

- Technical Complexity Factors
- Environment Complexity Factors
- Default Hours
- Effort Types
- Metric Types
- Risk Types

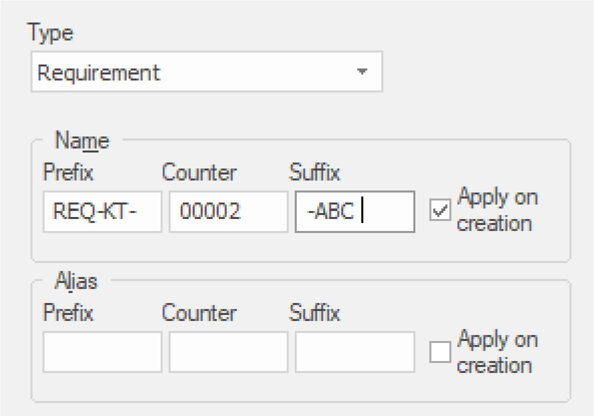
Auto Names and Counters

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 **Browser window**, 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 Browser window, the 'New Element' dialog **Auto button** applies the naming convention to your new element



Type

Requirement

Name

Prefix	Counter	Suffix	
REQ-KT-	00002	-ABC	☑ Apply on creation

Alias

Prefix	Counter	Suffix	
			□ Apply on creation

Auto Name and Counter window showing an example name and counter for new Requirements.

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.

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

Ribbon	Configure > Reference Data > Settings > **Auto Names and Counters**

Set up auto naming for an element type

Option	Action
Type	Click on the **drop-down arrow** and select the element type to which to apply auto naming/auto numbering.
Name Prefix	Type a prefix for the new names (optional).

Name Counter	Type the counter value; use as many 0's as required to pad the number.
Name Suffix	Type a suffix for the new names (optional).
Name Apply on creation	Select the checkbox to apply auto naming to each new element of the specified type, on creation. If you do not select the checkbox (but click on Save), 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 (<element type><n>).
Alias Prefix	Type a prefix for the new Aliases (optional).
Alias Counter	Type the counter value; use as many 0's as required to pad the number.
Alias Suffix	Type a suffix for the new Aliases (optional).
Alias Apply on creation	Select the checkbox to apply Alias auto naming to each new element of the specified type, on creation. If you do not select the checkbox (but click on the **Save button**), 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 (<element type><n>).
Save	Click on this button to preserve the values you have entered. New elements of this type now have an automatically-generated name and/or Alias with an incrementing counter value. • If the user creates an element on a diagram, such as from the **Diagram Toolbox** or Quick Linker, the element is automatically assigned the defined name and/or Alias • If the user creates an element outside a diagram (such as in the **Browser window**) using the 'New Element' dialog, they can click on the **Auto button** next to the 'Name' and/or 'Alias' field on the appropriate dialog to apply the auto-generated text • If the user already has some text in the 'Name' or 'Alias' field, it is over-written by the auto-counter text If an Alias is active, to view the Alias in a diagram requires that the option 'Use Alias if Available' is selected in the 'Properties' dialog for the diagram.

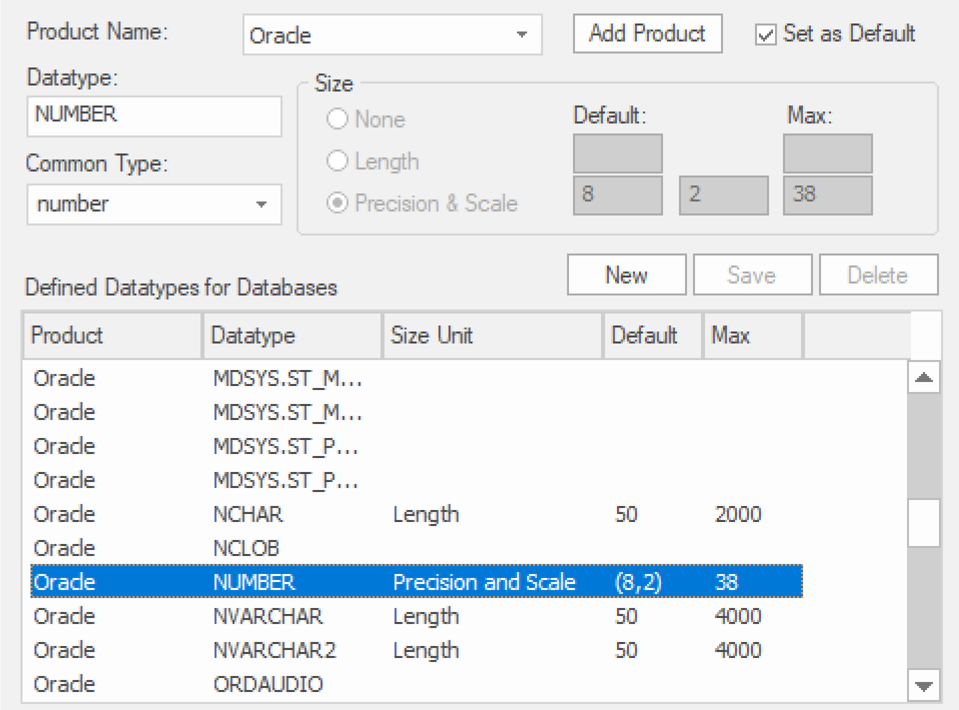
Notes

- 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

Data Types and Namespaces

There is rich support in the tool for modeling database schemas and programming constructs. The tools supports a wide variety of database systems and programming languages each of which have a set of pre-defined data types that specify and constrain the information that the system will contain.

Enterprise Architect supports the definition and management of these data types and contains lists of each system's data types. When adding database objects such as Tables or Views a modeler can specify the type of RDBMS that is being modeled and the then when adding Columns the list of data types will be available for selection. A team will typically specify a default RDBMS system and this be the value used when a new database object is created.



Product Name: Oracle

☑ Set as Default

Datatype: NUMBER

Common Type: number

Size
○ None
○ Length
● Precision & Scale

Default: 8 2

Max: 38

Defined Datatypes for Databases

Product	Datatype	Size Unit	Default	Max
Oracle	MDSYS.ST_M...			
Oracle	MDSYS.ST_M...			
Oracle	MDSYS.ST_P...			
Oracle	MDSYS.ST_P...			
Oracle	NCHAR	Length	50	2000
Oracle	NCLOB			
Oracle	NUMBER	Precision and Scale	(8,2)	38
Oracle	NVARCHAR	Length	50	4000
Oracle	NVARCHAR2	Length	50	4000
Oracle	ORDAUDIO			

The same applies to programming languages, when adding a Class a modeler specifies the programming language and then when working with attributes or operations the applicable language specific data types will be available.

Data Types

Every Table column that you define in your data model has a data type assigned that specifies the type of information that can be stored by the column. The available datatypes for a column are dependent on the selected DBMS for the Table, because each DBMS supports its own list of datatypes. Whilst each DBMS supports the same basic types, such as string, whole or decimal numbers, each DBMS calls them by different names and have different properties.

Each Enterprise Architect repository contains the definitions of the core datatypes for a number of standard DBMS products. However, since data types vary from one DBMS product to another, and from one version of a product to another, 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
- Import and export datatypes between repositories

Coding 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

Ribbon	Configure > Reference Data > Settings > Coding Datatypes or Develop > Preferences > Options > Configure Code Engineering Datatypes

Programming Language Datatype Options

Option	Description
Product Name	Click on the **drop-down arrow** and select the name of the programming language.
Add Product	Click on this button to display a prompt for the name of a new programming language. Type the name and click on the **OK button**.
Datatype	Type the language-specific name of the datatype.
Common Type	Type the common (or generic) name of the datatype; for example, the Java Boolean datatype has a common datatype Boolean.
New	Click on this button to clear the fields to create a new datatype.
Save	Click on this button to save the newly created or updated datatype.
Delete	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.
Defined Datatypes for Programming Languages	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 'Configure > Model > Transfer > Export Reference Data' and 'Import Reference Data' ribbon options
- In the Corporate, Unified and Ultimate Editions of Enterprise Architect, if security is enabled you must have 'Configure Datatypes' permission to update and delete data types

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, click on the Package in the **Browser window** and select the 'Develop > Preferences > Options > Set as Namespace Root' ribbon option. The Package icon in the Browser window changes to show a colored corner indicating this Package is 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 the Class in the Package hierarchy below the namespace root.

To clear an existing namespace root, click on the namespace root Package in the Browser window and deselect the 'Develop > Preferences > Options > Set as Namespace Root' ribbon option

To view a list of namespaces, select the 'Configure > Reference Data > Settings > Namespace Roots' ribbon option; the 'Namespaces' dialog displays. If you double-click on a namespace in the list, the Package is highlighted in the Browser window; alternatively, right-click on the namespace and select the 'Locate Package in Browser' option.

You can also clear the selected namespace root by selecting the 'Clear Namespace Attribute' option.

To omit a subordinate Package from a namespace definition, select the 'Develop >Preferences > Options > Suppress Namespace' ribbon option; to include the Package in the namespace again, deselect the ribbon option.

Notes

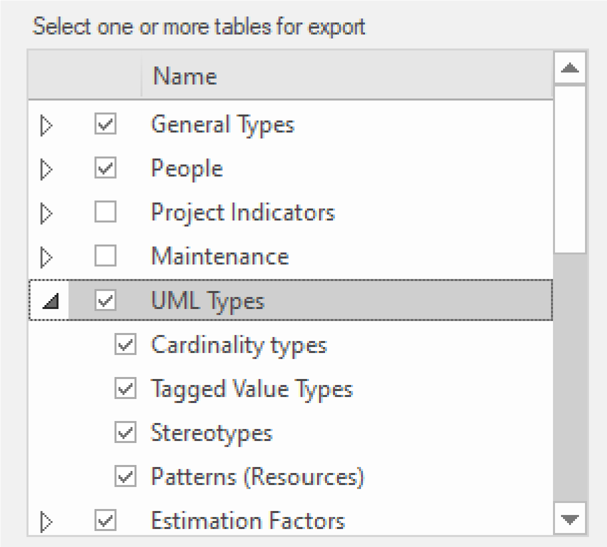
- When performing code generation, any Package name that contains whitespace characters is automatically treated as a namespace root

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.



Select one or more tables for export

Name

- ☑ General Types
- ☑ People
- □ Project Indicators
- □ Maintenance
- ☑ UML Types
  - ☑ Cardinality types
  - ☑ Tagged Value Types
  - ☑ Stereotypes
  - ☑ Patterns (Resources)
- ☑ Estimation Factors

Alternatively if your repository is MySQL, MS SQL Server or Oracle it is possible to configure your repository to use the Reference Data defined in another repository (on the same server), see Linking Reference Data for details.

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

Ribbon	Configure > Model > Transfer > Import Reference Data > Import File Configure > Model > Transfer > Import Reference Data > Shared File

Import reference data manually (Import File)

Step	Action
1	On the 'Import Reference Data' dialog, click on the 'Import File' tab and on the **Select File** button, then select the filename to import data from. This would be an XML file produced by the **Enterprise Architect Data Exporter**.
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 **Ctrl** or **Shift** to click on multiple tables.
4	Click on the **Import 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
1	On the 'Import Reference Data' dialog, click on the 'Shared File' tab.
2	If you are changing an existing configuration to import from a different XML file, click on the **Clear button** to clear the dialog fields.

3	Click on the **Select File** 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 **Ctrl** or **Shift** as you click on multiple tables.
5	If you prefer to control whether or not the automatic import takes place, select the 'Always prompt before import' checkbox.
6	Click on the **Import 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, Library 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

Concerning Images in Microsoft SQL Server Repositories:

- Importing images as reference data into a SQLServer repository is not supported when connected via Pro Cloud Server
- To import images as reference data you must have additional MS SQL Server permissions, so that you can perform 'Set Identity On/Off' commands; the role of db_ddladmin provides the required permissions; however, this approach might not be allowed in your environment, so discuss this requirement with your DBA to find an appropriate solution

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

Ribbon	Configure > Model > Transfer > Export Reference Data

Select and Export Reference Data

Step	Action
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. If you select the top-level, group-name checkbox, all tables in that group are selected for export.
3	Click on the **Export button**.
4	When prompted to do so, enter a valid file name with a .XML extension.
5	Click on the Save and OK 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

Linking Reference Data

If you have a number of projects that use the same reference data, you can use a central repository containing all common data and configure the other repositories to link directly to this data. This data includes system resources such as Security Users and Groups, Permissions, Scripts, Glossary terms or Data Types. Sharing common data between repositories makes it possible for teams working on multiple projects to leverage a common and consistent set of system resources, thereby promoting or enforcing common usage and standards.

Sharing common data can be applied to both existing and new projects, however each individual repository can be configured to use as much of the common data as is deemed necessary.

The ability to link to common data in a central repository uses the native features provided by the various Database Management System (DBMS), MySQL, SQL Server and Oracle. Meaning that it is configured at the database level. The benefit of this is that Enterprise Architect will function normally without needing any modifications in order to access the data, in fact it is not even aware that the common data is physically located in another repository.

Access

Ribbon	Configure > Model > Transfer > Shared Repository

Configure a Repository to use shared data

The 'Shared Repository' screen provides an Administrator the ability to generate a SQL script that can be run on the current repository's database and configure it to share common data of a centralized repository. The generated script is tailored based on user selection and the DBMS of the current repository.

The process to configure the current repository to share a centralized repository's data, is:

Step	Action
1	In 'Script File' field, enter the complete filename and path of the script, or click on the ... button and search for it.
2	In the 'External Repository' field enter the database name of the centralized repository. Enterprise Architect will only continue if the entered database name exists on the Database Server hosting the Current Repository.
3	Select which Table Group(s) will be shared, as explained in this table.
4	Click the **Generate button** to create the script and save it to the specified location.
5	Open the SQL console of your choice and execute the generated SQL script. Since the schema of the database is being updated an elevated level of permissions are required to execute these statements.

Select Table Groups

There are approximately 30 tables that can be potentially shared between repositories, the 'Shared Repository' screen organizes these tables into logical Groups. While some Table Groups only contain 1 table, others contain multiple.

You can select all Table Groups or as many as is appropriate for the current repository.

Option	Action
Security Users and Groups	This group of tables contains a number of tables that define the Model Users, the Security Groups and the Security Groups each Model User belongs to.
Security Permissions	This group of tables contains the security permissions that are assigned to each Model User and Security Group. While at first glance it might seem strange that the Security User and Group does not include Security Permissions, it was a deliberate decision to keep these two separate because it provides a greater level of flexibility.
People	This group of tables contains the definition of all people/resources in the repository.
Glossary	This group contains the Glossary table that defines the glossary terms for the repository.
Types	This group of tables contains the definition of reference types that are typically used to populate drop down lists within Enterprise Architect's screens. Types like Complexity, Constraints, Effort, Problem, Property, Risk, etc.
Complexity Factors	This group of tables contains the definition of complexity factors, like ECF and TCF.
Scripts	This group contains the Script table that defines all custom JSScript, JavaScript and VBScript.
Data Types	This group contains the Data Types table that defines both the data types for all code languages and DBMSs.

How it works

The scripts generated by this screen work by renaming each Enterprise Architect Table that has been selected and then creating a Synonym (or View) that references the centralized database's table of the same name.

Consider the example that the current repository/database called EA1 should be configured to share EAMAIN's T_GLOSSARY table.

The script will:

- renamed table EA1.T_GLOSSARY to EA1.T_GLOSSARY_BAK

- create a new synonym (or view) call EA1.T_GLOSSARY that references EAMAIN.T_GLOSSARY

Both Microsoft SQL Server and Oracle support database Synonyms, which are a mechanism that provides an alternative name for another database object, in the case of shared data the other database object is a table in another database. MySQL does not support Synonyms, however a View can be used to achieve the same behavior.

Removing Shared data

As described in the 'How it works' section, since the scripts generated by this screen simply rename the original tables by appending "_BAK" to their names, it is a relative easy process to rename (or delete) the synonym (or view) and then rename the table back its original name.

Notes

- Enterprise Architect can only generate Shared Repository scripts for MySQL, SQL Server and Oracle based DBMS repositories
- In order to share common data all repositories must use the same DBMS and must be on the same network, preferable on the same database server (for performance reasons); that is, if the central repository is MySQL then all repositories that will use the common data must be MySQL and have direct network access
- The Shared Repository screen cannot be opened while the current connection is using a Cloud connection, because Cloud-based connections do not provide the necessary database information needed to generate a working SQL script; however, once the script has been generated (while connected with a direct ODBC connection) and then executed, Cloud-based connections will read shared data in the same way as their direct connection counter-parts

Shared Security Reference Data

When deciding which Table Groups of the centralized repository each individual repository should share, the Security related tables should be carefully considered, and the decisions based on a number of factors:

- How are users going to be validated - via an external Single Sign-On (SSO) mechanism (like OpenID or Windows NTLM) or via a simple User ID and password stored within the Enterprise Architect repository?
- Should all repositories share the same list of users?
- Should user passwords be synchronized across all repositories?
- Do all users have the same permissions across all repositories?

There are two Shared Table Groups related to Security: 1) 'Security Users and Groups' and 2) 'Security Permissions'. Whether a repository should be configured to share either or both of these Table Groups depends on the environment. Consider these scenarios:

Scenario 1. All repositories have the same Users, and all Users have the same permissions within the Repositories; the individual repository should share both Table Groups: 'Security Users and Groups' and 'Security Permissions' from the centralized repository. The user list and permissions will need to be managed within the centralized repository.

Scenario 2. All repositories have the same users, but the Users have different permissions across the Repositories; the individual repository should share the Table Groups: 'Security Users and Groups' from the centralized repository. The user list will need to be managed within the centralized repository while the user permissions will need to be managed within the individual repositories.

With each of the above scenarios it is still a requirement that the passwords for Users are maintain within the individual repositories; the only way to avoid this is to implement SSO authentication or to use Pro Cloud Server's Global Authentication.

Authentication via SSO

When using an SSO authentication mechanism, Enterprise Architect is no longer responsible for determining if the entered user credentials are valid, instead it is OpenID or Windows NTLM that validates the user credentials and advises Enterprise Architect that the given user is valid. Enterprise Architect then uses the supplied UserID to determine what features are available within the current repository. This requires that the complete list of Users is defined within the repository, however if each individual repository is configured to share the Security related reference data the list of Users (and their permissions) can be defined and managed in the centralized repository.

Authentication via Global Authentication

It's possible to configure the Ports of the Pro Cloud Server to use a 'Global' repository for authentication purposes. When this option is set the initial log in process uses the 'Global' repository to validate the supplied user credentials, once those credentials have been confirmed any further security related queries use the information saved in the current repository. The benefit of this configuration means that each repository has the ability for the same user to have different permissions in each repository. However the downside is that the security user definitions must be manually kept up-to-date in each repository.

Using Shared Repository data (as described in the topic Link Reference Data to a Shared Repository) it is possible to remove the duplicated data by choosing certain Table Groups to shared and the remaining to be stored locally, the exact configure will depend on the individual environment.

More Information

Project Management

Project Resources

Edition Information

- In the Corporate, Unified and Ultimate Editions of Enterprise Architect, if security is enabled you must have 'Manage Reference Data - Update' permission to update and delete reference items