



MDG Technology for DDS User Guide

Welcome to the MDG Technology for DDS User Guide. The MDG Technology for DDS enables you to work simultaneously with both Enterprise Architect and DDS and merge the changes with minimal effort.



© Copyright 2007-2009 Sparx Systems Pty Ltd

MDG Technology for DDS User Guide

Introduction

by Simon Zhang

MDG Technology for DDS provides a lightweight bridge between Enterprise Architect and DDS

MDG Technology for DDS User Guide

© 2007- 2009 Sparx Systems Pty Ltd

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

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

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

Printed: March 2009

Publisher

Sparx Systems

Managing Editor

Geoffrey Sparks

Technical Editor

Simon Zhang

Special thanks to:

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

Table of Contents

Foreword	1
Welcome	2
Copyright Notice	3
Software Product License Agreement	4
Acknowledgement of Trademarks	7
Support	8
System Requirements	9
Getting Started	10
Register the MDG Technology for DDS	11
Configure the DDS Modeling Environment	13
Manage DDS Technology	14
Using the MDG Technology for DDS	15
DDS Technology Menu	16
DDS Options	17
General Options	17
Qos Policy Specifications Dialog.....	18
RTIDDS Options	18
OSPLDDS Options	20
DDS Toolboxes	22
Domain	22
Local Reconstruction	23
Types	25
QoS	26
Application Target	28
Create DDS Project	30
Using a Template	30
Without a Template	30
Create DCPS Model.....	31
Create DLRL Model.....	34
Model Validation	38
Model Validation Rules for Elements	38
Generate PSM and Code	40
DDS Menu Command	40
EA Transformation	41
DDS Example Model	45
Index	47

Foreword

MDG Technology for DDS provides a light weight bridge between Enterprise Architect and DDS.

1 Welcome



Welcome to the *MDG Technology for Data Distribution Service (DDS)* - Enterprise Architect MDG Add-In, Version 2.0.

The Add-In extends the capability of Enterprise Architect to enable you to create Data Distribution Service (DDS) models. The language specification for DDS is available from the Object Management Group (OMG) website: [Data Distribution Service for Real-time Systems Specification](#)

Getting Started

For instructions on how to install the MDG Technology for DDS, see [Getting Started](#)¹⁰.

See Also

- [Copyright Notice](#)³
- [Trademarks](#)⁷
- [Support](#)⁸
- [License Agreement](#)⁴
- [System Requirements](#)⁹

1.1 Copyright Notice

Copyright © 2007-2009 Sparx Systems Pty. Ltd. All rights reserved.

The *MDG Technology For DDS* software contains proprietary information of Sparx Systems Pty Ltd. It is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright law. Reverse engineering of the software is prohibited. Please read the [license agreement](#) ⁴ for full details.

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

Sparx Systems Pty. Ltd.

7 Curtis St,
Creswick, Victoria 3363,
AUSTRALIA

Phone: +61 (3) 5345 1140

Fax: +61 (3) 5345 1104

Support Email: support@sparxsystems.com

Sales Email: sales@sparxsystems.com

Website: <http://www.sparxsystems.com>

1.2 Software Product License Agreement

MDG Technology for DDS - Enterprise Architect MDG Add-In

Copyright (C) 2007-2009 Sparx Systems Pty Ltd. All Rights Reserved

IMPORTANT-READ CAREFULLY: This End User License Agreement (EULA) is a legal agreement between YOU as Licensee and SPARX for the SOFTWARE PRODUCT identified above. By installing, copying, or otherwise using the SOFTWARE PRODUCT, YOU agree to be bound by the terms of this EULA.

If YOU do not agree to the terms of this EULA, promptly delete the unused SOFTWARE PRODUCT.

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

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

DEFINITIONS

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

- "EULA" means this End User License Agreement
- "SPARX" means Sparx Systems Pty Ltd A.C.N 085 034 546
- "Licensee" means YOU, or the organization (if any) on whose behalf YOU are taking the EULA.
- "Registered Edition of **MDG Technology for DDS**" means the edition of the SOFTWARE PRODUCT which is available for purchase from the web site: (http://www.sparxsystems.com/ea_purchase.htm). Following the thirty day free evaluation period.
- "SOFTWARE PRODUCT" or "SOFTWARE" means **MDG Technology for DDS**, which includes computer software and associated media and printed materials, and may include online or electronic documentation.
- "Trial edition of **MDG Technology for DDS**" means the edition of the SOFTWARE PRODUCT which is available free of charge for evaluation purposes for a period of 30 days.

GRANT OF LICENSE

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

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

EVALUATION LICENSE

The Trial Version of **MDG Technology for DDS** is not free software. Subject to the terms of this agreement, YOU are hereby licensed to use this software for evaluation purposes without charge for a period of 30 days.

Upon expiration of the 30 days, the SOFTWARE PRODUCT must be removed from the computer. Unregistered use of **MDG Technology for DDS** after the 30-day evaluation period is in violation of Australian, U.S. and international copyright laws.

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

If YOU choose to use this software after the 30 day evaluation period a license must be purchased (as described at http://www.sparxsystems.com/ea_purchase.htm). Upon payment of the license fee, YOU will be sent details of where to download the registered edition of **MDG Technology for DDS** and will be provided with a suitable software 'key' by email.

ADDITIONAL RIGHTS AND LIMITATIONS

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

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

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

ASSIGNMENT

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

TERMINATION

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

WARRANTIES AND LIABILITY

WARRANTIES

SPARX warrants that the SOFTWARE PRODUCT will perform substantially in accordance with the accompanying written materials for a period of ninety (90) days from the date of receipt, and any Support Services provided by SPARX shall be substantially as described in applicable written materials provided to YOU by SPARX, and SPARX support engineers will make commercially reasonable efforts to solve any problems associated with the SOFTWARE PRODUCT.

EXCLUSIONS

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

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

LIMITATION

The SOFTWARE PRODUCT and any documentation are provided "AS IS" and all warranties whether express, implied, statutory or otherwise, relating in any way to the subject matter of this EULA or to this EULA generally, including without limitation, warranties as to: quality, fitness; merchantability; correctness; accuracy; reliability; correspondence with any description or sample, meeting your or any other requirements; uninterrupted use; compliance with any relevant legislation and being error or virus free are excluded. Where any legislation implies in this EULA any term, and that legislation avoids or prohibits provisions in a contract excluding or modifying such a term, such term shall be deemed to be included in this EULA. However, the liability of SPARX for any breach of such term shall, if permitted by legislation be limited, at SPARX'S option to any one or more of the following upon return of the SOFTWARE PRODUCT and a copy of the receipt:

- If the breach relates to the SOFTWARE PRODUCT:
 - the replacement of the SOFTWARE PRODUCT or the supply of an equivalent SOFTWARE PRODUCT
 - the repair of such SOFTWARE PRODUCT; or the payment of the cost of replacing the SOFTWARE PRODUCT or of acquiring an equivalent SOFTWARE PRODUCT, or
 - the payment of the cost of having the SOFTWARE PRODUCT repaired.
- If the breach relates to services in relation to the SOFTWARE PRODUCT:
 - the supplying of the services again, or
 - the payment of the cost of having the services supplied again.

TRADEMARKS

All names of products and companies used in this EULA, the SOFTWARE PRODUCT, or the enclosed

documentation may be trademarks of their corresponding owners. Their use in this EULA is intended to be in compliance with the respective guidelines and licenses. Windows, Windows 98, Windows NT, Windows ME, Windows Vista, Windows XP and Windows 2000 are trademarks of Microsoft.

GOVERNING LAW

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

1.3 Acknowledgement of Trademarks

Trademarks of Microsoft

- Microsoft®
- Windows®

Trademarks of the OMG

- OMG™
- Object Management Group™
- UML™
- Unified Modeling Language™
- DDS™
- Data Distribution Service™

Trademarks of PrismTech Limited

- OpenSplice DDS®

Trademarks of Real Time Innovations Inc

- RTIDDS®

1.4 Support

Technical support for the MDG Technology for DDS is available to registered users of Enterprise Architect. Responses to support queries are sent by email. Sparx Systems endeavors to provide a rapid response to all product-related questions or concerns.

Registered users can lodge a support request, by visiting:
http://www.sparxsystems.com/registered/reg_support.html.

Trial users can contact Sparx Systems with questions regarding their evaluation at:
support@sparxsystems.com.

An online user forum is also available for your questions and perusal, at
<http://www.sparxsystems.com/cgi-bin/yabb/YaBB.cgi>.

1.5 System Requirements

MDG Technology for DDS runs under the following environments:

Windows Operating Systems

- Windows ME
- Windows NT® (SP5 or later)
- Windows XP Professional
- Windows XP Home
- Windows XP Media Edition
- Windows XP Tablet Edition
- Windows 2000 Professional (SP3 or later).

Enterprise Architect

- Enterprise Architect Professional Version 7.5 (or later)
- Enterprise Architect Corporate Version 7.5 (or later).

DDS Middleware

- PrismTech OpenSplice DDS 3.3 (or later)
- RTIDDS 4.3e (or later).

Notes:

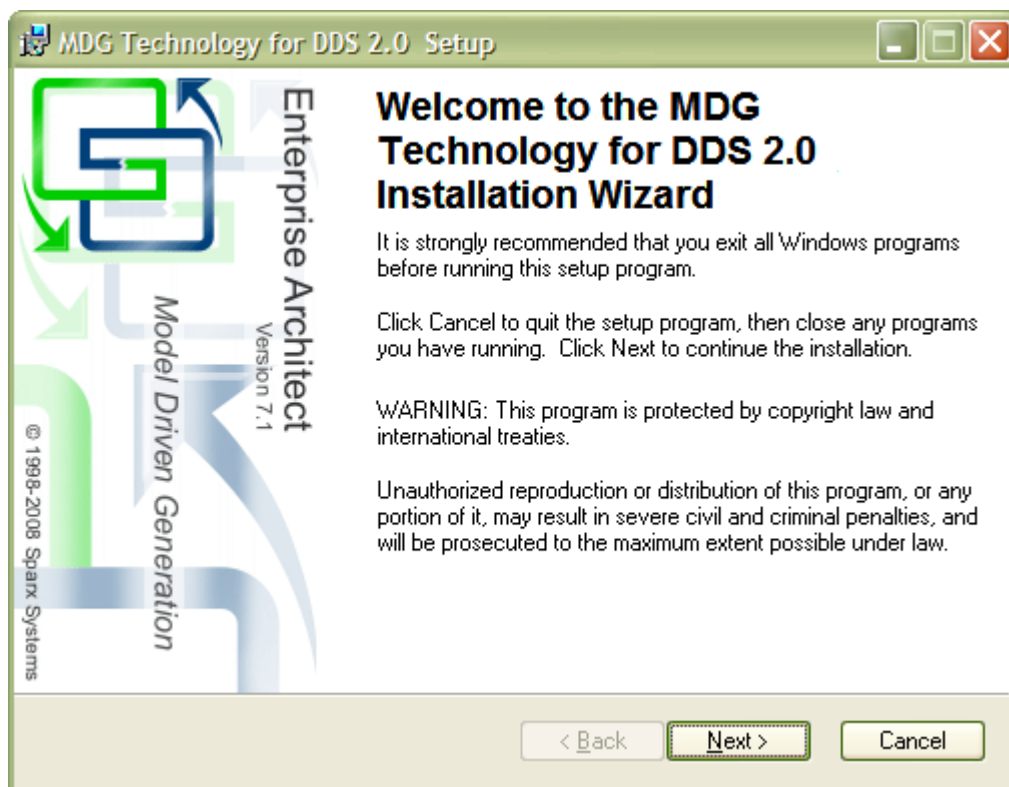
- While the MDG Technology for DDS isn't specifically blocked from executing with the Enterprise Architect Desktop Edition, be aware that the Desktop Edition cannot make use of some of the capabilities of the product, such as Model Validation..

2 Getting Started

The MDG Technology for DDS is available for download from the [Products](#) page on the Sparx Systems website. Once it has been downloaded, the program is in .exe format.

To install the MDG Technology for DDS follow the steps below:

1. Download the *EADDS.exe* file from the Sparx Systems website.
2. Ensure that you meet the [System Requirements](#)^[9] and have Enterprise Architect version 7.1 or higher.
3. To install the MDG Technology for DDS component, double-click on the *EADDS.exe* Installer executable. The **MDG Technology for DDS Installation Wizard** screen displays. Click on the **Next** button.



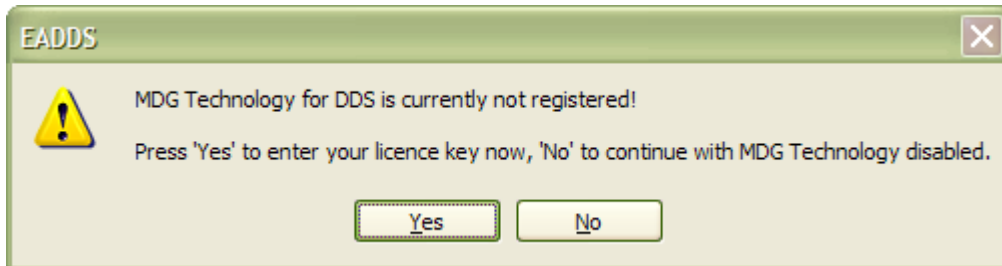
4. Read the licensing agreement and, if you accept the terms, click on the **Next** button.
5. Read the *Readme* information, and then click on the **Next** button.
6. In the **User name** and **Organization** fields, type your user and company names. Click on the **Next** button. (Optionally define the users who have access to this program by either choosing the **All users** option or registering specific users).
7. Choose an installation path for the program, and then click the **Next** button.
8. When installation is complete, click on the **Finish** button.

After the MDG Technology for DDS AddIn has been installed, you can go on to [Register the MDG Technology for DDS](#)^[11], [Configure the DDS Modeling Environment](#)^[13] and [Manage DDS Technology](#)^[14].

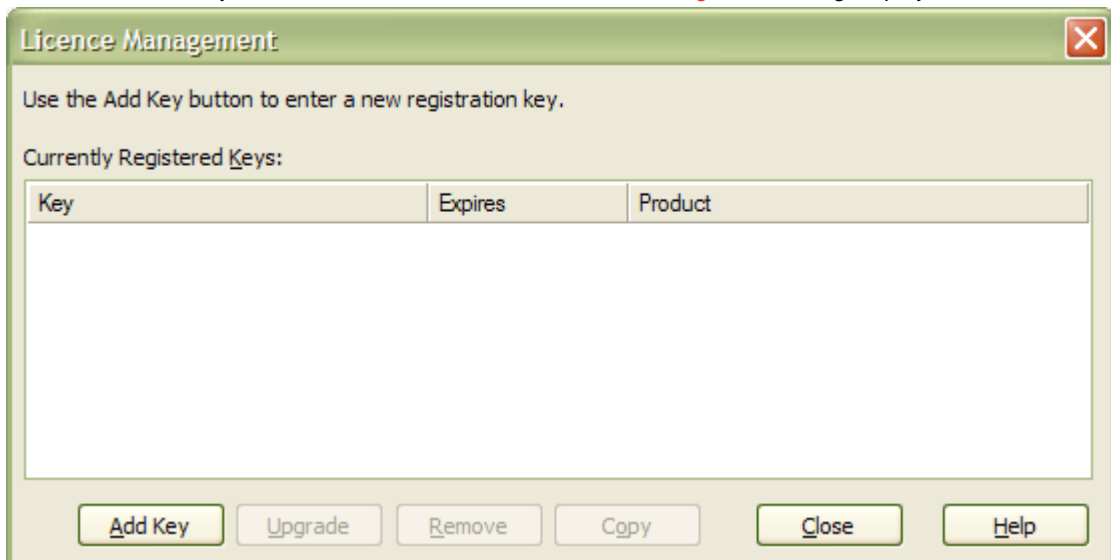
2.1 Register the MDG Technology for DDS

To register and activate the MDG Technology for DDS Add-In, follow the steps below:

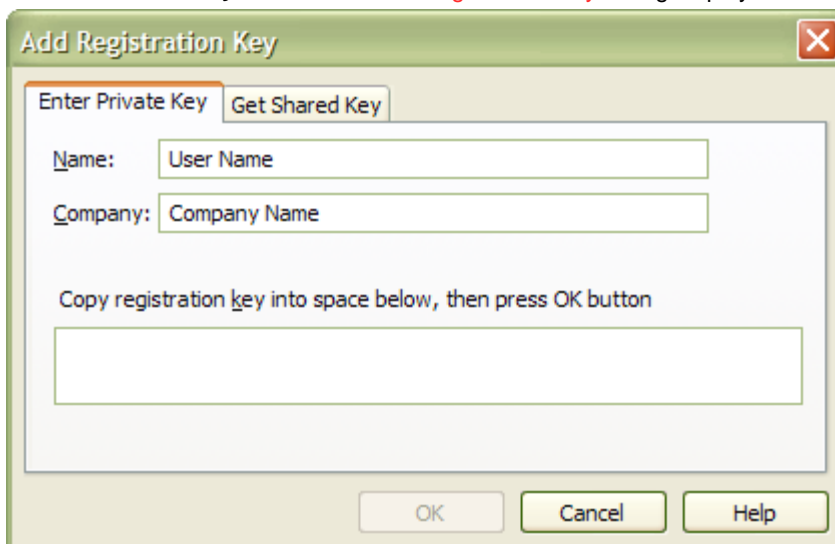
1. Purchase one or more licenses. Once you have paid for a licensed version of MDG Technology for DDS, you receive (via email or other suitable means)
 - a license key or keys
 - the address of the web site from which to download the full version.
2. Save the license key and download the latest full install package from the address supplied.
3. Run the setup program to install the full version.
4. Run Enterprise Architect. If this is the first time you have installed MDG Technology for DDS, a dialog displays prompting you to register MDG Technology for DDS or to continue the trial.



5. To enter the new key click on the **Yes** button. The **Licence Management** dialog displays.



6. Click on the **Add Key** button. The **Add Registration Key** dialog displays.



7. Copy the license key from the email and paste it into the **Copy registration key...** field, to avoid typing mistakes.
8. Click on the **OK** button. The full version of MDG Technology for DDS is now available for use with your version of Enterprise Architect.

2.2 Configure the DDS Modeling Environment

MDG Technology for DDS AddIn requires either RTIDDS or OpenSplice DDS Middleware products to be installed in the same Computer, otherwise it can not generate fully executable project based on the specific DDS Middleware.

Configuration for RTIDDS

After RTIDDS middleware is installed to the system, it will create an environment variable `NDDSHOME` which will be the directory in which the RTIDDS software installed. DDS AddIn will check this environment variable in the [RTIDDS Options](#)^[18].

It is also required to specify a *Project Root* folder in the [RTIDDS Options](#)^[18]. This is the location for all generated RTIDDS based source code.

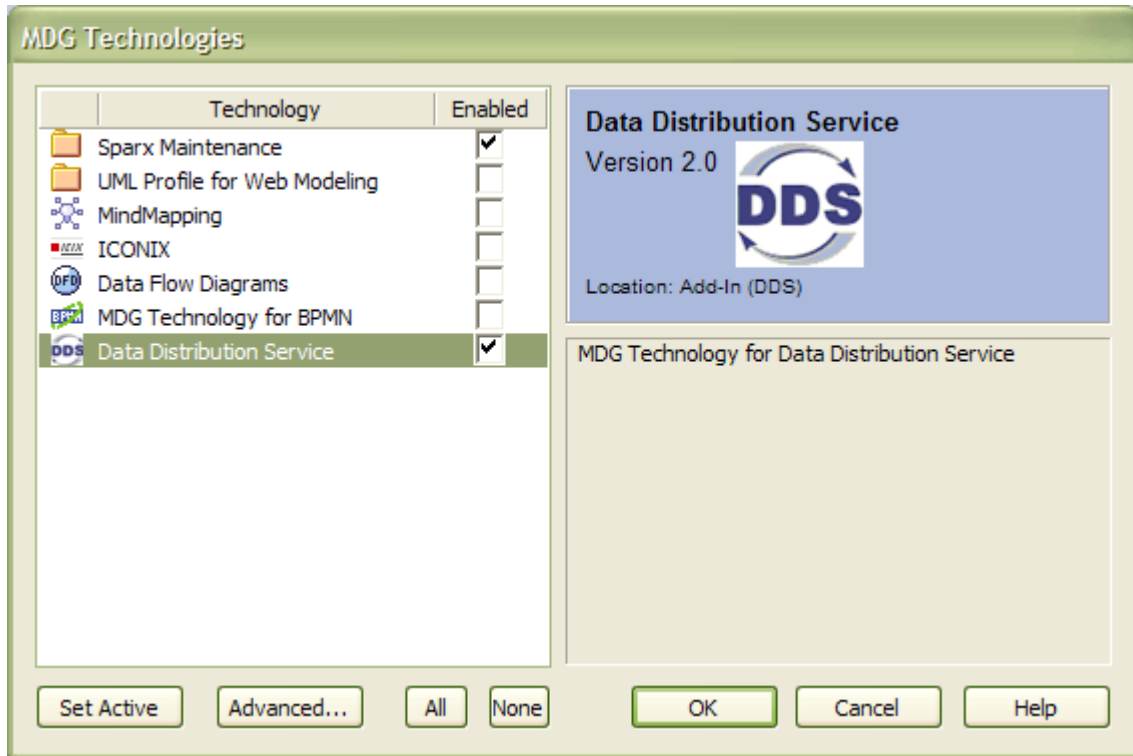
Configuration for OpenSplice DDS

After OpenSplice DDS middleware is installed to the system, it will create an environment variable `OSPL_HOME` which is the home directory in which the OpenSplice DDS software installed. DDS AddIn will check this environment variable in the [OSPLDDS Options](#)^[20].

It is also required to specify a *Project Root* folder in the [OSPLDDS Options](#)^[20]. This is the location for all generated OpenSplice DDS based source code.

2.3 Manage DDS Technology

To manage DDS Technology, select the **Settings | MDG Technologies** menu option.



The Data Distribution Service (DDS) can be made available (enabled) or removed from use (disabled). To enable or disable it, select "**Data Distribute Service**" from the list and click on its **Enabled** checkbox.

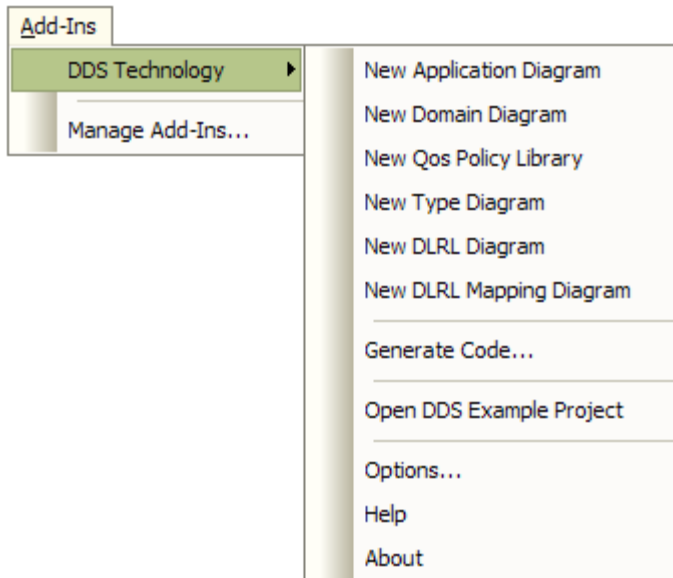
3 Using the MDG Technology for DDS

A DDS Specification and tutorial is outside the scope of this document. Rather, this document describes the following features of the MDG Technology for DDS:

- [DDS Technology Menu](#) ^[16]
- [DDS Options](#) ^[17]
- [DDS Toolboxes](#) ^[22]
- [Create DDS Project](#) ^[30]
- [Model Validation](#) ^[38]
- [Generate PSM and Code](#) ^[40]
- [DDS Example Model](#) ^[45]

3.1 DDS Technology Menu

The **DDS Technology** sub-menu is available from the **Add-Ins** menu on the main menu bar.



Menu Option	Description
New Application Diagram	Creates a new DDS Application diagram in the currently selected package. The diagram is an extended UML Deployment diagram.
New Domain Diagram	Creates a new DDS Domain diagram in the currently selected package. The diagram is an extended UML Component diagram.
New Qos Policy Library	Creates a new DDS Qos policy library package in the model.
New Type Diagram	Creates a new DDS Type diagram in the currently selected package. The diagram is an extended UML Class diagram.
New DLRL Diagram	Create a new DDS DLRL diagram in the currently selected package. The diagram is an extended UML Class diagram.
New DLRL Mapping Diagram	Create a new DDS DLRL Mapping diagram in the currently selected package. The diagram is an extended UML Class diagram.
Generate Code...	Opens the Generate Executable Class ^[40] dialog.
Open DDS Example Project	Opens the example DDS model.
Options	Opens the DDS Options ^[17] dialog.
Help	Opens this help file.
About	Displays the version information for the MDG Technology for DDS.

Tip:

If the **Add-Ins** menu or the **DDS Technology** sub-menu are not visible after installing the DDS Add-In, try to reset Enterprise Architect's menus with the **View | Visual Layouts | Default Layout** menu option.

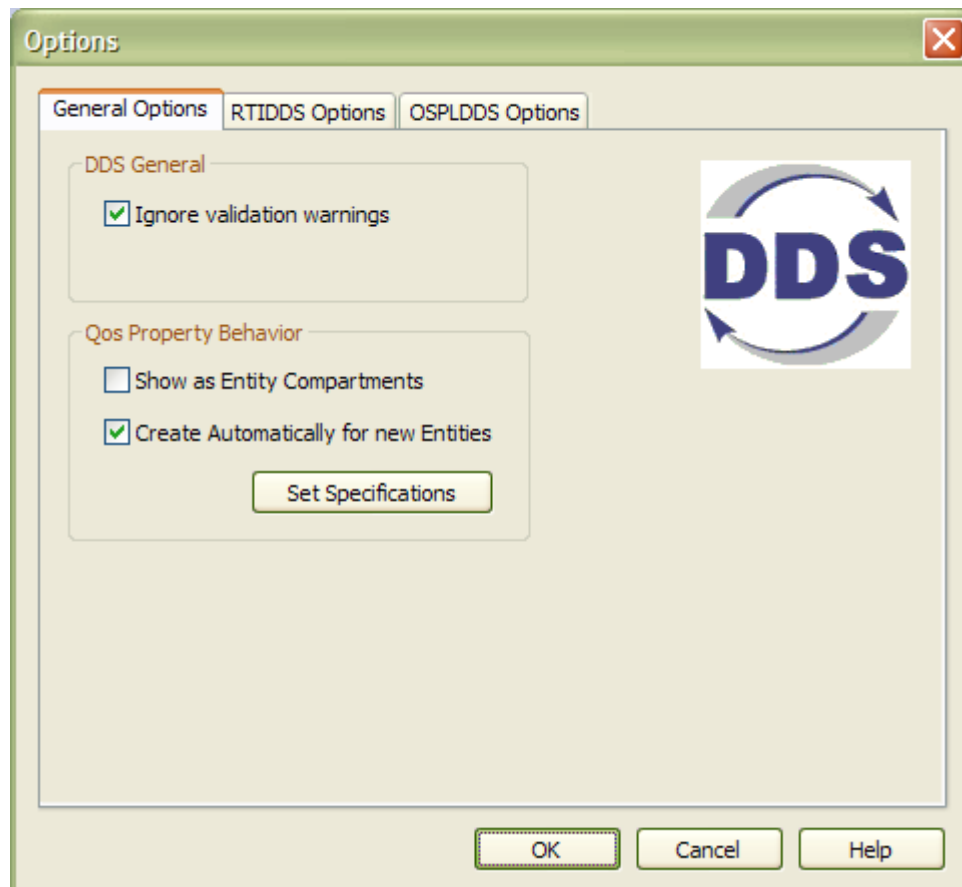
3.2 DDS Options

Select **Add-Ins | DDS Technology | Options...** menu option to display the Options dialog. There are three groups of DDS Options:

- [General Options](#) ^[17]
- [RTIDDS Options](#) ^[18]
- [OSPLDDS Options](#) ^[20]

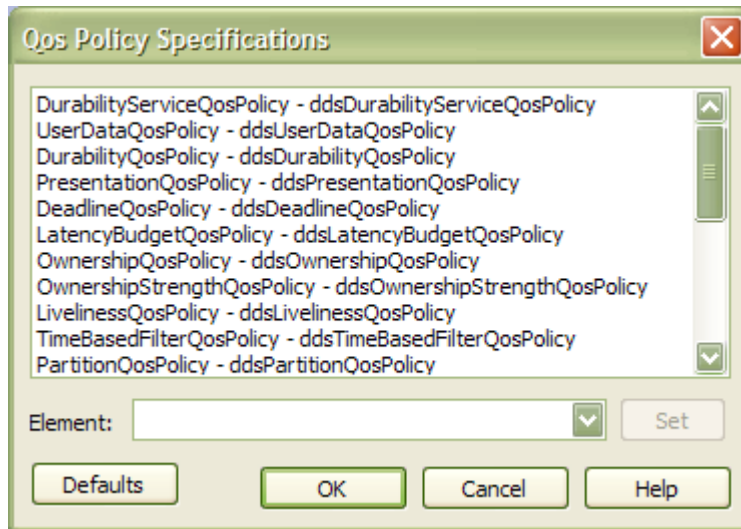
3.2.1 General Options

The options in the **General Options** tag are used to specify the general behavior and Qos properties of the MDG Technology for DDS.



Field/Button	Description
Ignore validation warnings	Select the checkbox to ignore any DDS warning messages during the model validation process.
Show as Entity Compartments	Select the checkbox to enable the Qos Policies to be displayed on DDS entities using the compartment notation.
Create Automatically for new Entities	Select the checkbox to enable the Qos Policy properties to be automatically created when new DDS Entities are created in the model. Enabling this option requires the Qos Policy specifications to be defined using the Qos Policy Specifications ^[18] dialog ^[18] .
Set Specifications...	Click on this button to open the Qos Policy Specifications ^[18] dialog ^[18] .

3.2.1.1 Qos Policy Specifications Dialog



This dialog is used to specify which *Qos Policy Specifications* are to be used as the default classifiers for *QosProperty* elements when new DDS Entities are created. Each of the policies can be specified by any one *Qos Policy Specification* defined for that Policy.

The **Defaults** button can be used to automatically enable the MDG Technology to allocate the specifications for this project.

3.2.2 RTIDDS Options

The options in the **RTIDDS Options** tag are used to specify the behavior of the [PIM to PSM transformation](#)⁴¹ for RTIDDS implementations.

Note:

The version of RTIDDS used is specified by the %NDDSHOME% environment variable.

The screenshot shows the 'Options' dialog box with the 'RTIDDS Options' tab selected. The 'Project Root' is 'C:\RTI\src\Test'. The 'General Options' section includes checked boxes for 'Enable RTIDDS PSM', 'Autoname Source Files', 'Autogenerate Source', 'Generate Listener for DDS Reader', 'Autogenerate IDL', 'Run NDDSGen on completion', and 'Generate Listener for DDS Writer'. The 'Project Files Choice' is 'Generate Workspace and Project Files' and the 'Workspace Name' is 'RTIDDSApplication1'. The 'Advanced Options' section has 'Generate Shared Files in Folder' checked with a folder name of 'SharedFileFolder'. A note indicates this option only applies when generating code from the DDS AddIn menu command. Buttons for 'OK', 'Cancel', and 'Help' are at the bottom.

Field	Description
Project Root	Root directory in which all the PSM output should be created.
Enable RTIDDS PSM	Select the checkbox to enable the transformer; deselect to disable the transformer.
Autoname Source Files	Select the checkbox to enable the RTIDDS transformer to automatically allocate filenames to the generated source code (recommended).
Autogenerate Source	Select the checkbox to enable the RTIDDS transformer to automatically invoke the code generation process to the PSM after it has been successfully generated.
Generate Listener for DDS Reader	Select the checkbox to enable the RTIDDS transformer to automatically generate a Listener for each DDS Reader.
Autogenerate IDL	Select the checkbox to enable the DDS Topic Type elements to automatically be generated to IDL code during the transformation process.
Run NDDSGen on completion	Select the checkbox to enable the RTIDDS transformer to automatically invoke the NDDSGEN command to the generated IDL, to automatically generate the type-specific DDS code for the implementation.
Generate Listener for DDS Writer	Select the checkbox to enable the RTIDDS transformer to automatically generate a Listener for the DDS Writer.
Project Files Choice	Click on the drop-down arrow and select the type of project files for the RTIDDS transformer to generate. For Visual Studio .NET, the workspace file is the Solution file.
Workspace Name	The name of the workspace. For Visual Studio .Net, this is the name of the solution.
Generate Shared Files in Folder:	Select the checkbox to enable the RTIDDS transformer to generate the common shared files to a specific folder. The specific folder is a relative

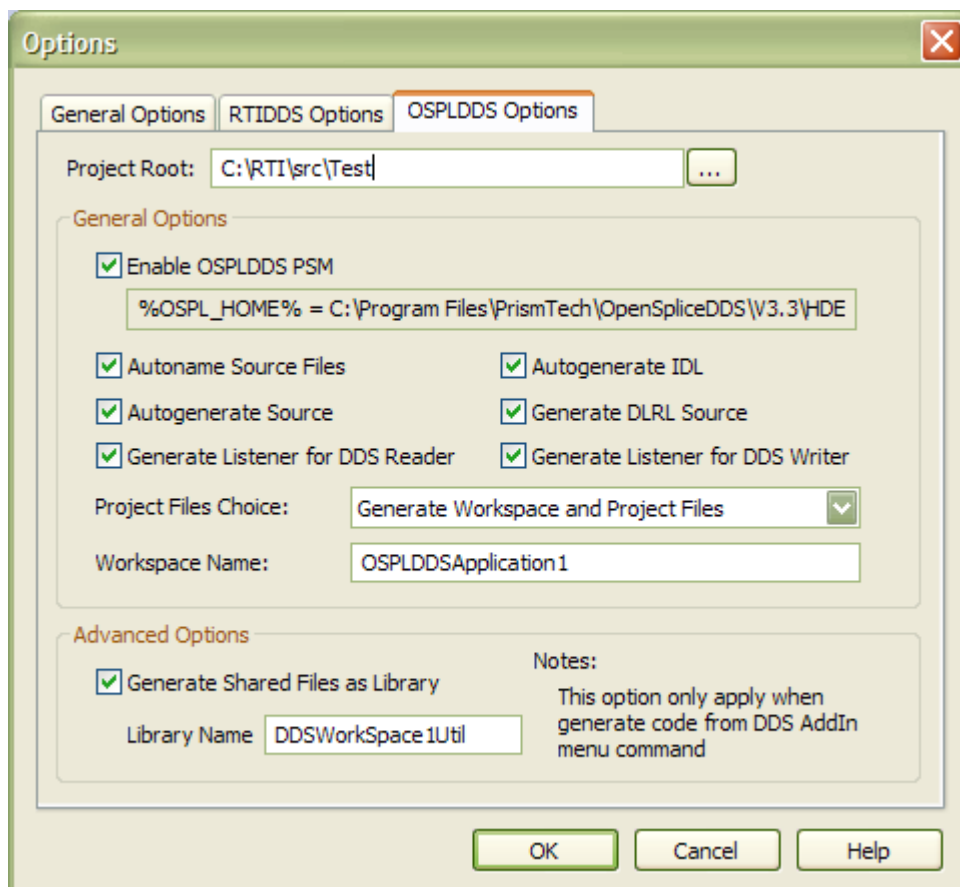
Field	Description
	path to the Project Root folder. This option only apply when it generates code from "Generate Executable Class" dialog (by select menu " Addin DDS Technology Generate Code ... " to open the dialog).

3.2.3 OSPLDDS Options

The options in the **OSPLDDS Options** tag are used to specify the behavior of the [PIM to PSM transformation](#) for OpenSplice DDS implementations.

Note:

The version of OpenSplice DDS used is specified by the %OSPL_HOME% environment variable.



Field	Description
Project Root:	Root directory in which all the PSM output should be created.
Enable OSPLDDS PSM	Select the checkbox to enable the transformer; deselect to disable the transformer.
Autoname Source Files	Select the checkbox to enable the OSPLDDS transformer to automatically allocate filenames to the generated source code (recommended).
Autogenerate Source	Select the checkbox to enable the OSPLDDS transformer to automatically invoke the code generation process to the PSM after it has been successfully generated.
Generate Listener for DDS	Select the checkbox to enable the OSPLDDS transformer to automatically

Field	Description
Reader	generate a Listener for each DDS Reader.
Autogenerate IDL	Select the checkbox to enable the DDS Topic Type elements to automatically be generated to IDL code during the transformation process.
Generate DLRL Source	Select the checkbox to enable the codes for DLRL specialized interfaces and classes to automatically be generated during the transformation process.
Generate Listener for DDS Writer	Select the checkbox to enable the OSPLDDS transformer to automatically generate a Listener for the DDS Writer.
Project Files Choice	Click on the drop-down arrow and select the type of project files for the OSPLDDS transformer to generate. For Visual Studio .NET, the workspace file is the Solution file.
Workspace Name	The name of the workspace. For Visual Studio .Net, this is the name of the solution.
Generate Shared Files as Library:	Select the checkbox to enable the OSPLDDS transformer to generate the common files as a shared library (.lib). This option only apply when it generates code from "Generate Executable Class" dialog (by select menu " Addin DDS Technology Generate Code ... " to open the dialog).

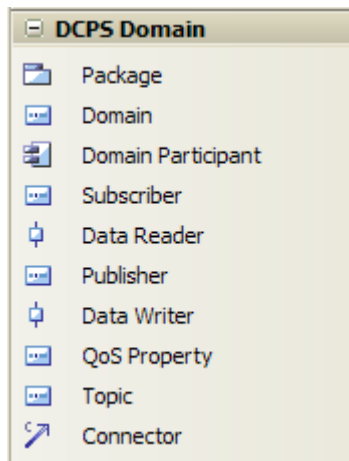
3.3 DDS Toolboxes

This section describes the toolboxes provided with the MDG Technology for DDS.

- [Domain](#) ^[22]
- [Local Reconstruction](#) ^[23]
- [Types](#) ^[25]
- [QoS](#) ^[26]
- [Application Target](#) ^[28]

3.3.1 Domain

The **Domain** toolbox provides elements and connectors for use in designing the Data Centric Publish Subscribe (DCPS) services for DDS system.



Element	Description
Package	A Package element which is used to contain and organize other DDS element.
Domain	A stereotyped <i>Part</i> element that represents a DDS DCPS Domain.
Domain Participant	A stereotyped <i>Component</i> element that represents a DDS DCPS Domain Participant. It applies the following Tagged Values: <ul style="list-style-type: none"> • <i>domain</i>: the DDS DCPS Domain that this DomainParticipant participates in.
Subscriber	A stereotyped <i>Part</i> element that represents a DDS DCPS Subscriber. It is modeled as a child of a DomainParticipant and can contain one or more <i>DataReader</i> elements as its children.
Data Reader	A stereotyped <i>Port</i> element that represents a DDS DCPS DataReader. It is modeled as a child of a Subscriber and can be connected to the DDS Topic that it reads.
Publisher	A stereotyped <i>Part</i> element that represents a DDS DCPS Publisher. It is modeled as a child of a DomainParticipant and can contain one or more DataWriter elements as its children.
Data Writer	A stereotyped <i>Port</i> element that represents a DDS DCPS DataWriter. It is modeled as a child of a Publisher and can be connected to the DDS Topic that it writes.
Qos Property	A stereotyped <i>Part</i> element that represents a DDS Qos Policy for a DDS DCPS Entity. It is modeled as a child element for the following DCPS Entities: <ul style="list-style-type: none"> • DomainParticipant • Subscriber

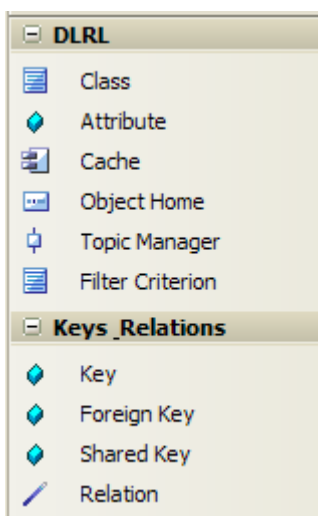
Element	Description
	<ul style="list-style-type: none"> • Publisher • DataReader • DataWriter • Topic <p>The name of the property specifies the Qos Policy defined by the property. It is also typed by a Qos policy specification, which defines the specific properties for that policy.</p> <p>Note:</p> <p>A Qos Property must be typed by a Qos Policy Specification in order to have the correct Tagged Values for the Qos Policy.</p> <p>For more information about Qos Policy Specifications refer to DDS Qos Policies^[26].</p>
Topic	<p>A stereotyped <i>Part</i> element that represents a DDS DCPS Topic. DDS Topics are connected to one or more DataReader and DataWriter elements that perform the role of reading or writing data to this topic. It applies the following Tagged Values:</p> <ul style="list-style-type: none"> • <i>expression</i>: a string expression used as the filter to be applied to this topic (CONTENT_FILTERED), or used as the subscription query to be applied to this topic (MULTI_TOPIC) • <i>kind</i>: an enumeration representing the Topic's kind - STANDARD, MULTI_TOPIC, CONTENT_FILTERED • <i>type</i>: a DDS type^[25] element used to describe the data structure for this topic. The type element is an IDL class.
Connector	A Connector to connect a DataReader/DataWriter with a Topic

Note:

Package and *Connector* are redefined from the standard Enterprise Architect toolset and are described in full in the *Enterprise Architect User Guide*.

3.3.2 Local Reconstruction

The **Domain** toolbox provides elements and connectors for use in designing the Data Centric Publish Subscribe (DCPS) services for DDS system.



Element	Description
Class	A stereotyped <i>Class</i> element that represents a DDS DLRL Class. It applies the following Tagged Values: <ul style="list-style-type: none"> • <i>mappedTopic</i>: the DDS Topic that this DLRL Class is mapped to.
Attribute	A stereotyped <i>Attribute</i> that represents an attribute within a DLRL Class. It applies the following Tagged Values: <ul style="list-style-type: none"> • <i>mappedField</i>: the Field of the DDS Topic Type that this attribute is mapped to for DLRL Read and Write operations.
Cache	A stereotyped <i>Component</i> element that represents a DDS DLRL Cache. It can contain one or more <i>ObjectHome</i> elements as its children and is associated with the DLRL Classes that it processes.
Object Home	A stereotyped <i>Part</i> element that represents a DDS DLRL ObjectHome. It is modeled as a child of a Cache element and can be related to a set of DCPS Topics.
Topic Manager	A stereotyped <i>Port</i> element that represents a DDS DLRL TopicManager. It is modeled as a part of ObjectHome element and can be connected to a specific DCPS Topic on behalf of its related ObjectHome.
Filter Criterion	A stereotyped <i>Class</i> element that represents a DDS DLRL FilterCriterion. It applies the following Tagged Values: <ul style="list-style-type: none"> • <i>filterClass</i>: the DLRL Class that implements the filterCriterion "check_object" operation.

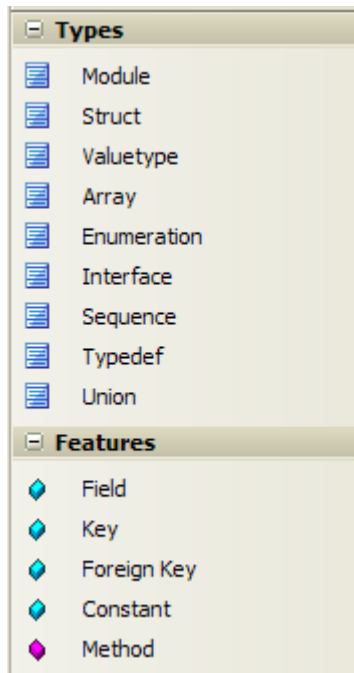
Relationship	Description
Key	A stereotyped <i>Attribute</i> Topic field that represents the identity of corresponding topic sample. Some topics may have more than 1 key fields to determine their identity.
Foreign Key	A stereotyped <i>Attribute</i> Topic field that represents a reference to a key field of another Topic. It is used to form relationships between topics.
Shared Key	A stereotyped <i>Attribute</i> Topic field which is both Key and Foreign Key.
Relation	A stereotyped <i>Association</i> that represents a relationship between two DLRL Classes.

Note:

Package and *Connector* are redefined from the standard Enterprise Architect toolset and are described in full in the *Enterprise Architect User Guide*.

3.3.3 Types

The **DDS Types** Toolbox provides elements and features for use on a DDS Types diagram.

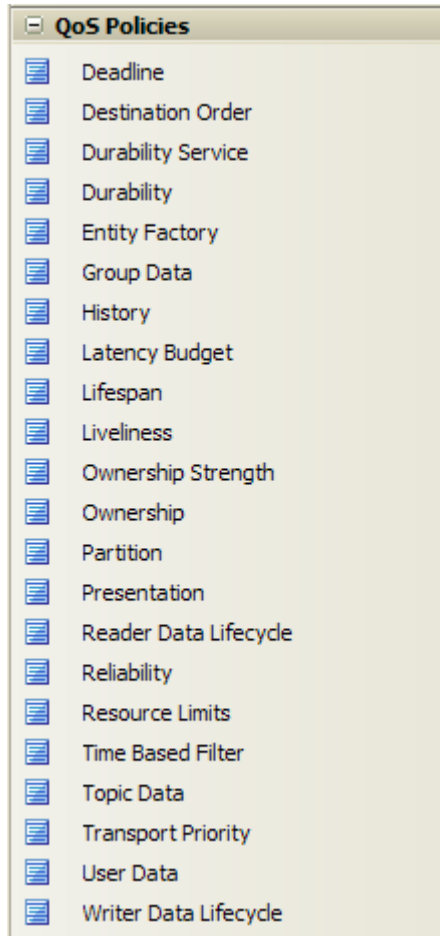


Elements	Description
Module	A stereotyped <i>Class</i> element that represents an IDL Module.
Struct	A stereotyped <i>Class</i> element that represents an IDL Struct.
Valuetype	A stereotyped <i>Class</i> element that represents an IDL Valuetype.
Array	A stereotyped <i>Class</i> element that represents an IDL Array.
Enumeration	A stereotyped <i>Class</i> element that represents an IDL Enumeration.
Interface	A stereotyped <i>Class</i> element that represents an IDL Interface.
Sequence	A stereotyped <i>Class</i> element that represents an IDL Sequence.
Typedef	A stereotyped <i>Class</i> element that represents an IDL TypeDef.
Union	A stereotyped <i>Class</i> element that represents an IDL Union.

Features	Description
Field	A stereotyped <i>Attribute</i> that represents an IDL field.
Key	A stereotyped <i>Attribute</i> that represents an IDL field which is the (or part of) the identify.
Foreign Key	A stereotyped <i>Attribute</i> that represents an IDL field which is the Key field of another IDL element.
Constant	A stereotyped <i>Attribute</i> that represents an IDL Constant.
Method	A stereotyped <i>Operation</i> that represents an IDL Method.

3.3.4 QoS

The **DDS Qos Policies** toolbox provides elements for use on a DDS Qos Policies Library diagram.



Element	Description
Deadline	A stereotyped <i>Class</i> element that specifies the Deadline Qos Policy. It applies the following Tagged Values: <ul style="list-style-type: none"> <i>period</i>: duration in nanoseconds (default -1).
Destination Order	A stereotyped <i>Class</i> element that specifies the Destination Order Qos Policy. It applies the following Tagged Values: <ul style="list-style-type: none"> <i>kind</i>: Enumeration: BY_RECEPTION_TIMESTAMP (default), BY_SOURCE_TIMESTAMP
Durability Service	A stereotyped <i>Class</i> element that specifies the Durability Service Qos Policy. It applies the following Tagged Values: <ul style="list-style-type: none"> <i>history_depth</i>: Integer depth value (default 1) <i>history_kind</i>: Enumeration: KEEP_LAST (default), KEEP_ALL <i>max_instances</i>: Integer maximum value (default -1 to keep all instances) <i>max_samples</i>: Integer maximum samples (default -1 to keep all samples) <i>max_samples_per_instance</i>: Integer maximum samples per instance (default -1 to keep all samples per instance) <i>service_cleanup_delay</i>: duration in nanoseconds (default 0).
Durability	A stereotyped <i>Class</i> element that specifies the Durability Qos Policy. It applies the following

Element	Description
	Tagged Values: <ul style="list-style-type: none"> • <i>kind</i>: Enumeration: TRANSIENT, TRANSIENT_LOCAL, VOLATILE (default), PERSISTENT.
Entity Factory	A stereotyped <i>Class</i> element that specifies the EntityFactory Qos Policy. It applies the following Tagged Values: <ul style="list-style-type: none"> • <i>autoenable_created_entities</i>: Boolean (default true).
Group Data	A stereotyped <i>Class</i> element that specifies the Groupdata Qos Policy. It applies the following Tagged Values: <ul style="list-style-type: none"> • <i>datavalue</i>: String.
History	A stereotyped <i>Class</i> element that specifies the History Qos Policy. It applies the following Tagged Values: <ul style="list-style-type: none"> • <i>depth</i>: Integer (default 1) • <i>kind</i>: Enumeration: KEEP_LAST (default), KEEP_ALL.
Latency Budget	A stereotyped <i>Class</i> element that specifies the LatencyBudget Qos Policy. It applies the following Tagged Values: <ul style="list-style-type: none"> • <i>duration</i>: duration in nanoseconds (default 0).
Lifespan	A stereotyped <i>Class</i> element that specifies the Liveliness Qos Policy. It applies the following Tagged Values: <ul style="list-style-type: none"> • <i>duration</i>: duration in nanoseconds (default 0).
Liveliness	A stereotyped <i>Class</i> element that specifies the Deadline Qos Policy. It applies the following Tagged Values: <ul style="list-style-type: none"> • <i>kind</i>: Enumeration: AUTOMATIC (default), MANUAL_BY_PARTICIPANT, MANUAL_BY_TOPIC • <i>lease_duration</i>: duration in nanoseconds (default -1).
Ownership Strength	A stereotyped <i>Class</i> element that specifies the Ownership Strength Qos Policy. It applies the following Tagged Values: <ul style="list-style-type: none"> • <i>value</i>: Integer (default 0).
Ownership	A stereotyped <i>Class</i> element that specifies the Ownership Qos Policy. It applies the following Tagged Values: <ul style="list-style-type: none"> • <i>kind</i>: Enumeration: SHARED (default), EXCLUSIVE.
Partition	A stereotyped <i>Class</i> element that specifies the Partition Qos Policy. It applies the following Tagged Values: <ul style="list-style-type: none"> • <i>name</i>: String.
Presentation	A stereotyped <i>Class</i> element that specifies the Presentation Qos Policy. It applies the following Tagged Values: <ul style="list-style-type: none"> • <i>access_scope</i>: Enumeration: INSTANCE (default), TOPIC, GROUP • <i>coherent_access</i>: Boolean (default false) • <i>ordered_access</i>: Boolean (default false).
Reader Data Lifecycle	A stereotyped <i>Class</i> element that specifies the ReaderData Lifecycle Qos Policy. It applies the following Tagged Values: <ul style="list-style-type: none"> • <i>autopurge_nowriter_samples_delay</i>: duration in nanoseconds (default -1).
Reliability	A stereotyped <i>Class</i> element that specifies the Reliability Qos Policy. It applies the following Tagged Values: <ul style="list-style-type: none"> • <i>kind</i>: Enumeration: BEST_EFFORT (default), RELIABLE. • <i>max_blocking_time</i>: Integer (default -1).

Element	Description
Resource Limits	A stereotyped <i>Class</i> element that specifies the Resource Limits Qos Policy. It applies the following Tagged Values: <ul style="list-style-type: none"> • <i>max_instances</i>: Integer maximum value (default -1 to keep all instances) • <i>max_samples</i>: Integer maximum samples (default -1 to keep all samples) • <i>max_samples_per_instance</i>: Integer maximum samples per instance (default -1 to keep all samples per instance).
Time Based Filter	A stereotyped <i>Class</i> element that specifies the Time Based Filter Qos Policy. It applies the following Tagged Values: <ul style="list-style-type: none"> • <i>minimum_separation</i>: duration in nanoseconds (default 0).
Topic Data	A stereotyped <i>Class</i> element that specifies the Topic Data Qos Policy. It applies the following Tagged Values: <ul style="list-style-type: none"> • <i>value</i>: String.
Transport Priority	A stereotyped <i>Class</i> element that specifies the Transport Priority Qos Policy. It applies the following Tagged Values: <ul style="list-style-type: none"> • <i>value</i>: Integer (default 0).
User Data	A stereotyped <i>Class</i> element that specifies the User Data Qos Policy. It applies the following Tagged Values: <ul style="list-style-type: none"> • <i>value</i>: String.
WriterData Lifecycle	A stereotyped <i>Class</i> element that specifies the WriterData Lifecycle Qos Policy. It applies the following Tagged Values: <ul style="list-style-type: none"> • <i>autodispose_unregistered_instances</i>: Boolean (default true).

3.3.5 Application Target

The **DDS Application** toolbox provides elements and connectors for use on a DDS Application diagram.



Element	Description
DDS Application	A stereotyped <i>Component</i> element that represents a DDS application. It is used to represent a compilable software artifact that represents a single DomainParticipant (using the Use relationship). Application Component is the DDS application used by the transformation process to specify the DDS implementation PSM generated by the transformation. It applies the following Tagged Values: <ul style="list-style-type: none"> • <i>language</i>: An enumeration representing the target software language for this application implementation - C, C++, Java and C#. • <i>platform</i>: An enumeration representing the target implementation platform for this application implementation - i86Win32, i86JDK13.

Notes:

- Currently only the RTIDDS Transformation supports **C#**.

Connector	Description
Domain Participant / Cache Use	A stereotyped <i>Dependency</i> relationship that represents a usage of a DDS DomainParticipant or Cache by an Application. Many Applications can use the same DomainParticipant/Cache as required to create a heterogeneous DDS design.
Association	An association. It is required for a DLRL Cache to associate with a related DCPS DomainParticipant.

3.4 Create DDS Project

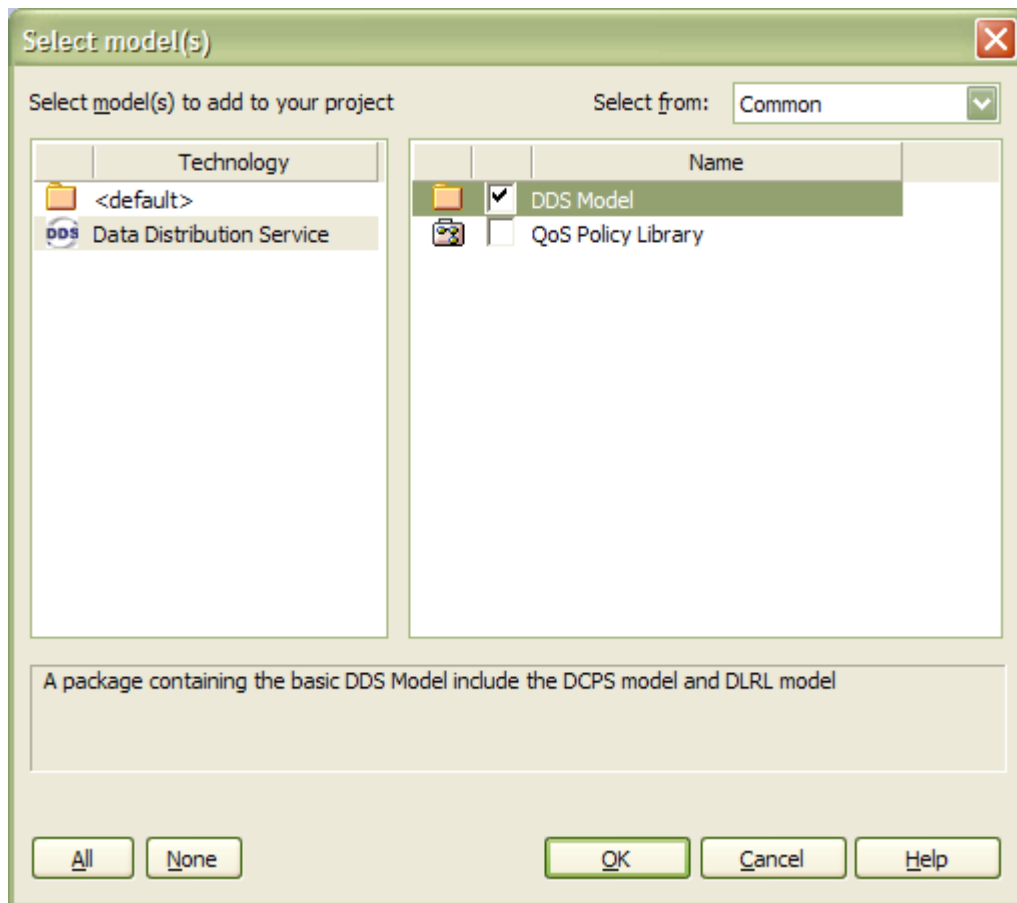
There are two ways in which you can create a DDS Model:

- [Using the predefined Model Template](#)^[30]
- [Without a Model Template](#)^[30]

3.4.1 Using a Template

To create a new DDS model using a template, follow the steps below.

1. Select the **File | New Project** menu option. In the **Name** field, type an appropriate name for the DDS project. Click on the **Save** button.
2. The **Model Wizard** dialog then displays as shown below. Select the **Data Distribution Service** technology, then tick the checkbox for **DDS Model**.
3. Click OK to load the template.



The new DDS Model is created from the selected template. It contains a blank DCPS and DLRL model for you to start designing your application with.

3.4.2 Without a Template

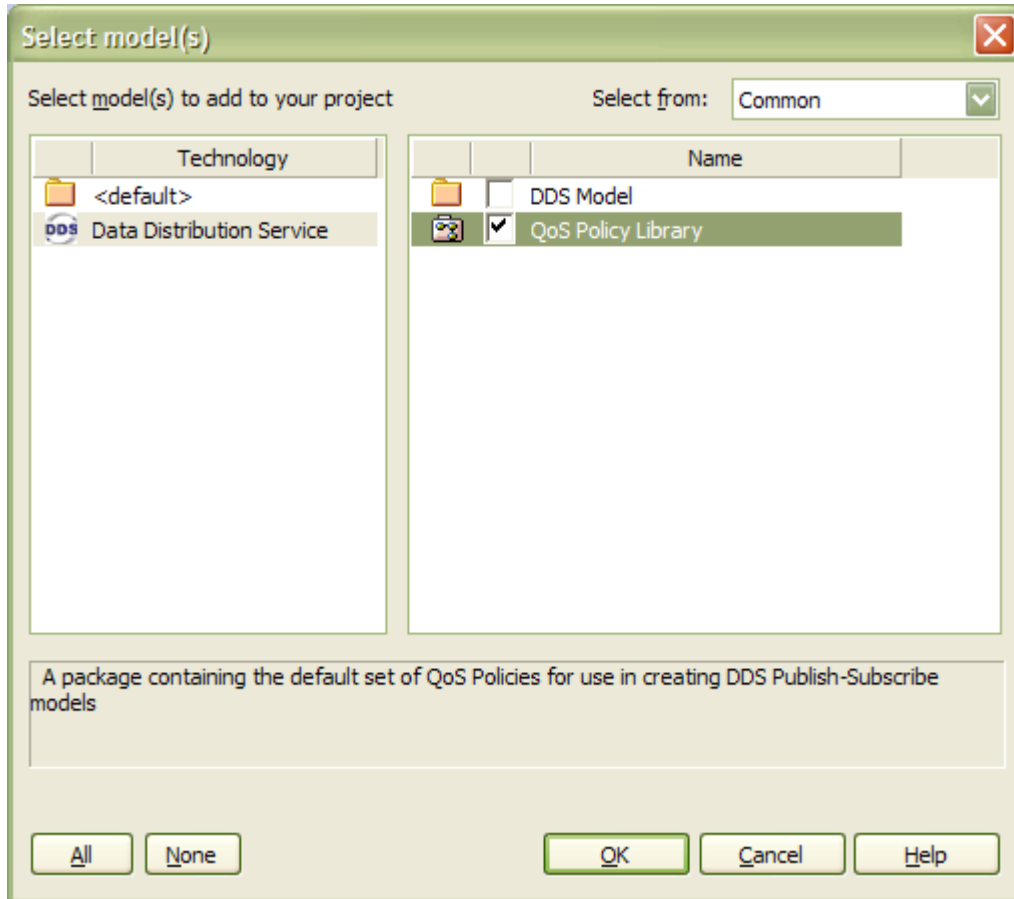
The following topics describe methods to create a DDS model without reference to a template:

- [Create DCPS Model](#)^[31]
- [Create DLRL Model](#)^[34]

3.4.2.1 Create DCPS Model

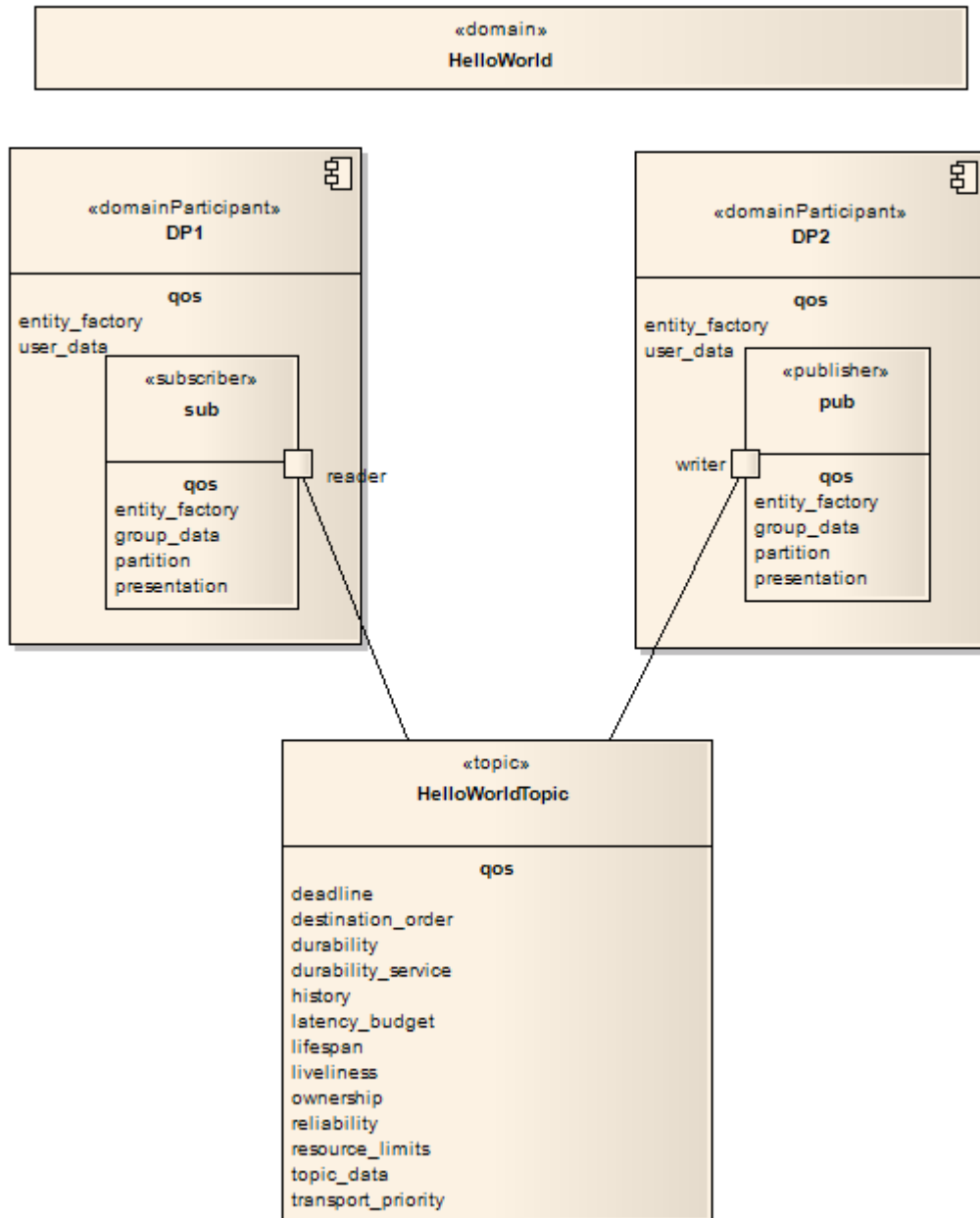
To create a new DCPS model follow the steps below.

1. Select the **File | New Project** menu option. In the **Name** field, type an appropriate name for the DDS project. Click on the **Save** button.
2. The **Model Wizard** dialog then displays as shown below. Select the **QoS Policy Library**.
3. Then OK to load the template.

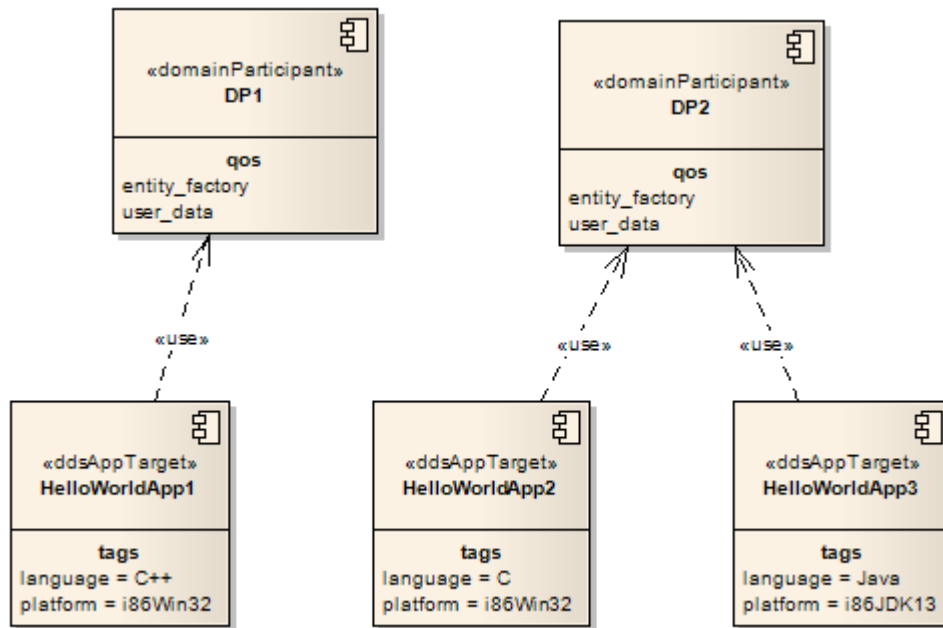


4. In the **Project Browser**, add two packages to the project. Give these packages names appropriate to the project and purpose:
 - A Platform Independent Model (PIM) package
 - An Platform Specific Model (PSM) package.
5. Add a new child Package in the PIM package. Give this package appropriate name.
6. Add a new *DDS Type Diagram* to the package: Select the **Add-Ins | DDS Technology | New Type Diagram** menu option to create a *DDS Type diagram*. In the diagram, create the IDL type elements and add key and other field to the elements.
7. Add a new *DDS Domain diagram* to the package: Select the **Add-Ins | DDS Technology | New Domain Diagram** menu option to create a *DDS Domain diagram*. In the diagram, create the *DDS Topic elements* and assign the Tagged Value Type to the elements. Create a *domain* element, then create the *DomainParticipant elements* and assign the Tagged Value of *Domain* to the elements.
 - (i) Create *DDS Topic* elements and assign the Tagged Value Type to the elements.
 - (ii) Create a *domain* element, then create *DomainParticipant elements* and assign the Tagged Value of *Domain* to the elements.
 - (iii) Within the *DomainParticipant* elements, create *Publisher/Subscriber* elements.
 - (iv) Create *DataWriters* for each Publisher and *DataReaders* for each Subscriber.
 - (v) Create connectors between the Topics and the DataWriter and DataReader elements.

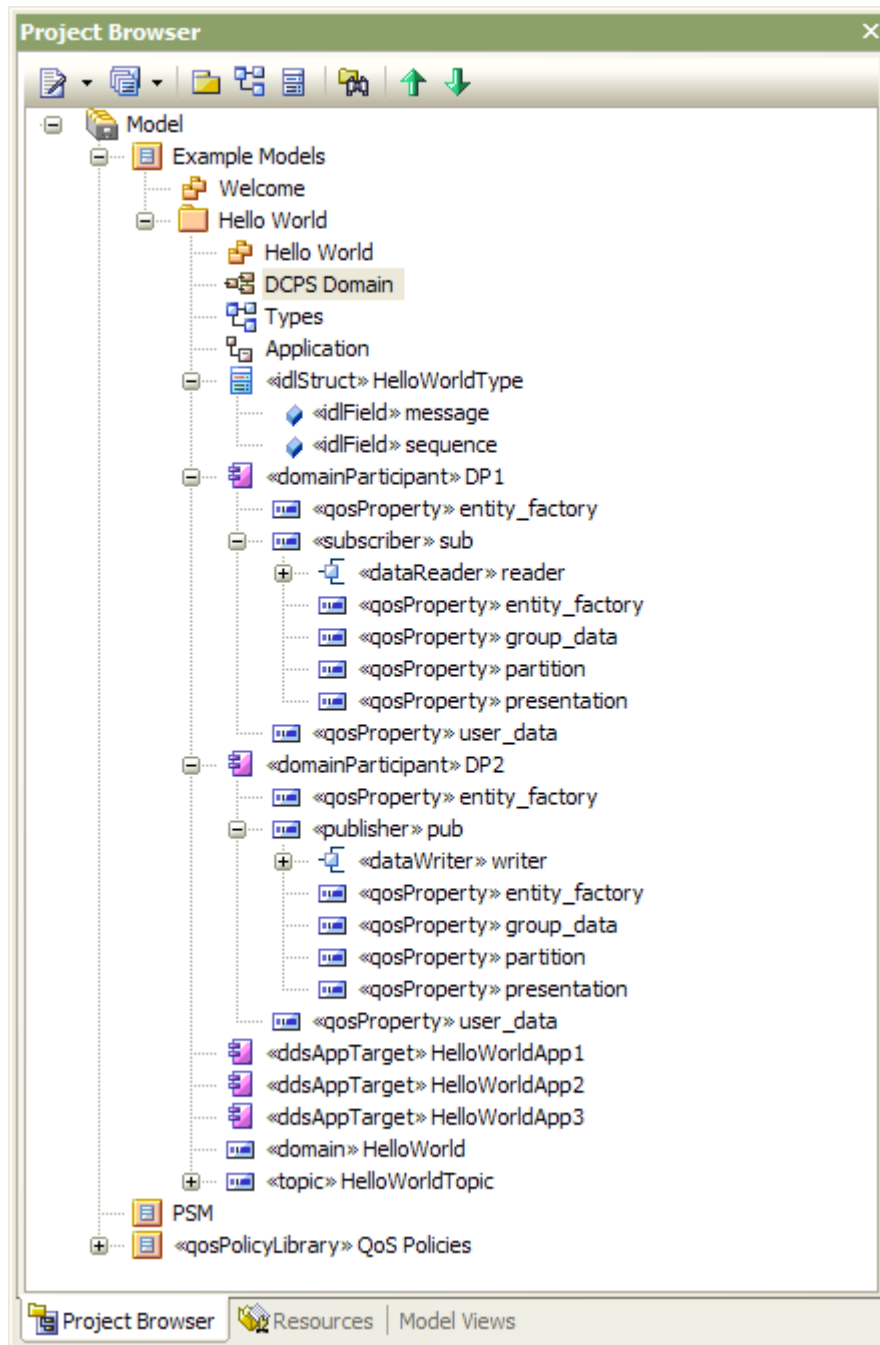
Refer to the example diagram below:



8. Add a new *DDS Application Diagram* to the package: Select the **Add-Ins | DDS Technology | New Application Diagram** menu option to create a *DDS Application diagram*. Drag the *DomainParticipant* elements from the *DDS Domain* package in the **Project Browser** onto the diagram. Create *Application Component* elements and then create *Usage* connectors between these *Application Component* elements and the *DomainParticipant* elements. See example diagram below:



When you have saved these diagrams, you have created a basic DDS Platform Independent Model (PIM) having similar structure in the [Project Browser](#) as following diagram. If you want to add a DLRL Model, see [Add DLRL Model](#)^[34] topic.



You can run [Model Validation](#)^[38] against the newly-created PIM to validate the model. You can then [Generate PSM and Code](#)^[40].

3.4.2.2 Create DLRL Model

Note:

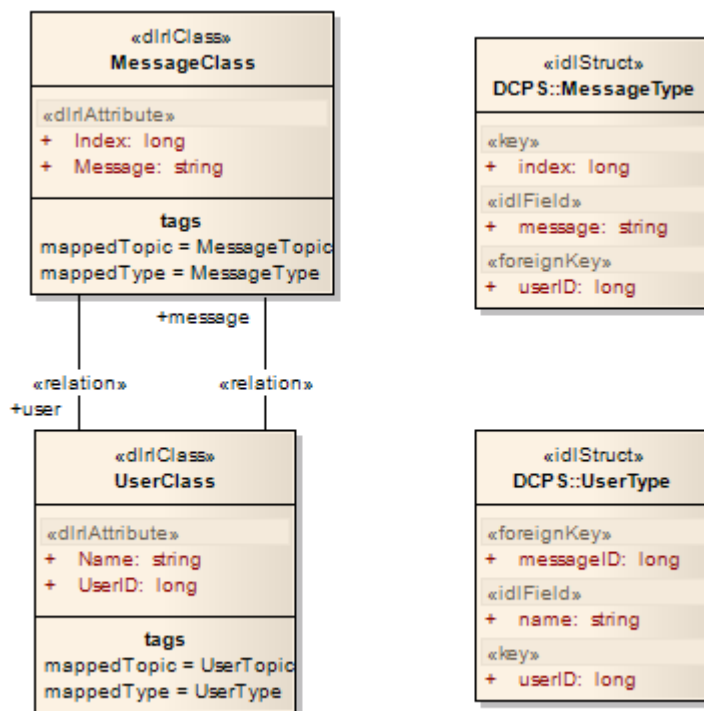
- Currently only OSPLDDS Supports generating Platform Specific Model (PSM) for **Java** source code from DLRL models.
- You need to firstly create a DDS DCPS model before you can create a valid DLRL model.

To add a DDS DLRL model, follow the steps below:

1. In the **Project Browser**, create a package for the DLRL model.

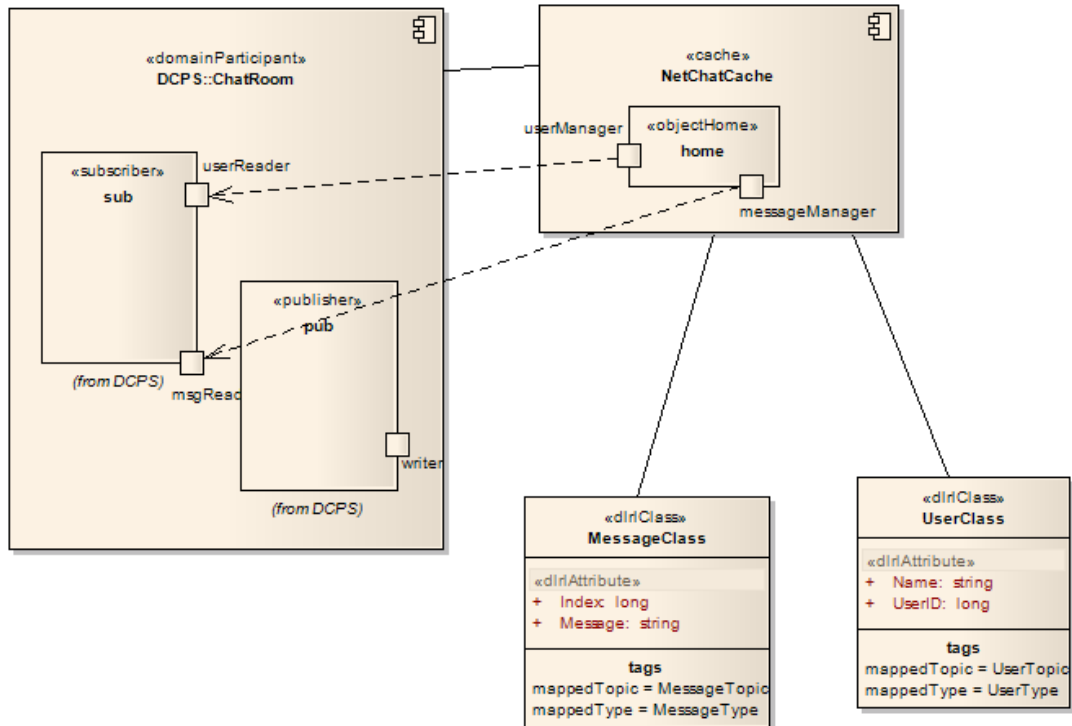
2. Add a new *DLRL Mapping Diagram* in the package. Select the **Add-Ins | DDS Technology | New DLRL Mapping Diagram** menu option to create a *DDS DLRL Mapping diagram*.
3. In the newly created DLRL Diagram:
 - (i) Create DLRL Classes, and assign the mappedTopic and mappedType tagged values to the appropriate elements.
 - (ii) Add some DLRL Attributes to the DLRL Class elements. Assign the mappedField Tagged Value to the attribute in the mapping IDL type.
 - (iii) Create *Relation* Relationships between the DLRL Classes. Assign the mapping Foreign Key name to the *Target Role* property of the Relation connector.

The diagram below is an example of DLRL class mapping:



4. Add a new *DDS DLRL Diagram*: Select the **Add-Ins | DDS Technology | New DLRL Diagram** menu option to create a *DDS DLRL diagram*. Give this diagram appropriate name. Then:
 - (i) Drag the DomainParticipant (and its children Subscriber/Publisher as well as Reader/Writer elements) in the DCPS package from **Project Browser** to the diagram. And drag the DLRL Class element in the current package to the diagram.
 - (ii) Add some *Cache* elements. Create *Associations* between the *Cache* element and the DCPS DomainParticipant element. Create *Associations* between *Cache* element and the DLRL Class elements.
 - (iii) Add a *Object Home* Part inside the *Cache* element. Create *Topic Manager* ports to the *Object Home* part. Then create *Dependency* connectors from each *Topic Manager* port to related *DataReader/DataWriter* port within DCPS elements

The diagram below is an example of the above process:

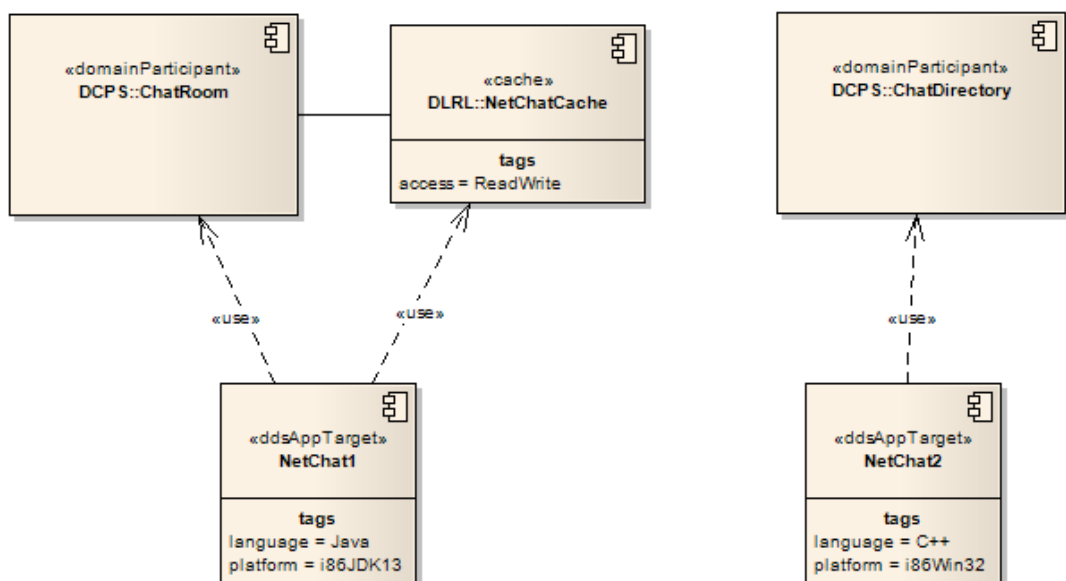


- Open an existing (or create a new) *DDS Application diagram*, then drag the DLRL Cache element from **Project Browser** to the diagram. You can either create a *Usage* relationship from an existing *DDS Application* component to the *Cache* element, or create a new *DDS Application* component and then create a *Usage* relationship between the new created *DDS Application* element and *Cache* element.

Note:

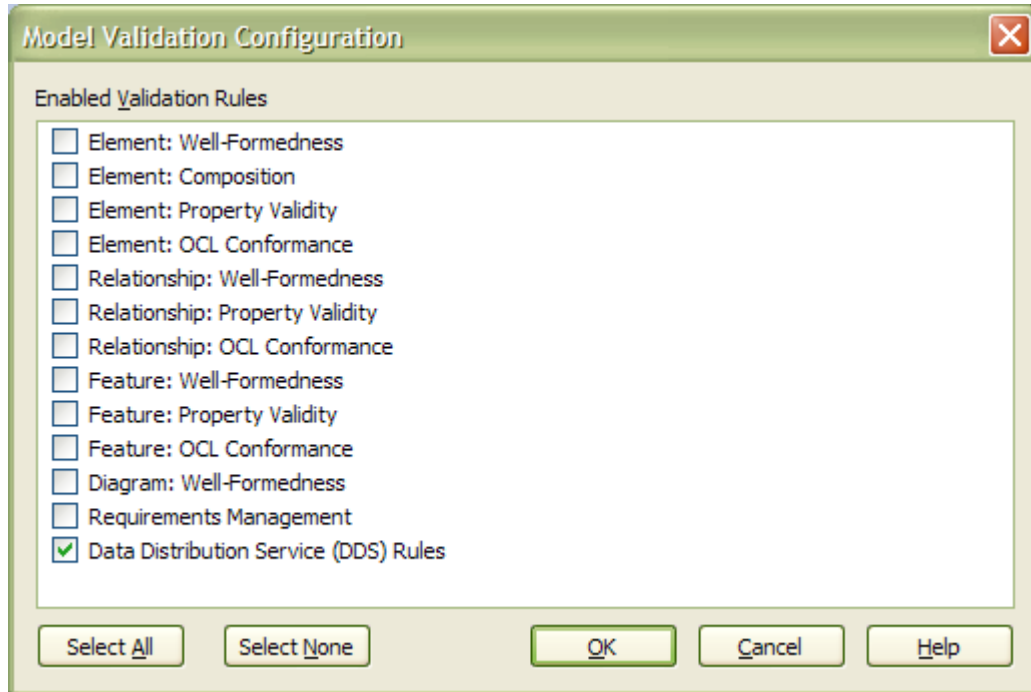
- The *DDS Application* component must also have a *Usage* relationship with related *DomainParticipant* components, otherwise the generated code will not be compilable.

The diagram below is an example *DDS Application* diagram including the DLRL Cache:



3.5 Model Validation

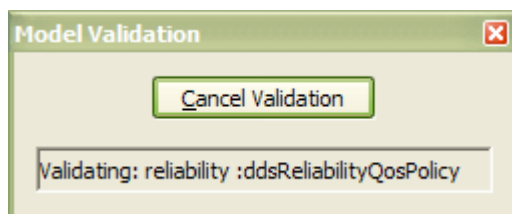
The MDG Technology for DDS registers with Enterprise Architect to receive model validation requests from users. To configure Enterprise Architect to perform DDS model validation, select the **Project | Model Validation | Configure** menu option from the Enterprise Architect main menu. The **Model Validation Configuration** dialog displays.



If you are only performing DDS validation, click on the **Select None** button, select the **Data Distribution Service (DDS) Rules** checkbox and click on the **OK** button. This is the default setting of the DDS Perspective. Note that DDS is strongly based on UML so it might be a useful exercise to perform the UML validation as well.

Validate a DDS Model

To validate an element and any connectors attached to it, a diagram and all its elements, or a package and all its diagrams and elements against the DDS rules, select the **Project | Model Validation | Validate Selected** menu option.



For an explanation of the error and warning messages that might be returned when validating a DDS model, refer to the [Model Validation Rules for Elements](#) ³⁸ topic.

3.5.1 Model Validation Rules for Elements

The following messages might be output by the validation of a DDS element:

Message	Meaning
DDS Entity missing data for Tagged Value:	A DDS <i>DomainParticipant</i> , <i>Topic</i> , <i>DLRL Class</i> or <i>Application</i> is missing the Tagged Value specified by the message.

Message	Meaning
<domain type language platform>	
DDS Entity contains invalid data for Tagged Value: <domain type topic>	The DDS <i>DomainParticipant</i> , <i>Topic</i> or <i>DLRL Class</i> refers to an invalid element in the specified Tagged Value.
Missing QosPolicy: <QOS property> PSM Transform will use system default values for this policy	The DDS <i>DomainParticipant</i> , <i>Topic</i> , <i>Subscriber</i> , <i>Publisher</i> , <i>DataReader</i> or <i>DataWriter</i> is missing the specified Qos Policy (specified by a <i>QosProperty</i> element). The PSM transformation process assumes the system default value for this Qos Policy.
Missing child entity: <Publisher or Subscriber DataReader DataWriter ObjectHome TopicManager>	The DDS <i>DomainParticipant</i> , <i>Publisher</i> , <i>Subscriber</i> , <i>DLRL Cache</i> or <i>ObjectHome</i> are missing any of their required children entities.
Invalid children entities found of type: <Any DDS Entity>	The given DDS entity contains other DDS entities as children, which it is not supposed to have.
Missing or invalid usage relationship: <other information>	The given DDS application is not bound to any <i>DomainParticipant/Cache</i> (or vice versa).
Missing or invalid association: <other information>	The given DDS topic is not associated with any <i>DataReaders</i> or <i>DataWriters</i> , or a <i>DataReader/DataWriter</i> is not associated with any topics.
Duplicated DDS Entity Name: <Any DDS Entity name>	The given DDS entity has the same name as another DDS entity.
Unsupported type found in <DDS Type> - <attribute> : <attribute type>	The type of attribute in the given <i>DDS Type</i> entity is neither an IDL primitive type, nor a user-defined DDS type.
Missing DDS parent: <Any DDS Entity type>	The given DDS entity does not belong to the specified parent type.
Invalid DDS parent: <Any DDS Entity type>	The given DDS entity belongs to the wrong parent type.
Incompatible Platform/Language - Platform <platform> does not comply with Language <language>	The language and platform value in a specified DDS Application entity tagged value is not compatible.

Note:

A DDS model might trigger additional UML validation messages, but these are outside the scope of this document. See the *Enterprise Architect User Guide*.

3.6 Generate PSM and Code

There are two ways in which you can generate PSM and Code:

- [Generate Code DDS Menu Command](#) ^[40]
- [EA Transformation](#) ^[41]

Note:

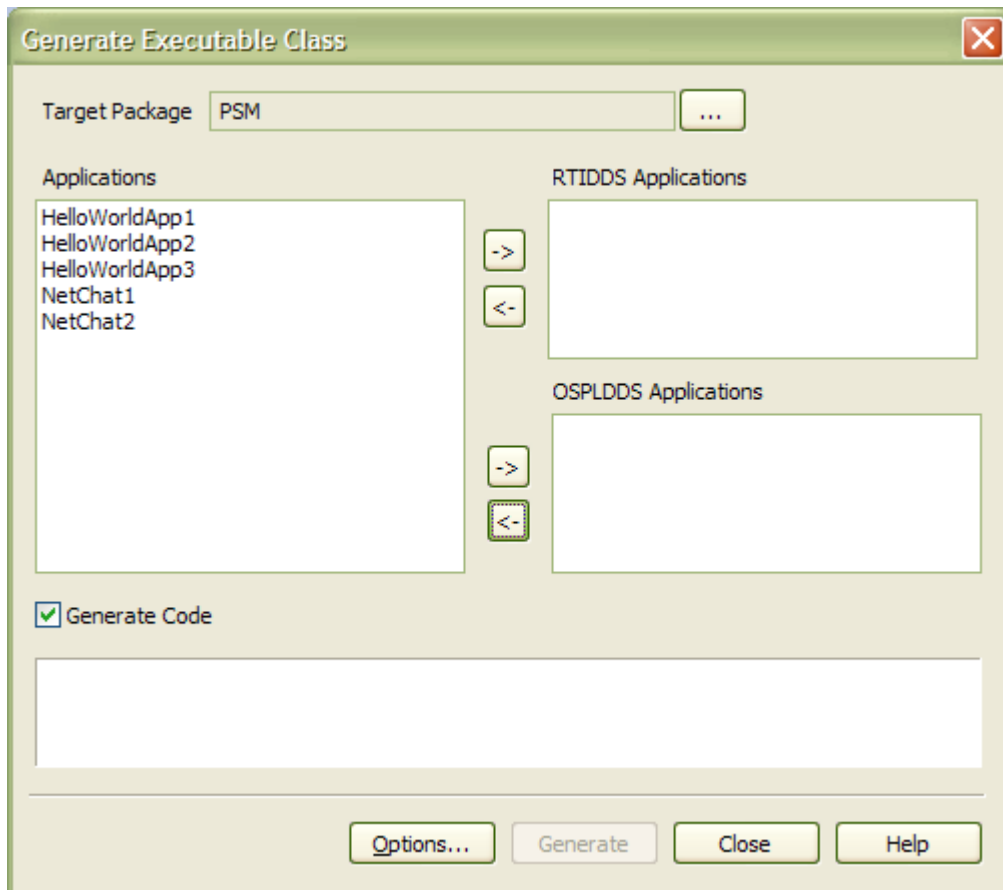
Before generating the PSM and Code, please ensure that the DDS Model is validated by running [Model Validation](#) ^[38].

3.6.1 DDS Menu Command

This process generates executable UML class and packages based on the DDS model. The generated class and packages can then be generated to source code using in-built code generator feature of Enterprise Architect.

To generate the Executable Class for a DDS model, follow the steps below:

1. Select the **Add-Ins | DDS Technology | Generate Code...** menu option to display the **Generate Executable Class** dialog:



2. Select the target package that the generated class and package will be created in. Click the Browse button to select the desired package.
3. From the *Applications* list box, select the applications you want them to be generated to RTIDDS applications, then click the ">" button beside *RTIDDS Applications* list box. The selected applications will be moved to *RTIDDS Applications* list box.
4. From the *Applications* list box, select the applications you want them to be generate to OSPLDDS applications, then click the ">" button beside *OSPLDDS Applications* list box. The selected applications will be moved to *OSPLDDS Application* list box.
5. Tick the *Generate Code* checkbox if you decide to have source code generated after the class and

package has been generated.

6. Click on the *Generate* button to initiate the process.

3.6.2 EA Transformation

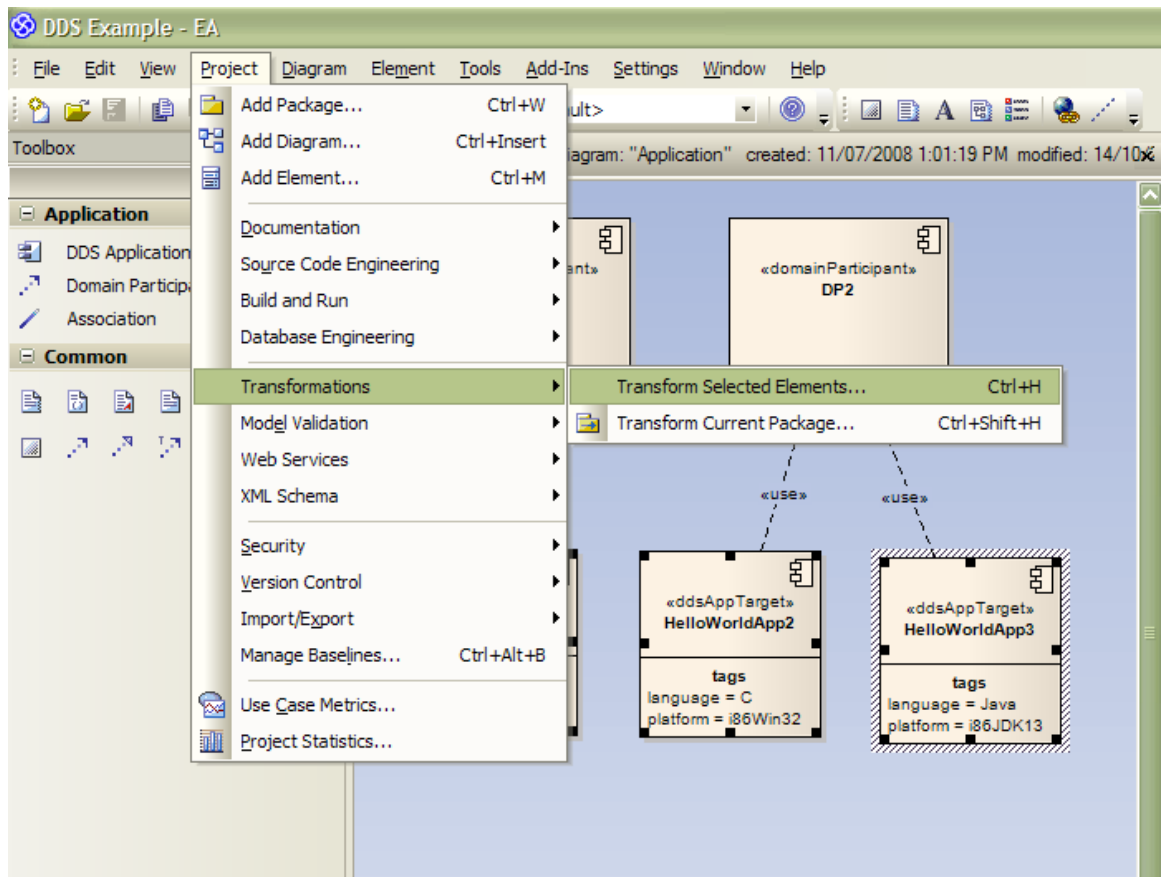
There are two Tagged Values for an Application Component - *Language* and *Platform*.

- The value of the *Language* Tagged Value defines the coding language to generate: For RTI NDDS, there are three possible values - **C**, **C++** and **Java**, for OSPLDDS, there are four possible values - **C**, **C++**, **Java** and **C#**.
- The value of the *Platform* Tagged Value depends on the language defined:
 - For C, C++ and C#, the value must be **i86Win32**.
 - For Java, the value must be **i86JDK13**.

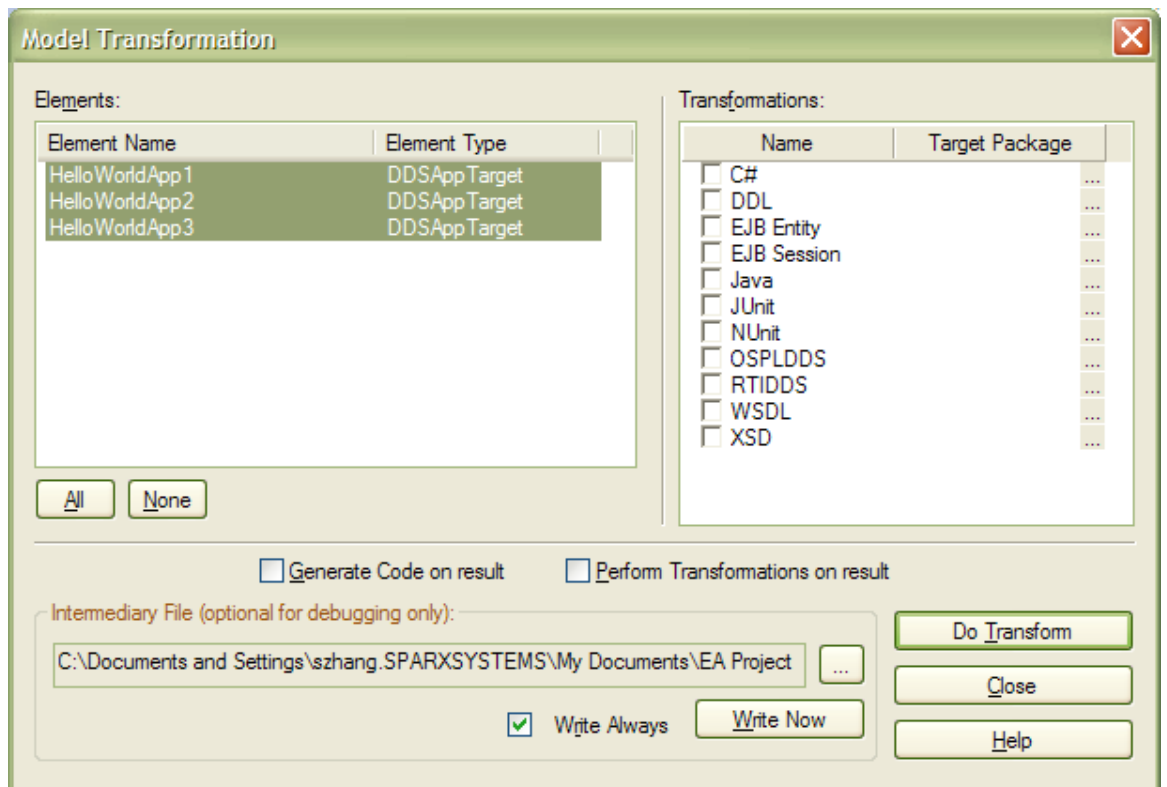
Before you generate the PSM and code, you might want to check and, if necessary, change the language and platform.

To generate the PSM and code, follow the steps below:

1. Open the DDS project and the PIM package, and then open the Application diagram.
2. In the Application diagram, select the Application Components from which to generate the PSM and code.
3. Select the **Project | Model Transformations | Transform Selected Elements** menu option.

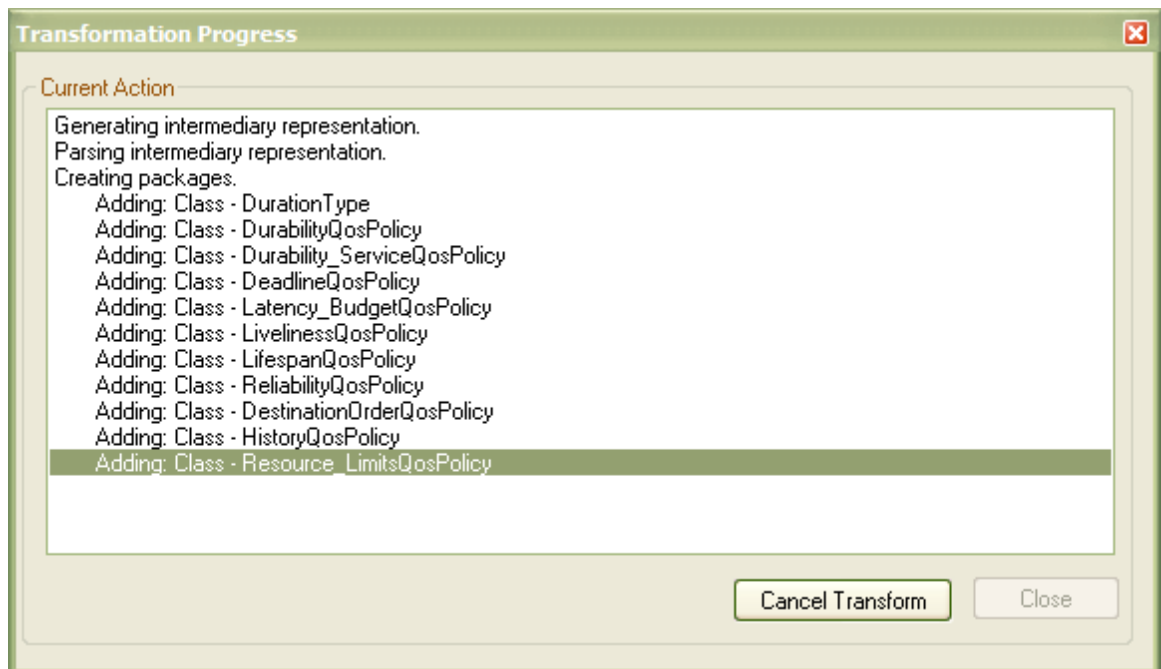


The **Model Transformation** dialog displays.

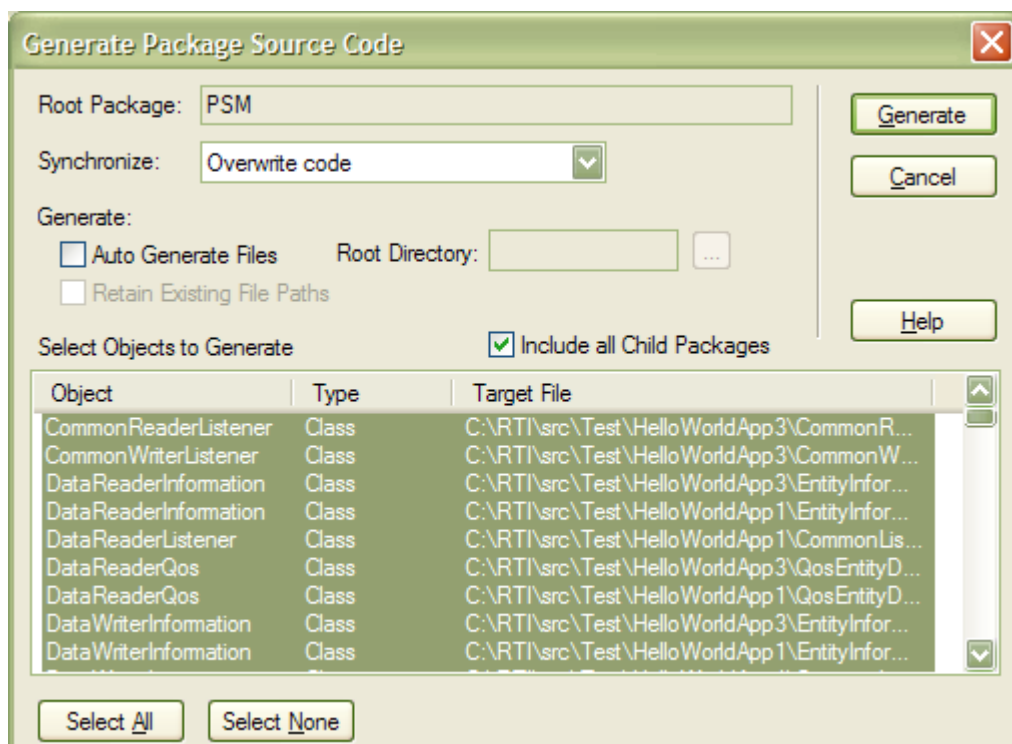


4. Select the **OSPLDDS** or **RTIDDS** checkbox. If the target package is not listed against the checkbox, the **Browse Project** dialog displays. Select the PSM package as the target for the generated PSM and click on the **OK** button.

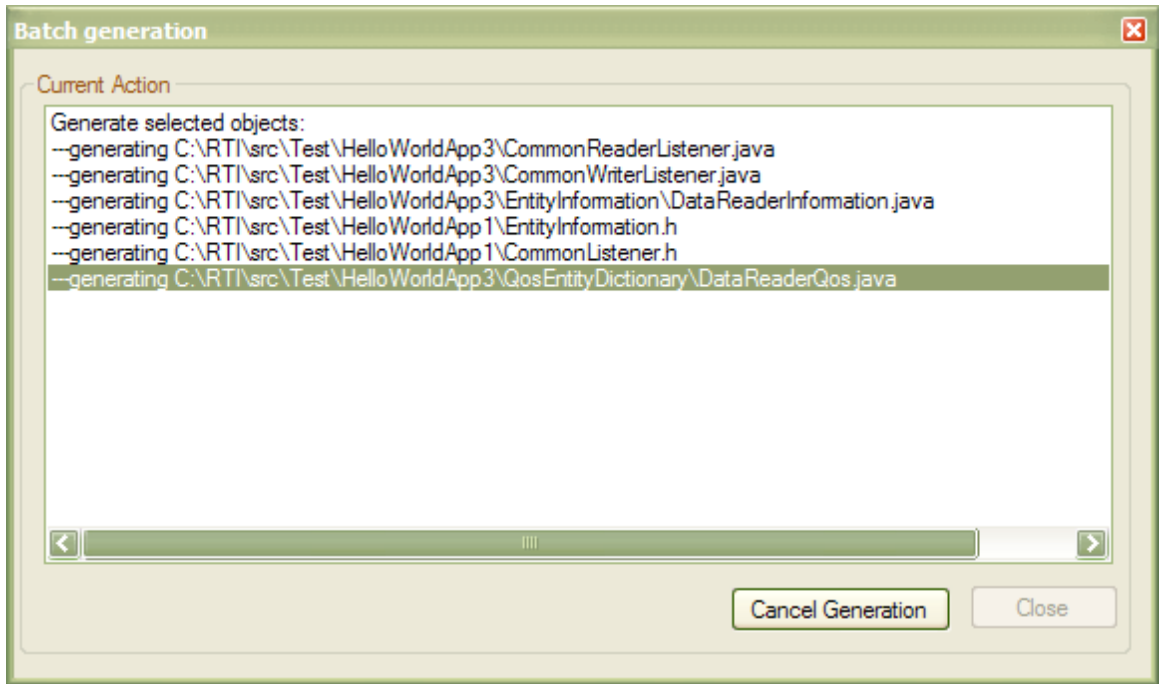
- On the **Model Transformation** dialog, click on the **Do Transform** button. (If you have not specified a project root folder in the **DDS Options** ^[17], the system prompts you for the folder at this point.) The system starts to generate the PSM based on the selected DDS components; the progress of the transformation is shown in the **Transformation Progress** window.



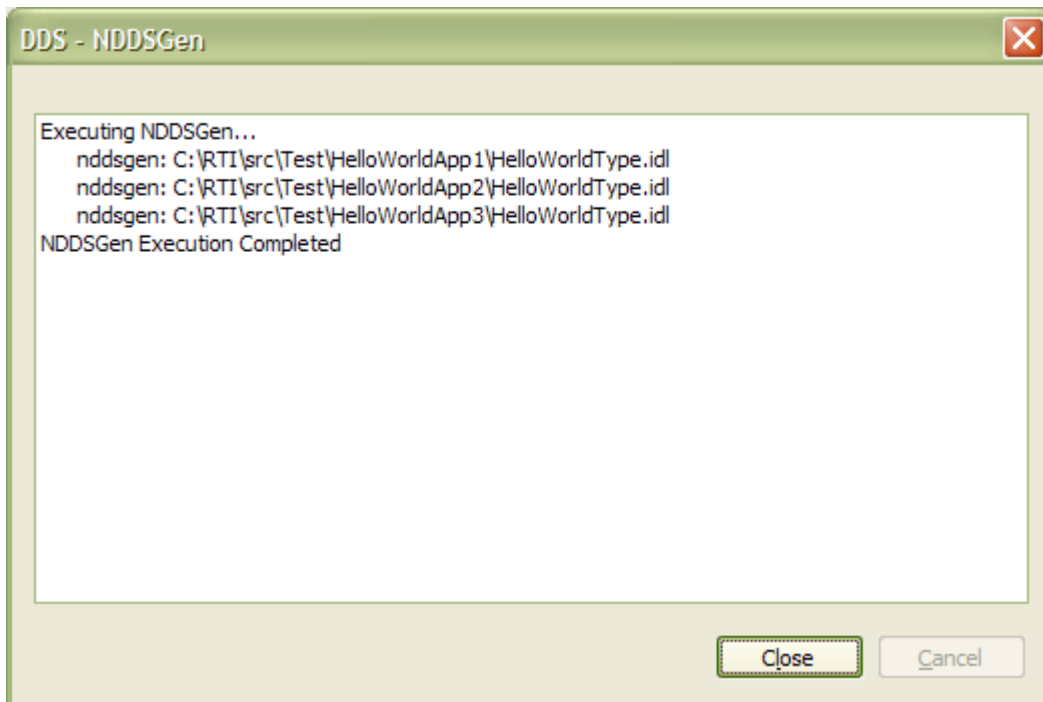
- When the transformation is complete, the **Generate Package Source Code** dialog displays.



- In the **Synchronize** field, select **Overwrite code**, then select the **Include all Child Packages** checkbox. Click on the **Generate** button. The system starts to generate the source code for the specified language into the project root folder. The progress of the code generation is shown in the **Batch Generation** window.



8. If user select RTIDDS Transformation in step 4, then after code generation is completed, the system runs the NDDSGen to generate code from the IDL type.

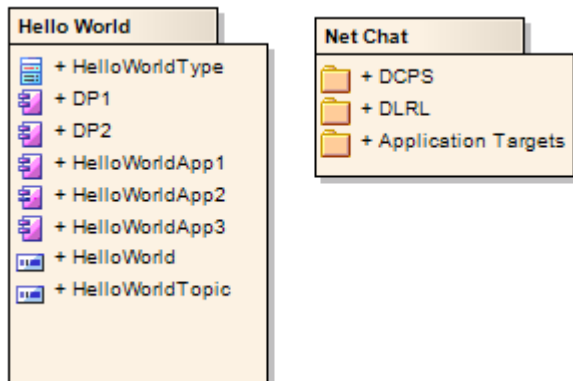


9. Finished.

3.7 DDS Example Model

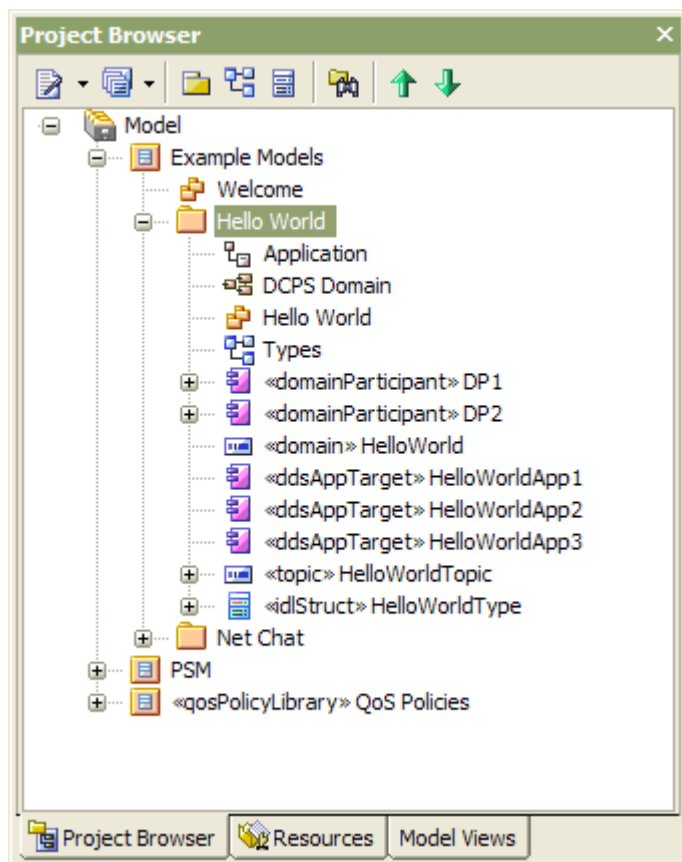
To open the example model, select the **Add-Ins | DDS Technology | Open DDS Example Model** menu option from the Enterprise Architect main menu.

The example model contains two separate model: **Hello World** and **Net Chat** as show in following diagram:



The **Hello World** model is a simple DCPS model, it contains a Type diagram, Domain diagram and DDS Application diagram.

In Type diagram it defines the IDL type to be used by the Topic. In Domain diagram it defines DDS topology using DDS Entities: Domain, DomainParticipant, Publisher and Subscriber, DataReader and DataWriter, and Topic. In Application diagram it defines the DDS Application entities and its usage relation with DomainParticipant entities.

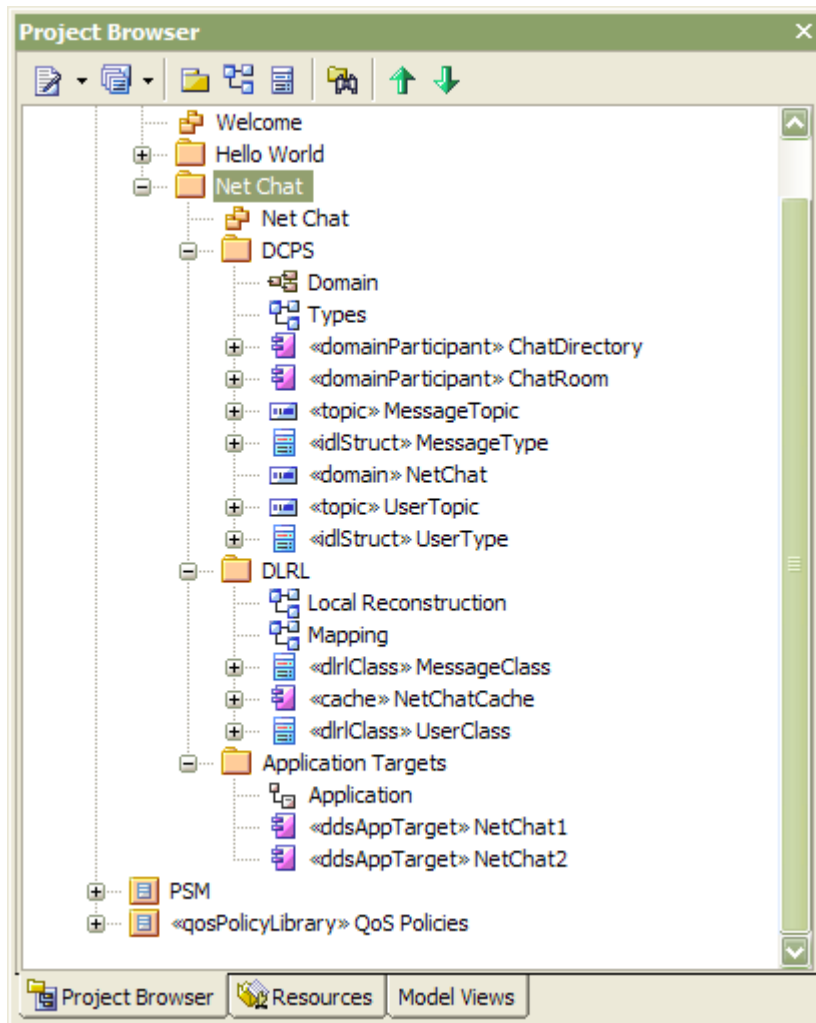


The **Net Chat** model contains DCPS Model as well as DLRL model which based on its related DCPS model.

Its DCPS model contains a Type diagram and a Domain diagram.

Its DLRL model contains a Mapping diagram and a DLRL diagram. In Mapping diagram it defines DLRL

classes as well as their mapping to related DCPS Type and Topic. In DLRL diagram it defines the DLRL topology using DDS DLRL Entities: Cache, ObjectHome, TopicManager, Class as well as related DCPS Entities.



Index

- C -

- Classifier
 - Default 18
- Code
 - C Generation 40, 41
 - C# Generation 40, 41
 - C++ Generation 40, 41
 - Generate 40, 41
 - Java Generation 40, 41
- Connector
 - DDS Application 28
 - DDS Domain 22
 - DDS Local Reconstruction 23
- Copyright Notice 3
- Create
 - DDS Project 30
 - DLRL Model 34
- Create DDS Project 30
 - Add DLRL Model 34

- D -

- DDS Options 17
 - General Options 17
 - Introduction 17
 - NDDS Options 18
 - OpenSplice DDS Options 20
 - OSPLDDS Options 20
 - Qos Property Behavior 17
 - RTIDDS Options 18
- DDS Project
 - Add DLRL Model 34
 - Create From Scratch 30
- DDS Technology Menu 16
- DDS Toolboxes 22
 - Application 28
 - Domain 22
 - Introduction 22
 - Local Reconstruction 23
 - QoS Policies 26
 - Types 25
- Default Classifiers 18
- Diagram
 - Application, Elements And Connectors 28
 - DLRL Element, Key Relations 23
 - Domain, Elements And Connectors 22

- Local Reconstruction, Elements And Connectors 23
- Qos Policies Library 26
- Types, Elements And Connectors 25

- E -

- Element
 - DDS Application 28
 - DDS Domain 22
 - DDS Local Reconstruction 23
 - DDS Type 25
 - Model Validation Rules 38
 - Qos Policies 26
- End User Licensing Agreement 4
- Example Model
 - Open 45

- G -

- Generate
 - C 40, 41
 - C# 40, 41
 - C++ 40, 41
 - Code 40, 41
 - Java 40, 41
 - PSM 40, 41
- Generate PSM and Code 40
 - by EA Transformation 41
 - from AddIn Command 40
 - Introduction 40
- Getting Started 10
 - Configure the DDS Modeling Environment 13
 - Register MDG Technology for DDS 11

- K -

- Key
 - DDS Local Reconstruction 23
 - DDS Type 25

- L -

- License Agreement 4

- M -

- MDG Technology For DDS
 - Acknowledgement of Trademarks 7
 - Copyright Notice 3
 - DDS Technology Menu 16

MDG Technology For DDS

Getting Started	10
License Agreement	4
Support	8
System Requirements	9
Using DDS	15
Welcome	2

Model Validation	38
Configuration	38
Perform	38
Rules for Elements	38

- N -

NDDS Generator	41
----------------	----

- O -

OpenSplice DDS	
Generate	40, 41
Options	20
OSPLDDS	
Generate	40, 41
Options	20

- P -

PIM	
Add DLRL Model	34
Create	30
Platform Independent Model	
Add DLRL Model	34
Create	30
Platform Specific Model	
DDS AddIn Generate Code	40
Generate	40, 41
Generate Executable Class	40
PSM	
DDS AddIn Generate Code	40
Generate	40, 41
Generate Executable Class	40

- Q -

Qos Policies	
Elements	26
Library Diagram	26
Toolbox	26
QoS Policy Specifications Dialog	18

- R -

RTIDDS	
Generate	40, 41
Options	18

- S -

Support	8
System Requirements	9

- T -

Trademarks	7
------------	---

- U -

Using DDS	15
-----------	----

- V -

Validation	
Of Model	38
Rules, For Elements	38

- W -

Welcome	2
---------	---

MDG Technology for DDS User Guide

www.sparxsystems.com