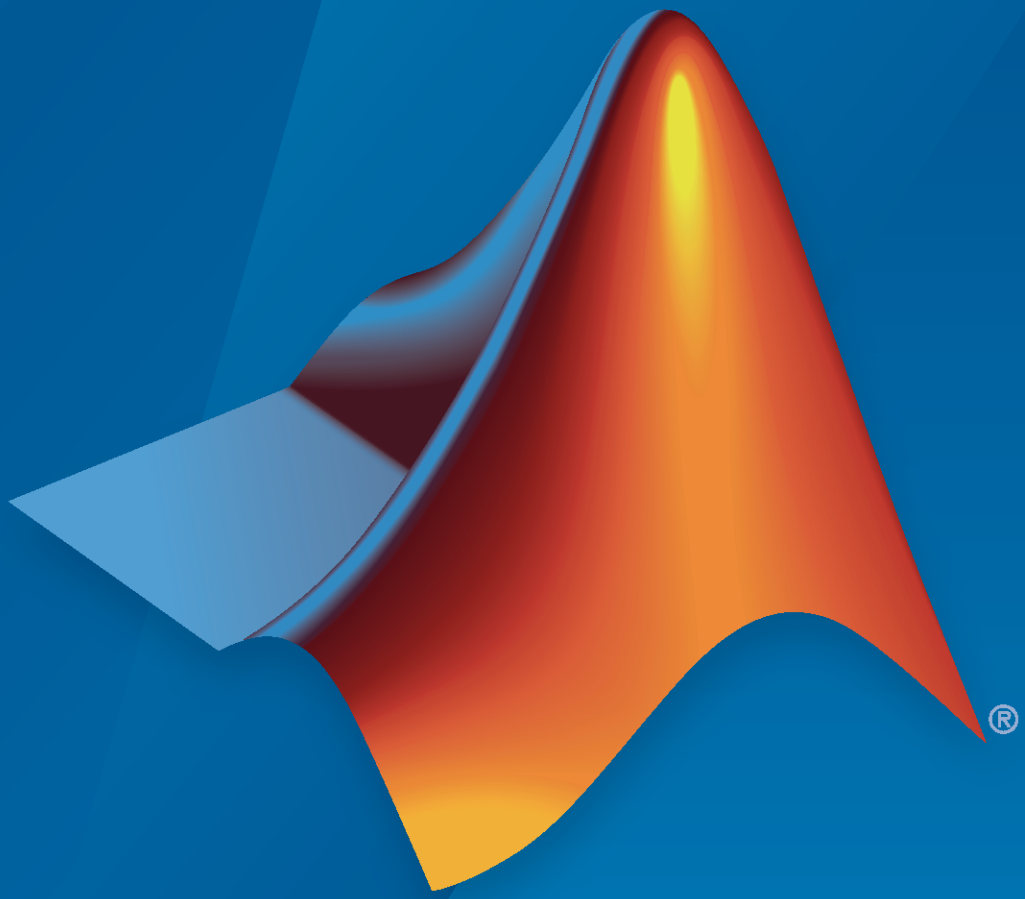
DDS Blockset

Getting Started Guide



MATLAB® & SIMULINK®

R2023a

MathWorks®

How to Contact MathWorks

Latest news: `www.mathworks.com`

Sales and services: `www.mathworks.com/sales_and_services`

User community: `www.mathworks.com/matlabcentral`

Technical support: `www.mathworks.com/support/contact_us`

Phone: 508-647-7000

The MathWorks, Inc.
1 Apple Hill Drive
Natick, MA 01760-2098

DDS Blockset Getting Started Guide

© COPYRIGHT 2021–2023 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

Trademarks

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See `www.mathworks.com/trademarks` for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

Patents

MathWorks products are protected by one or more U.S. patents. Please see `www.mathworks.com/patents` for more information.

Revision History

March 2021	Online Only	New for Version 1.0 (Release 2021a)
September 2021	Online Only	Revised for Version 1.1 (Release 2021b)
March 2022	Online Only	Revised for Version 1.2 (Release 2022a)
September 2022	Online Only	Revised for Version 1.3 (Release 2022b)
March 2023	Online Only	Revised for Version 1.4 (Release 2023a)

Contents

1 Product Tutorial

DDS Blockset Product Description 1-2

What Is DDS? 1-3
DDS Conceptual Tutorial 1-3
DDS Implementation 1-7

How does DDS Blockset Work? 1-8
Conceptual DDS Blockset Workflow 1-8
DDS Blockset workflow 1-12

Create DDS Applications in Simulink 1-14
Prerequisites 1-14
Example Model 1-14
DDS Workflow 1-14

Create DDS Definitions 1-15

View and Edit DDS Definitions 1-19

Construct Simulink Model as Publisher and Subscriber 1-22

Configure the DDS Interface 1-23

Build and Deploy Application to DDS Network 1-25

DDS Blockset Limitations and Considerations 1-26

DDS Blockset System Requirements 1-27
Supported Platforms for DDS Blockset 1-27
Supported Compilers to Build and Deploy DDS applications 1-27
Additional Vendor Setup 1-27

DDS Blockset Shapes Demo 1-29

Product Tutorial

- “DDS Blockset Product Description” on page 1-2
- “What Is DDS?” on page 1-3
- “How does DDS Blockset Work?” on page 1-8
- “Create DDS Applications in Simulink” on page 1-14
- “Create DDS Definitions” on page 1-15
- “View and Edit DDS Definitions” on page 1-19
- “Construct Simulink Model as Publisher and Subscriber” on page 1-22
- “Configure the DDS Interface” on page 1-23
- “Build and Deploy Application to DDS Network” on page 1-25
- “DDS Blockset Limitations and Considerations” on page 1-26
- “DDS Blockset System Requirements” on page 1-27
- “DDS Blockset Shapes Demo” on page 1-29

DDS Blockset Product Description

Design and simulate DDS applications

DDS Blockset provides apps and blocks for modeling and simulating software applications that publish or subscribe to Data Distribution Service (DDS) middleware. The blockset includes a DDS dictionary that lets you manage, create, and edit your DDS definitions in Simulink®. You can import DDS specifications as XML files to create a skeleton Simulink model as a starting point for developing algorithms for DDS applications.

DDS Blockset provides blocks for publishing and subscribing samples to DDS, including their corresponding Quality of Service (QoS). It fully integrates with the RTI Connext DDS and eProsima Fast DDS stacks.

DDS Blockset generates C++ code and XML files from Simulink models (with Embedded Coder®).

What Is DDS?

The Data Distribution Service (DDS) is a middleware standard that is ideal for production quality data communication. It is currently widely used in the fields of aerospace and defense, autonomous vehicles, as well as up-and-coming in robotics, medical devices, and power generation. DDS uses a publish-subscribe communication pattern to create a decentralized, architecture-independent, scalable, asynchronous network. To learn more about how the DDS middleware platform works, read through the DDS Conceptual Tutorial.

DDS Conceptual Tutorial

This tutorial covers the following aspects of DDS:

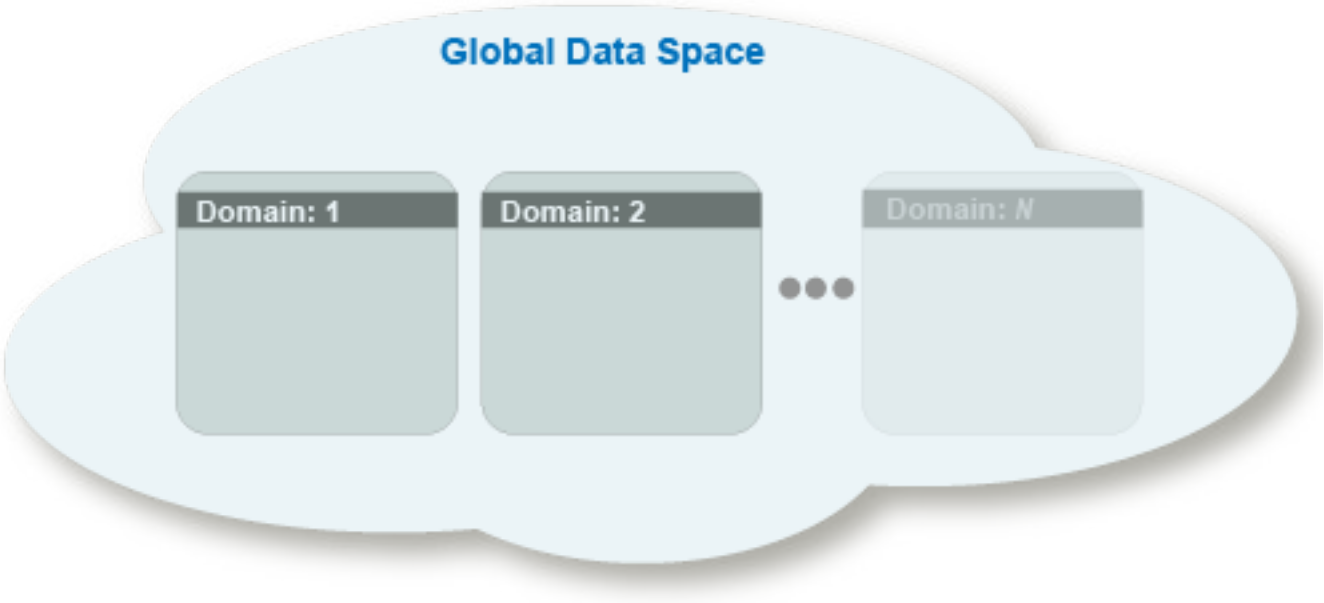
- “Global Data Space” on page 1-3
- “Domains” on page 1-3
- “Topics” on page 1-4
- “Publishers and Subscribers” on page 1-4
- “DataReaders, DataWriters, and Data Samples” on page 1-5
- “Quality of Service (QoS)” on page 1-6

Global Data Space

In DDS, the DDS network is referred to as the Global Data Space.

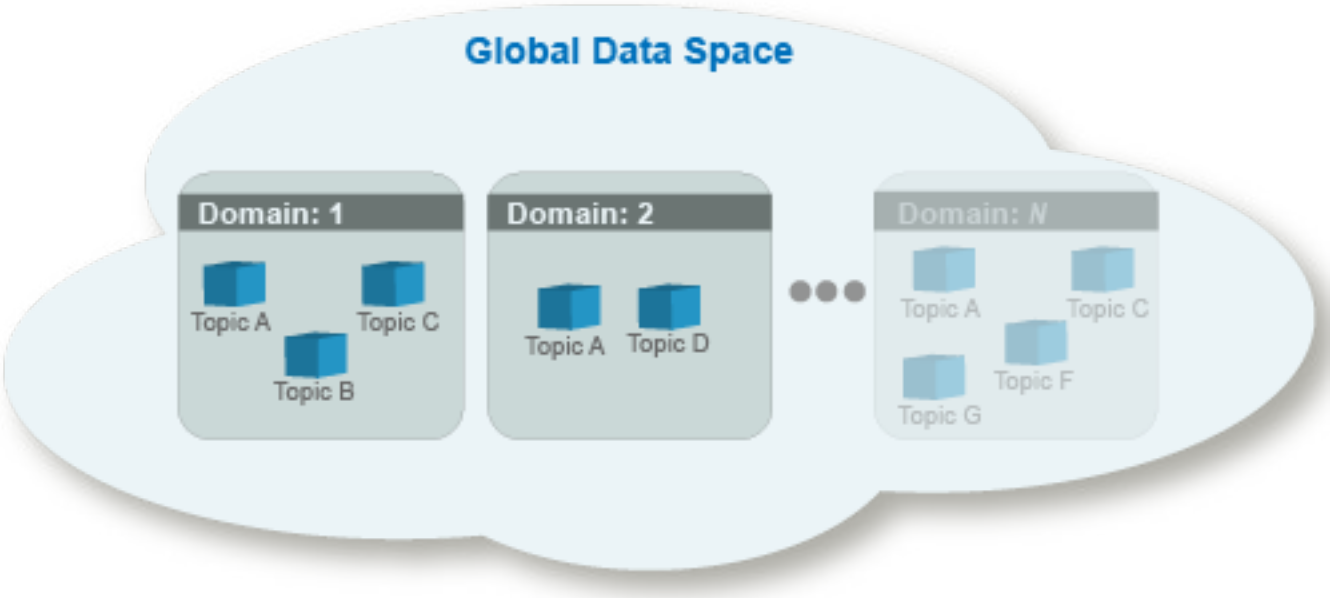
Domains

Within the Global Data Space, you can create a portion of the network referred to as a Domain. Domains are identified by a Domain name and a numeric Domain ID. There can be one or many Domains within the Global Data Space.



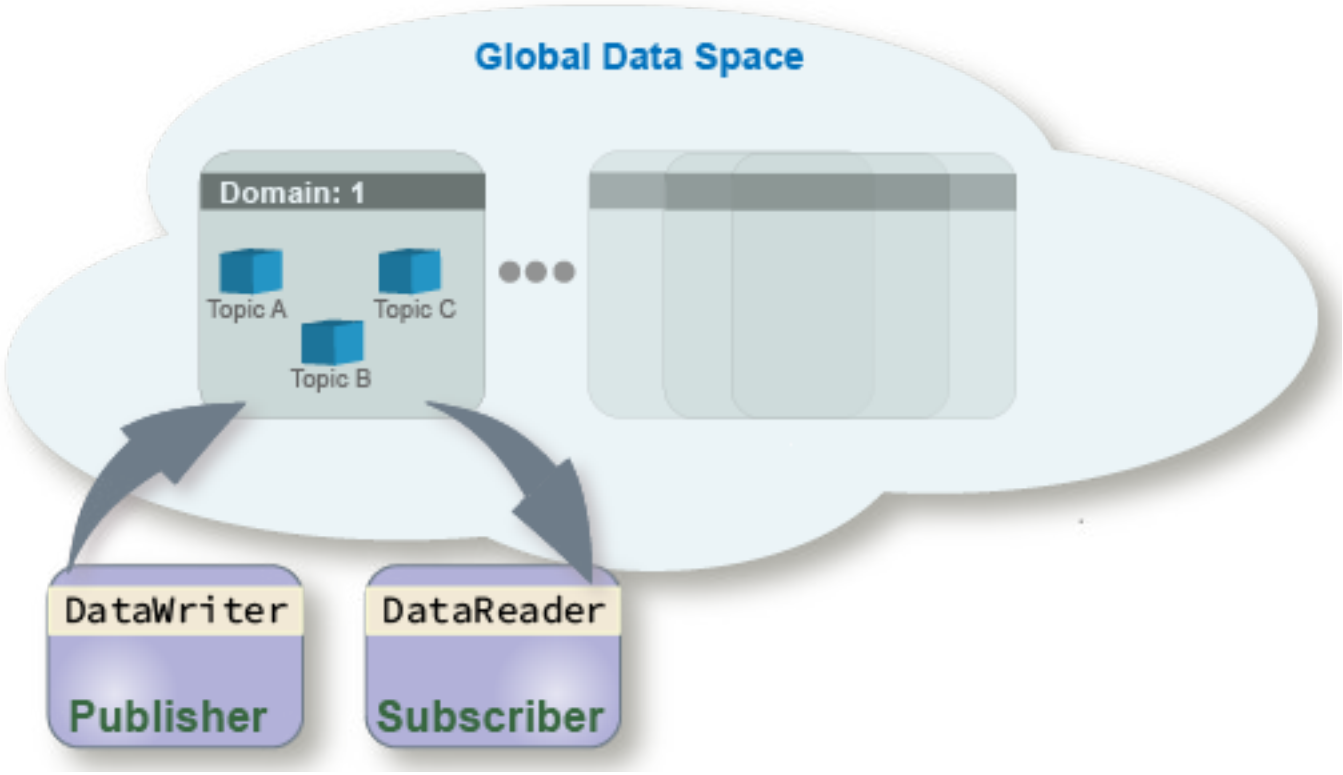
Topics

Within Domains there are Topics. Topics are categories that describe the type of data communicated on the network. For example, if you have an application that you would like to use DDS to receive information about the weather, a Topic could be Temperature. Domains can have one or many Topics. If you have the same Topic in different Domains they are considered distinct Topics.



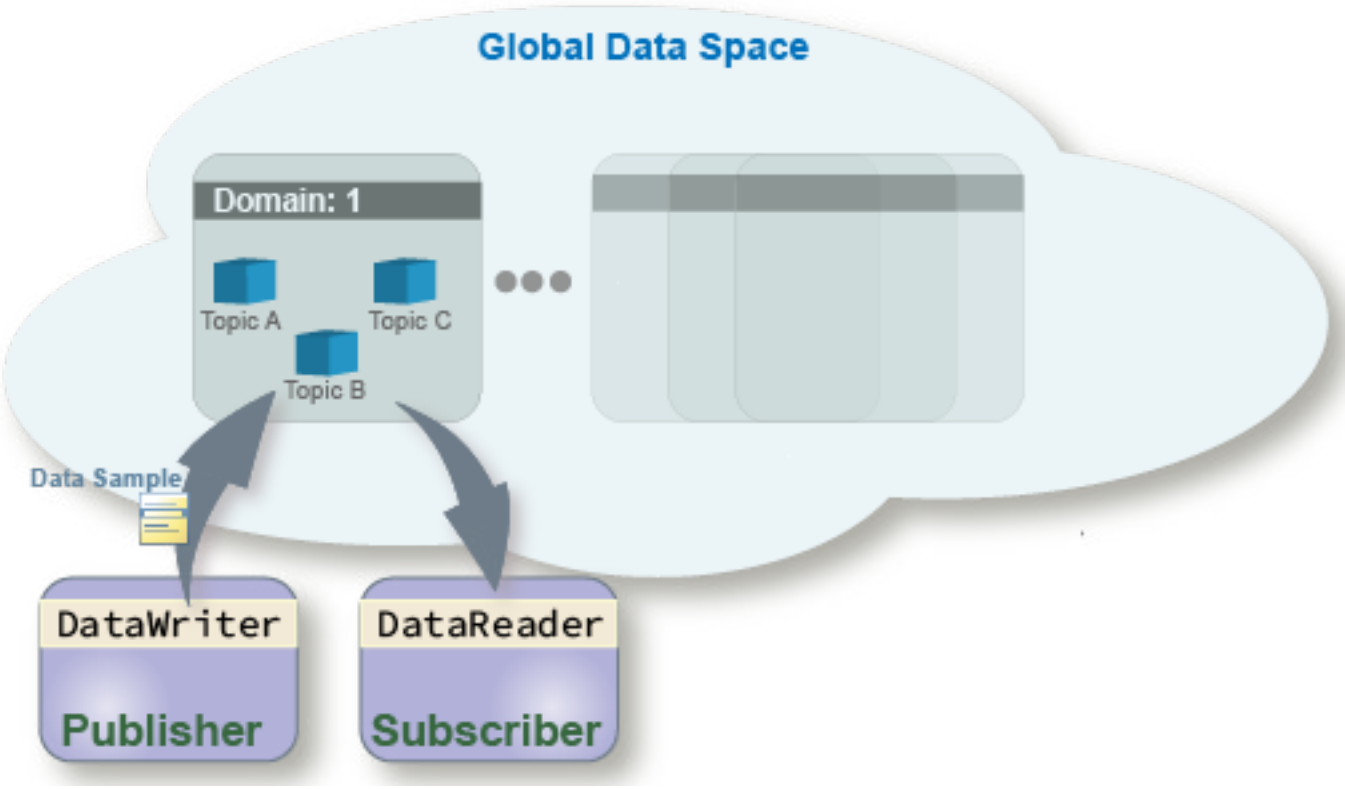
Publishers and Subscribers

Applications that use DDS are referred to a Domain Participants because they participate in a Domain. Applications that send data are referred to as Publishers and applications that receive data are referred to as Subscribers. Domain Participants can be Publishers, Subscribers, or both.



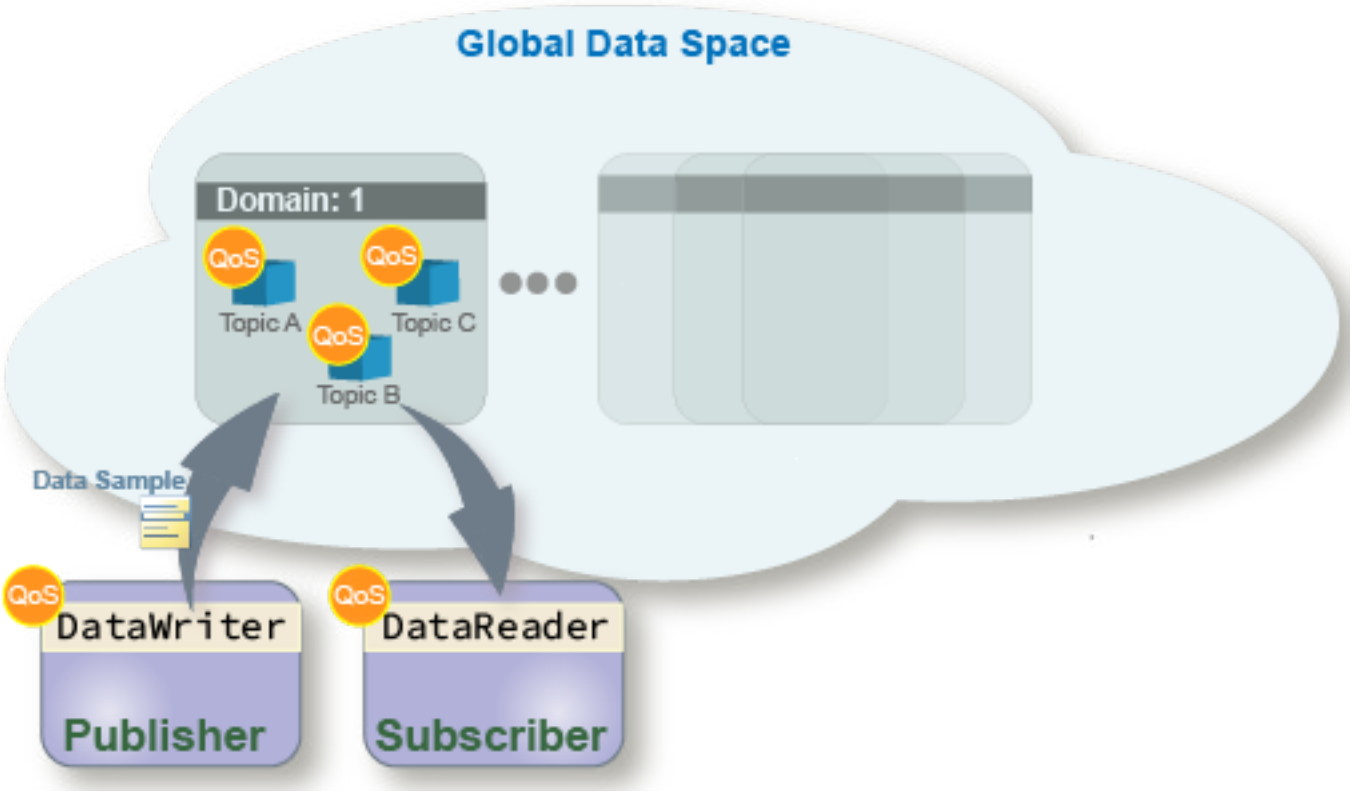
DataReaders, DataWriters, and Data Samples

Data is sent and received on a DDS network as a Data Sample, specified by a DDS data type. A Topic registers a DDS data type, then referred to as the Registered Type, to specify the data type that must be used to communicate about that Topic. A Publisher uses a DataWriter to send data on the DDS network. A Publisher can have one or many DataWriters that publish to one or many Topics. A Subscriber uses a DataReader to receive data from the DDS network. A Subscriber can have one or many DataReaders, that subscribe to one or many Topics.



Quality of Service (QoS)

In addition to being able to send and receive data, Quality of Service (QoS), which describes the quality of the data transport, can be specified for a Topic, a Publisher or Subscriber, or specific DataReaders or DataWriters. DDS is known to have several QoS policies to help ensure exact control over data transport. You can assign individual QoS policies to these entities, or you can assign a set of QoS policies referred to as a QoS profile.



DDS Implementation

DDS is specified by the Object Management Group (OMG) standard. The standard provides specifications that explain how the DDS network must be configured and how applications should use the DDS network. The DDS standard is implemented by several different DDS vendors. Each vendor can implement the DDS standard by using a programming language of its choice. Despite varied low-level implementations, DDS vendors are interoperable (an application that uses one vendor can communicate with an application on a different vendor) because its vendors implement to the DDS standard. DDS applications, composed of the definitions previously described (for example, Domains, Topics, Data Samples, Publishers, Subscribers) can be defined in a programming language agnostic way by using XML or IDL specifications. These XML and IDL specifications provide the ability to port DDS applications between different vendors with different programming language implementations.

See Also

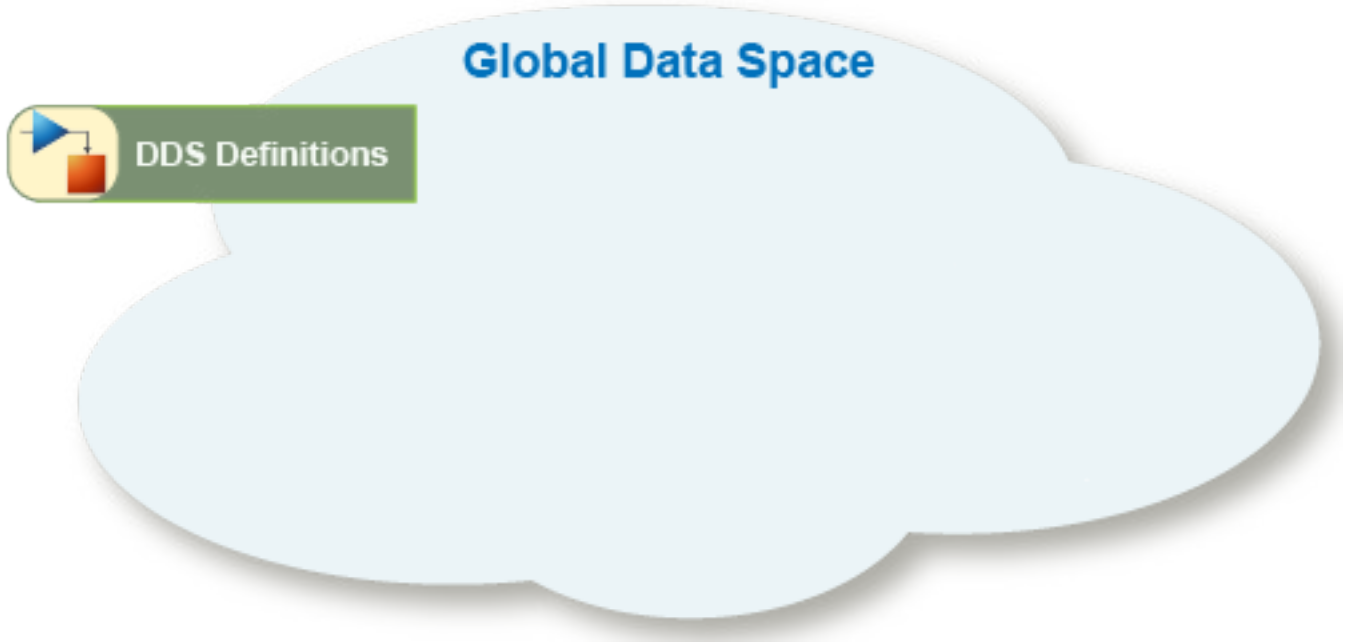
“How does DDS Blockset Work?” on page 1-8 | “Create DDS Applications in Simulink” on page 1-14

How does DDS Blockset Work?

The DDS Blockset enables you to create DDS applications in Simulink. If you have an application modeled in Simulink that you would like to be able to use DDS, you can use the DDS Blockset to easily connect to the DDS middleware platform. The DDS Blockset brings DDS concepts into the Simulink environment, models those concepts in your Simulink application, and then enables you to connect your application to DDS.

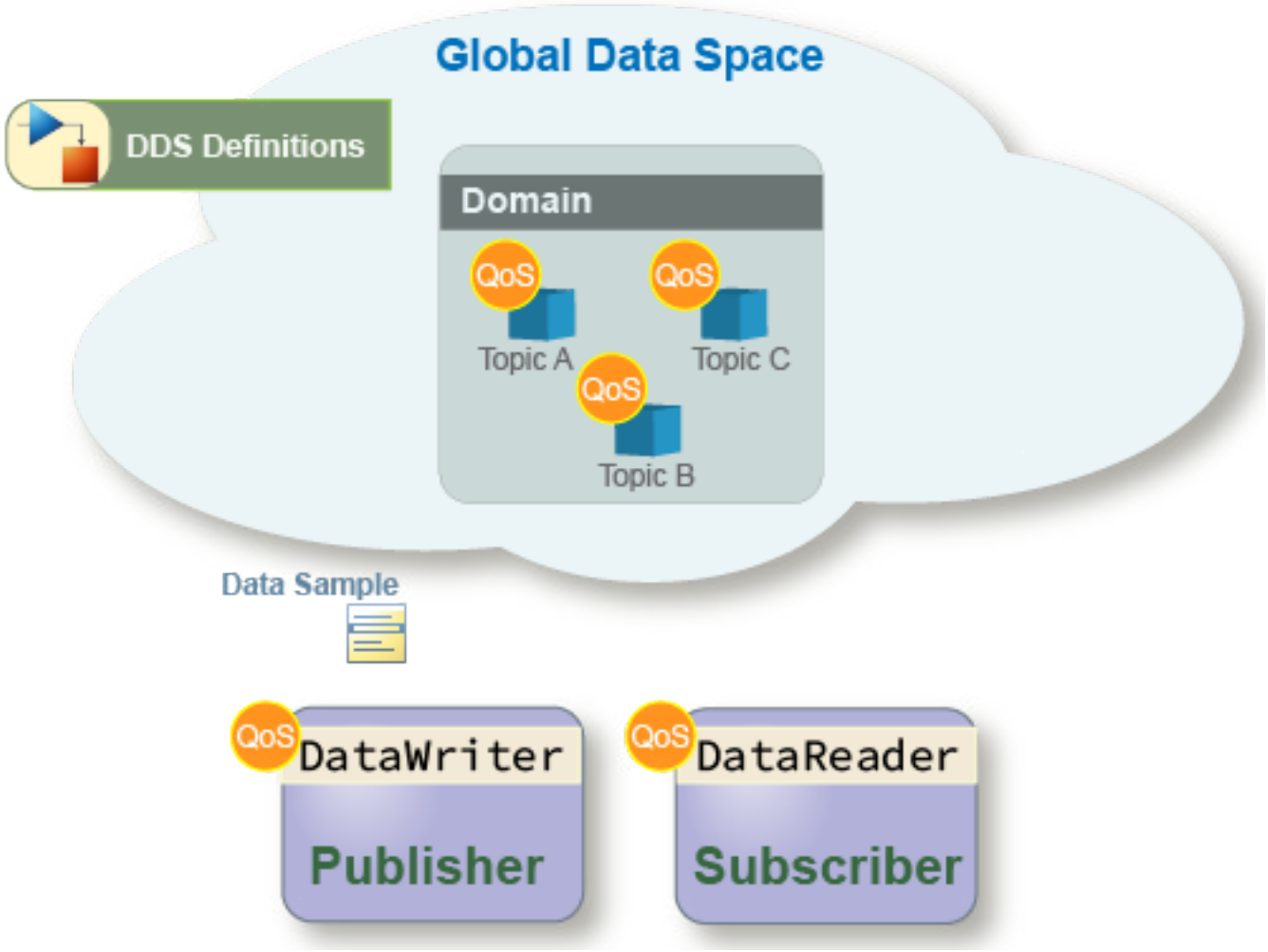
Conceptual DDS Blockset Workflow

You start with the DDS network and the DDS definitions required for your DDS application.



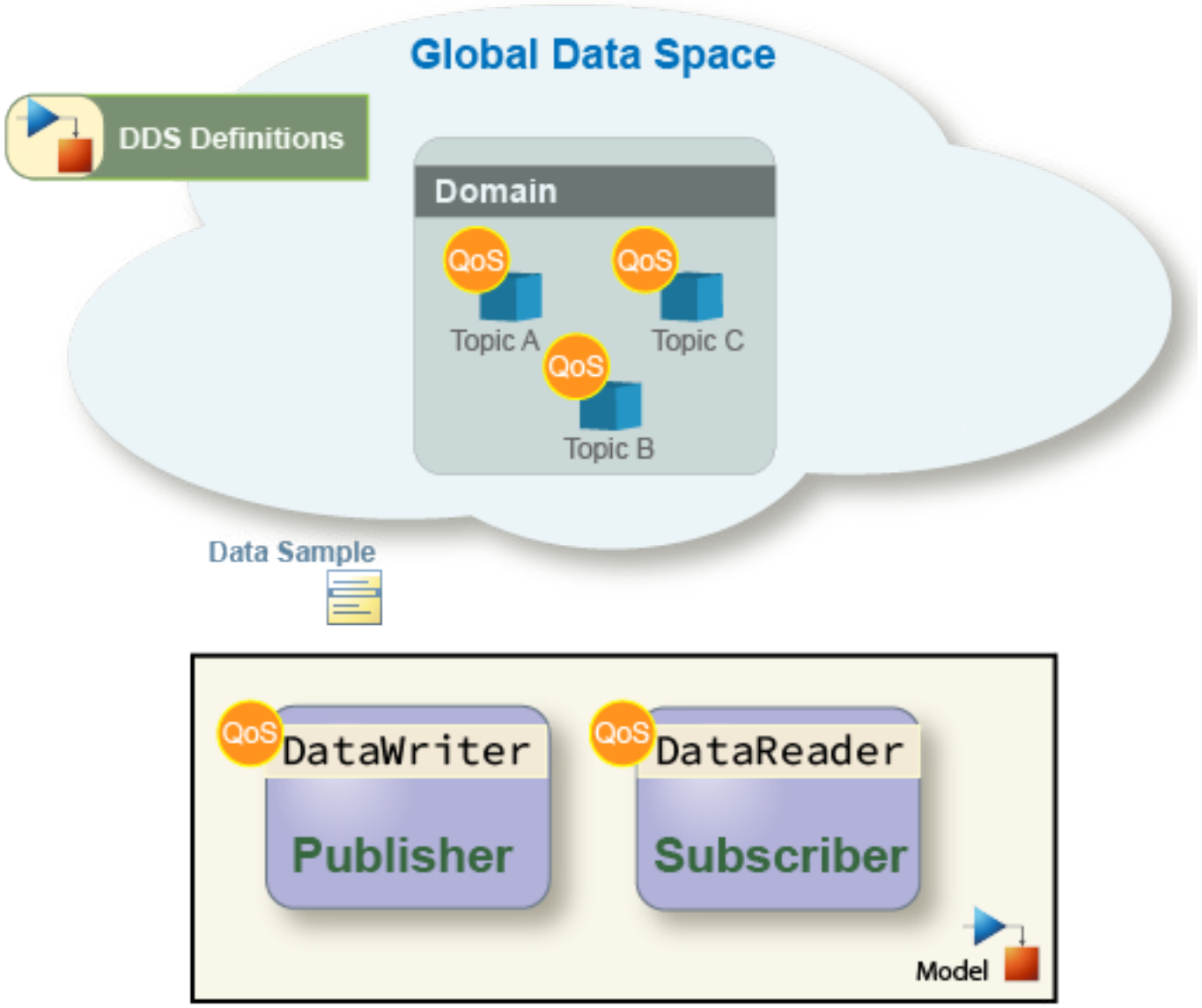
Create DDS Definitions

You then bring your DDS definitions into the Simulink environment. You can either import or create these definitions. Once the definitions are in the Simulink environment, Domains, Topics, Types, Quality of Service (QoS), and Domain Participant information is available in Simulink.

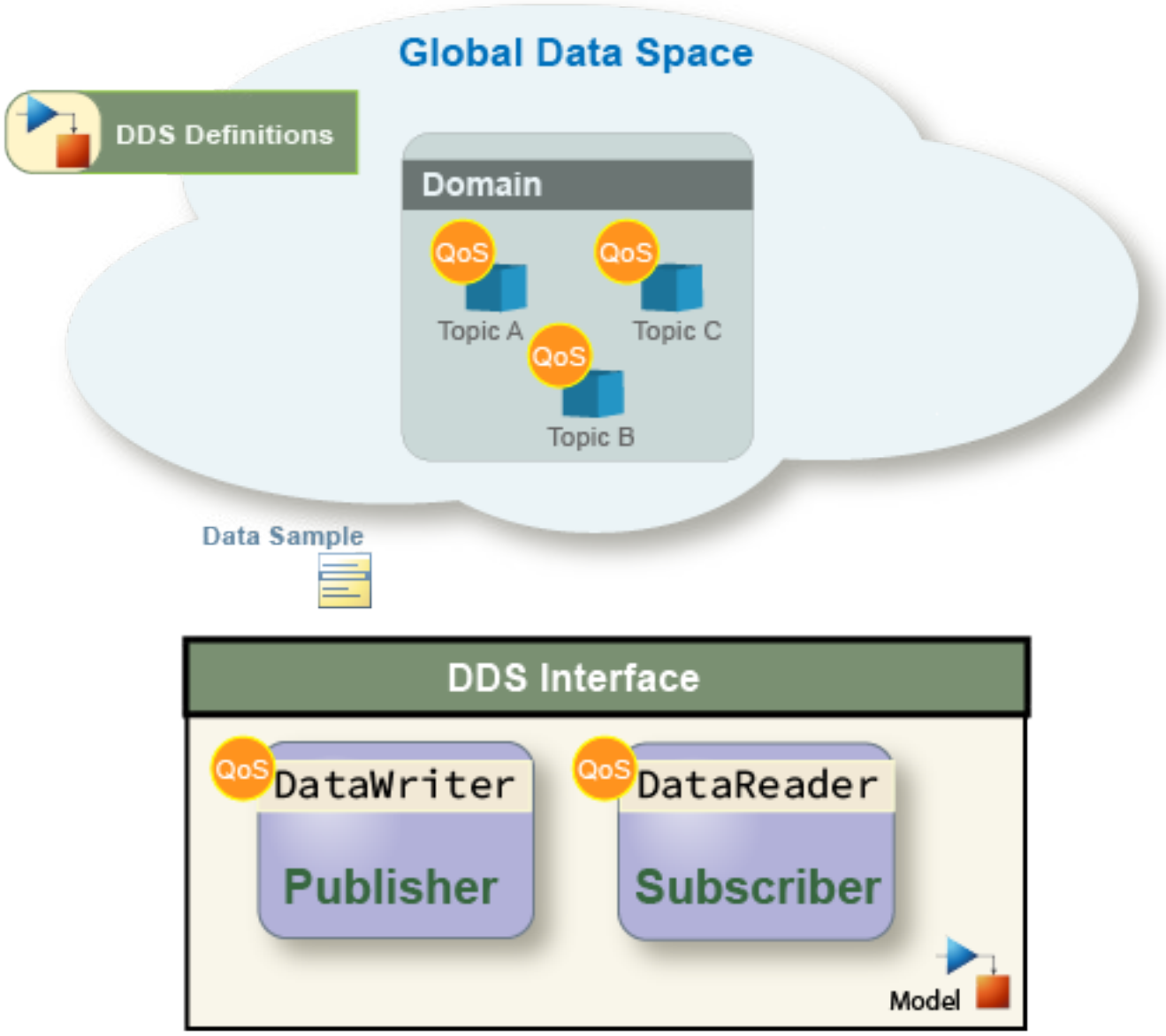


Model DDS Applications in Simulink

You can now construct or adapt a Simulink model so that it behaves as a Publisher, Subscriber, or an application that is both a Publisher and Subscriber.

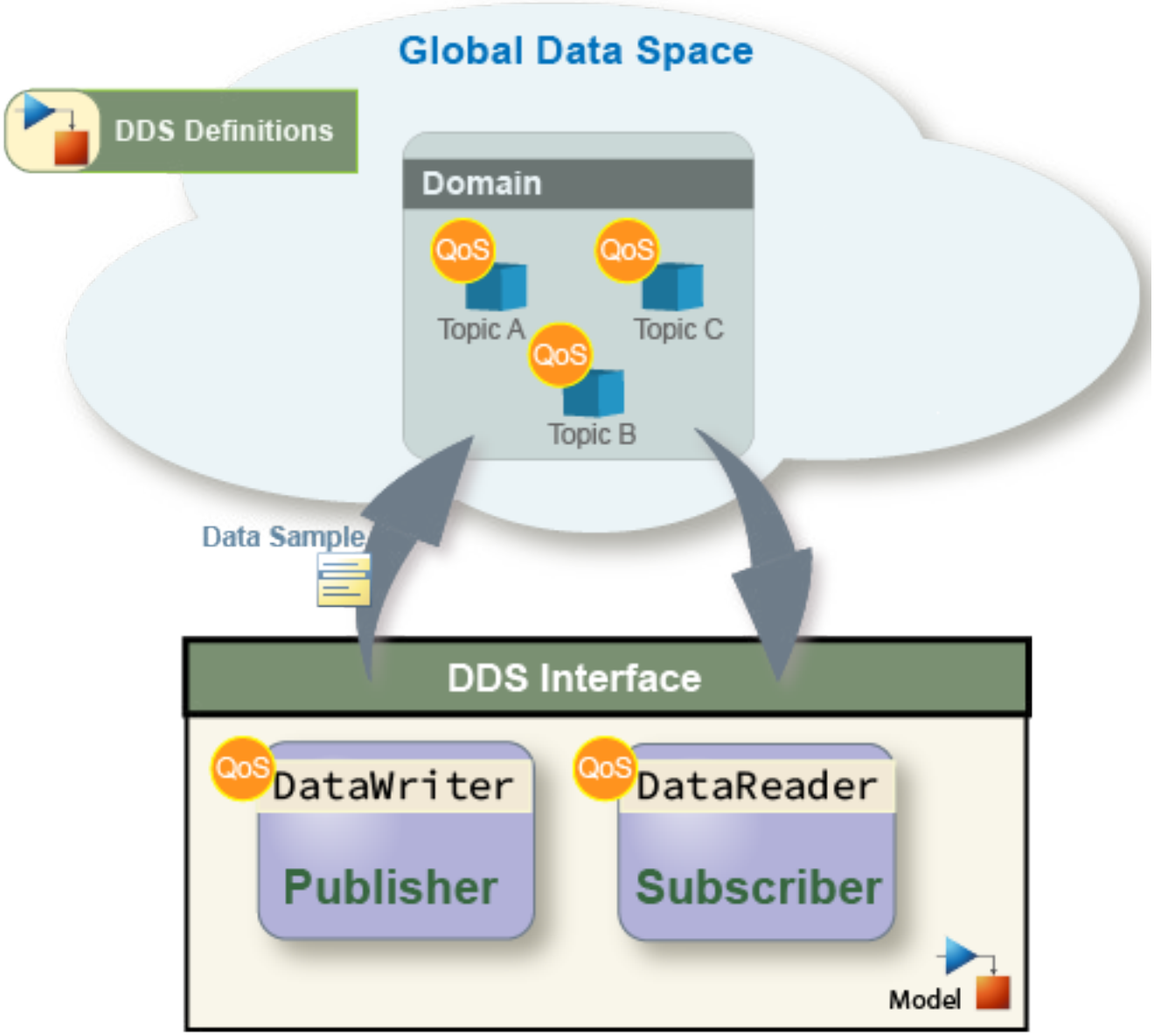


You then configure the DDS interface. The DDS interface is the connection point between the application and the DDS network. Specifically, when you use the DDS Blockset, the interface is the Topic and QoS specified on the DataWriters and DataReaders for your application.



Deploy Applications to DDS Network

After you have appropriately modeled your DDS application, build the model and use the executable to deploy your application on the DDS network. Now your DDS application can communicate with other applications by using DDS.



The DDS Blockset provides out-of-the-box support for the DDS vendors RTI and eProsima.

DDS Blockset workflow

To use the DDS Blockset tools to complete these conceptual steps:

1. Use the DDS Application Quick Start to import or create DDS Definitions.
2. View or edit DDS definitions in the DDS Dictionary.
3. Use DDS Blocks to model a Publisher or Subscriber.
4. Use the Code Mappings editor to configure the DDS interface (the Topic and QoS properties of DataReaders and DataWriters).

5 Build the application executable to deploy the application on DDS network.

See Also

“What Is DDS?” on page 1-3 | “Create DDS Applications in Simulink” on page 1-14

Create DDS Applications in Simulink

DDS Blockset enables an easy way to configure applications modeled in Simulink to use DDS.

Prerequisites

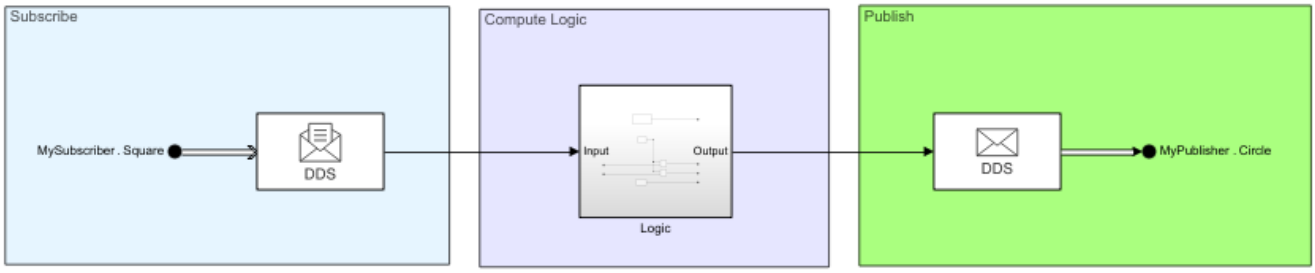
This tutorial requires the following products:

- MATLAB®
- Simulink

Additionally, to build and deploy a DDS application, you must have Embedded Coder (requires Simulink Coder™ and MATLAB Coder).

Example Model

This tutorial uses the Shapes Demo example model, `shapesdemo`.



DDS Workflow

The general workflow to create a DDS application is the following:

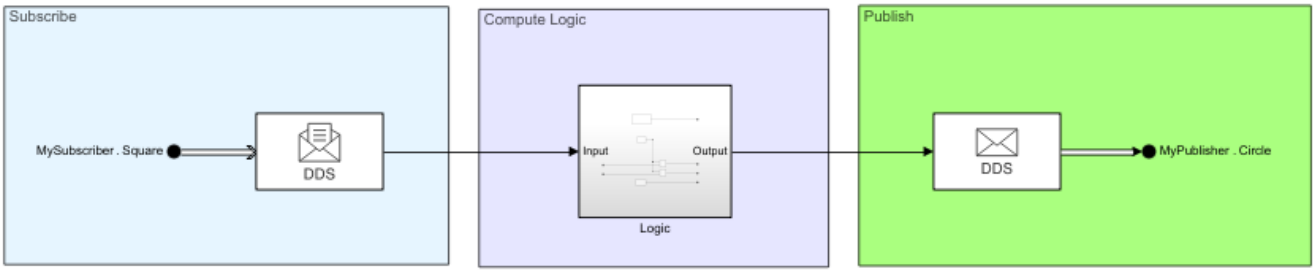
1. Create DDS definitions.
2. View and edit DDS definitions.
3. Construct the Simulink model as a DDS Publisher or Subscriber.
4. Configure the DDS interface.
5. Build and deploy the application to the DDS network.

To start the tutorial, see “Create DDS Definitions” on page 1-15.

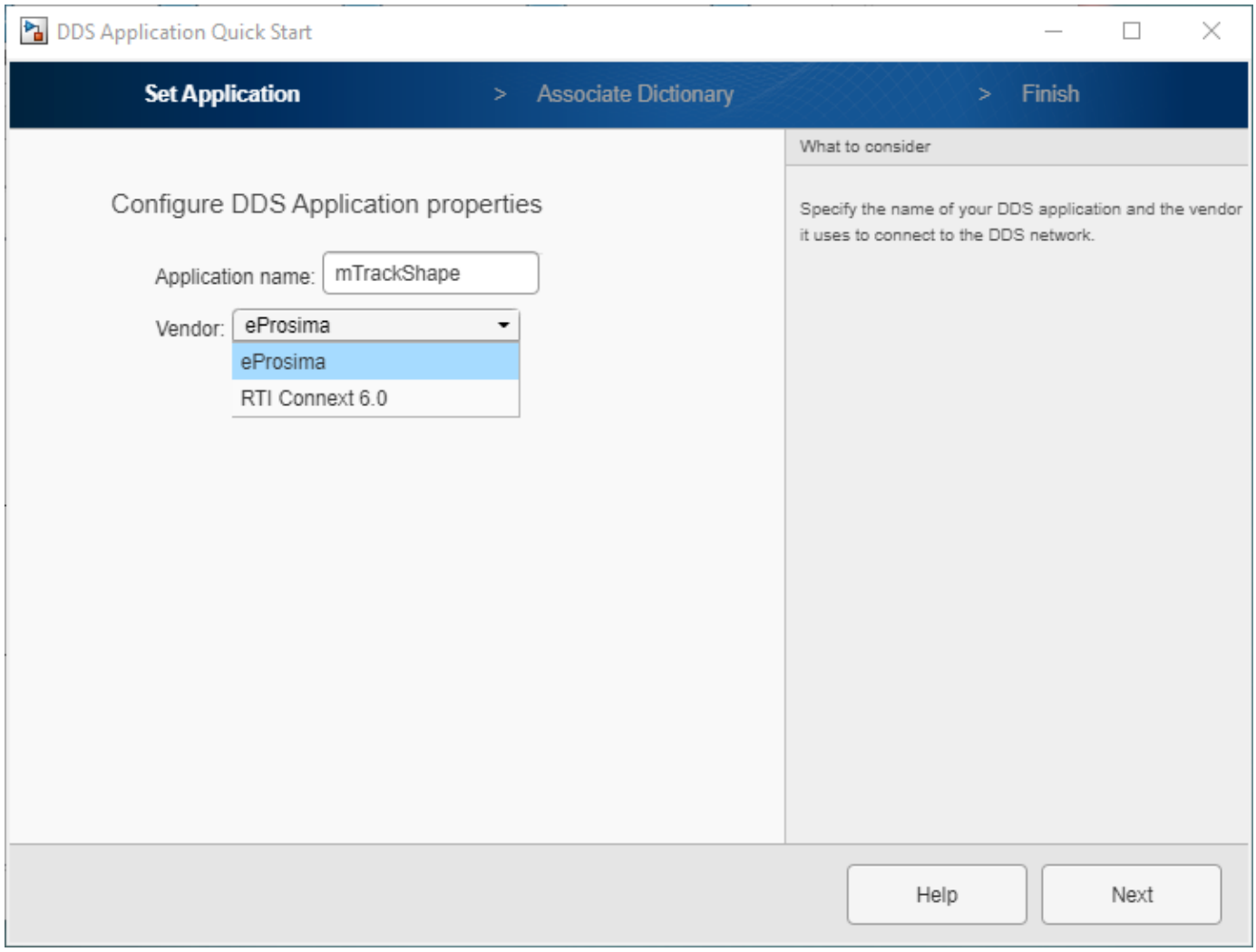
Create DDS Definitions

DDS Blockset enables applications modeled in Simulink to use DDS by bringing DDS concepts such as Domains, Topics, Types, and Quality of Service (QoS) into the Simulink environment. To bring these concepts into Simulink, you can use the DDS Application Quick Start to import DDS definitions from XML, use previously defined definitions stored in Simulink, or create default definitions from the application model. For this example, you can create DDS definitions from the application model.

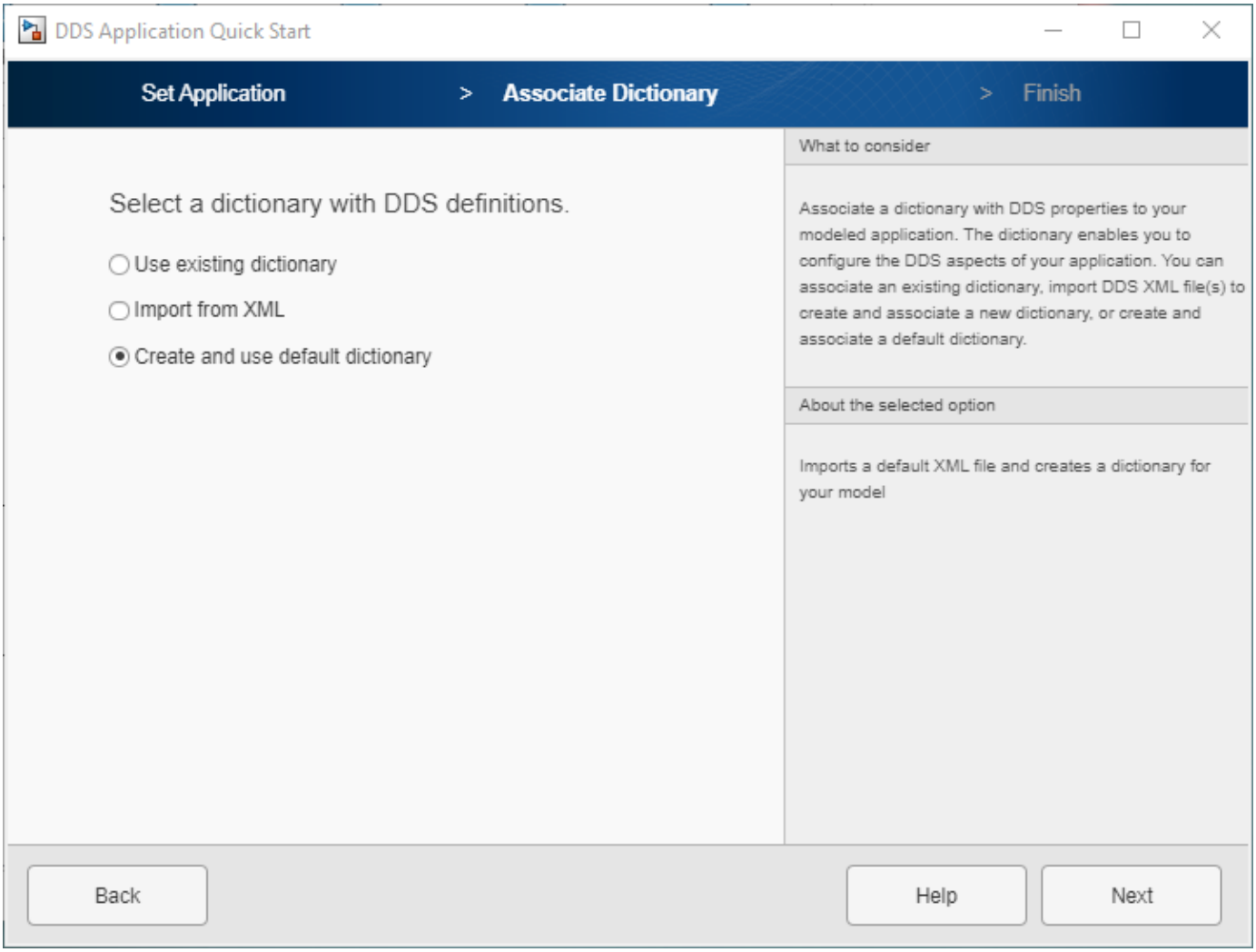
1. Open the model `shapesdemo`.



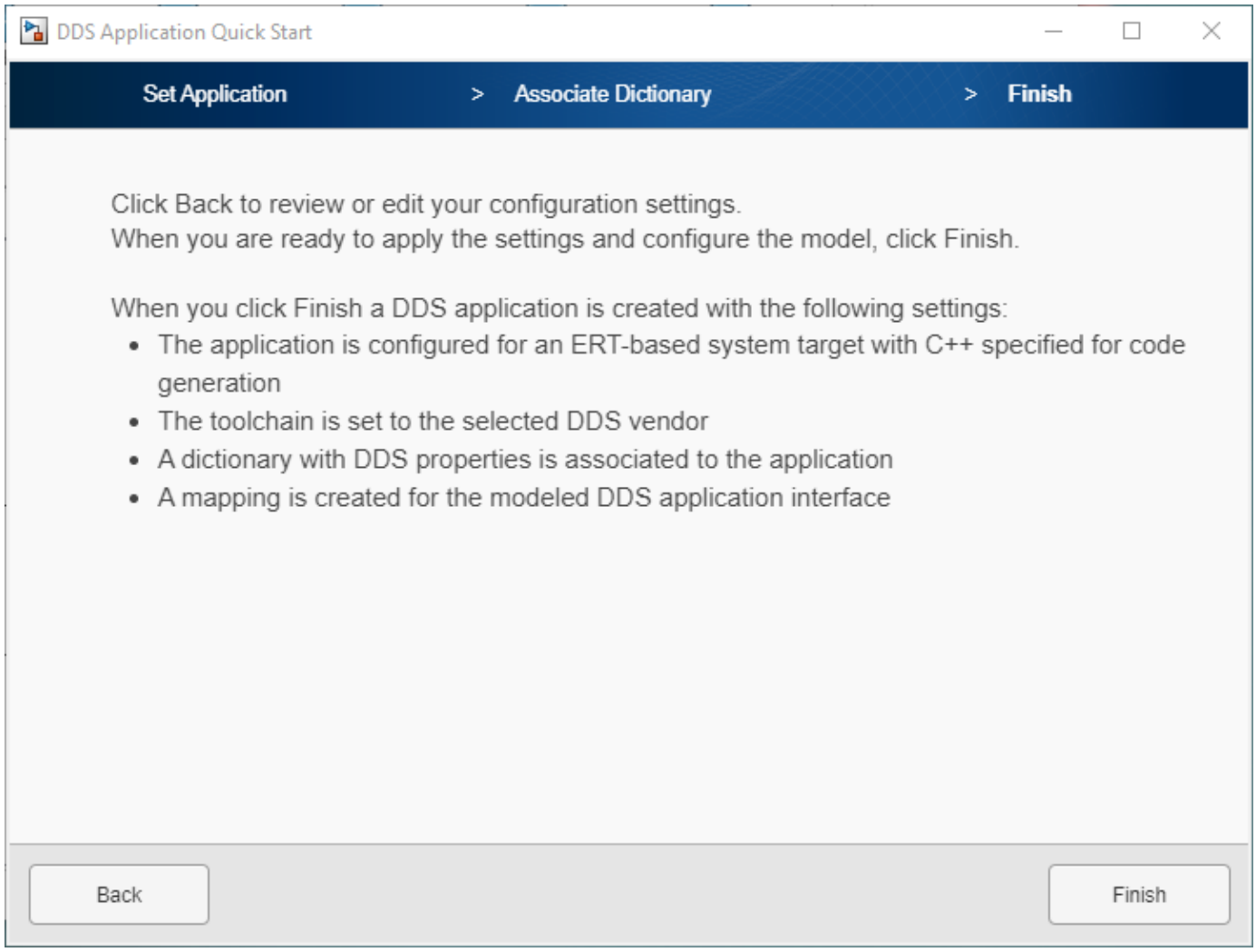
2. Open the DDS Blockset app. In the Apps Gallery, click **DDS Application Designer**.
3. When you start the DDS Blockset app, the DDS Application Quick Start opens. Walk through the Quick Start steps:
   a. Specify the DDS application name and vendor. Leave the application name as its default, which is the name of the application model, `shapesdemo`. Select the DDS vendor that you want to use for your application. For this example, select eProsima.



b Specify the source of the DDS definitions. You can import DDS definitions from XML, use previously defined definitions stored in Simulink, or create default definitions from the application model. For this example, select **Create and use default dictionary**.



c To wrap up the transition from DDS to Simulink, click **Finish**. When you finish the DDS Application Quick Start, the app creates your DDS application, associates it with the DDS Dictionary, and configures it appropriately for eProsima.



See Also

Related Examples

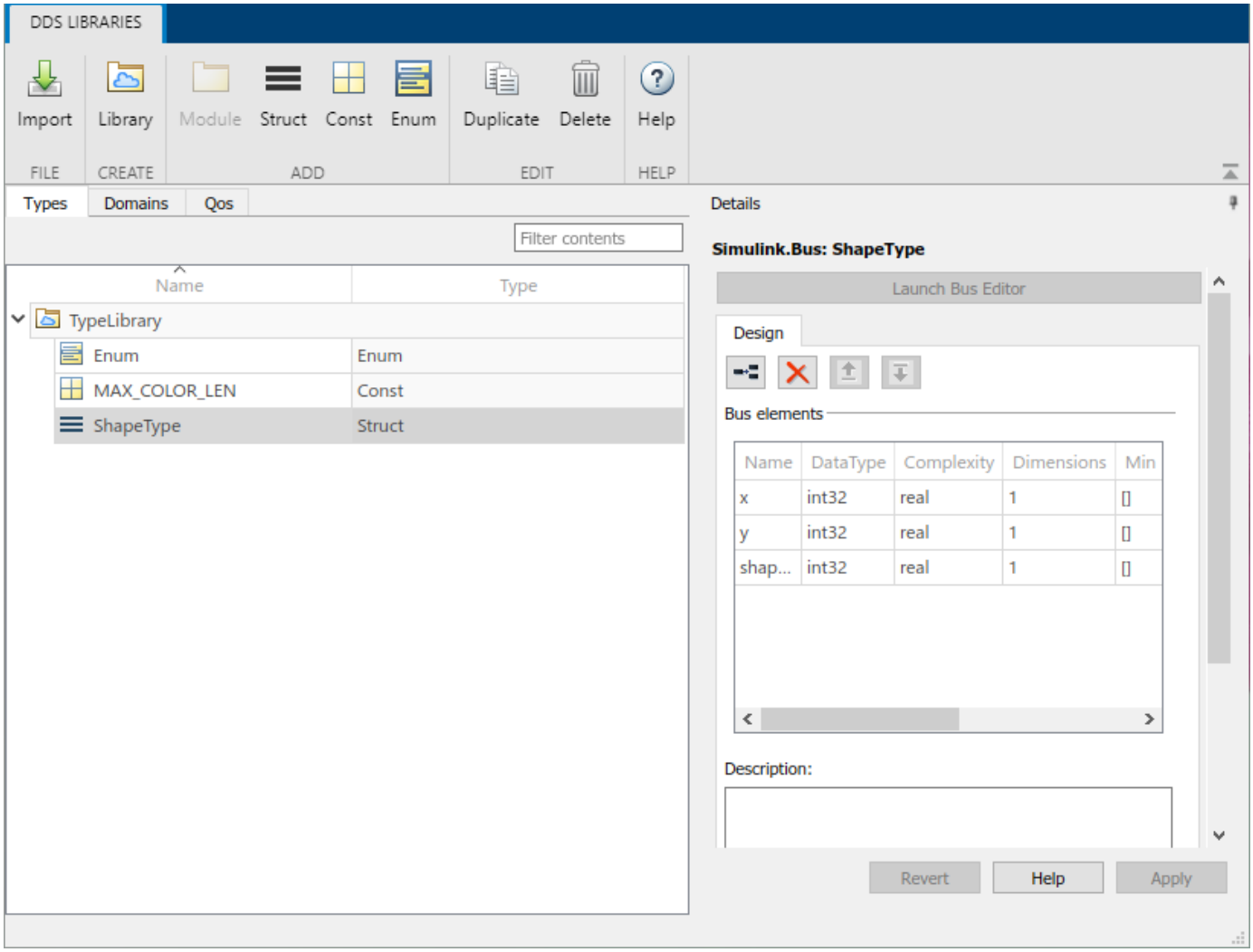
- “Import or Create DDS Definitions”

View and Edit DDS Definitions

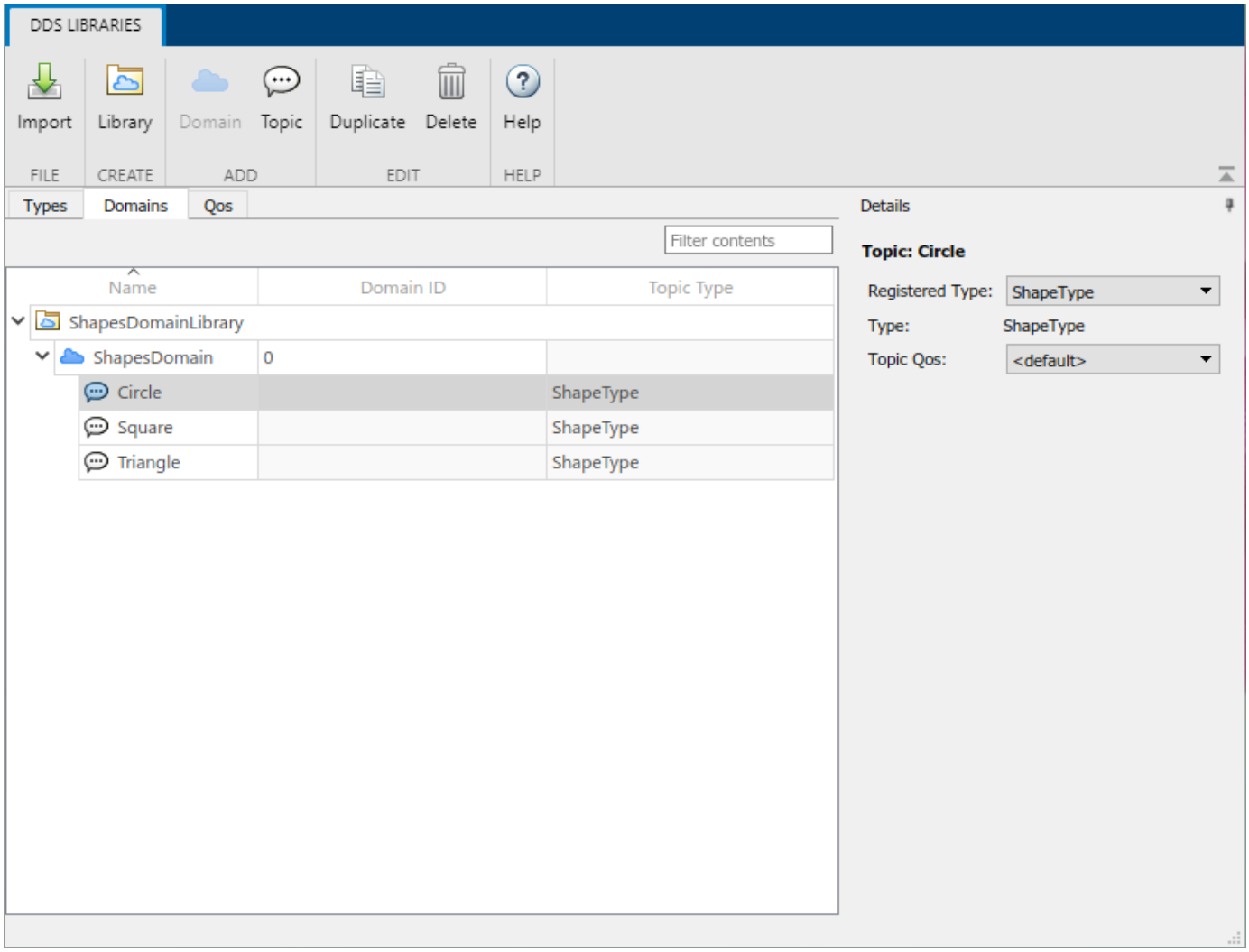
After the DDS Application Quick Start creates your default DDS definitions, these definitions are stored in the DDS Dictionary. You use these definitions to configure the DDS aspects of your application. You can view and edit these definitions at any time.

To view your recently created DDS definitions:

1. Open the DDS Dictionary. To open the DDS Dictionary, on the **DDS** tab, click **Code Interface** and select **DDS Dictionary**.
2. View your created default DDS Types. In the open **Types** pane, click the `TypeLibrary` to view the DDS data types. Click each DDS data type to view the equivalent Simulink data type in the **Details** pane.

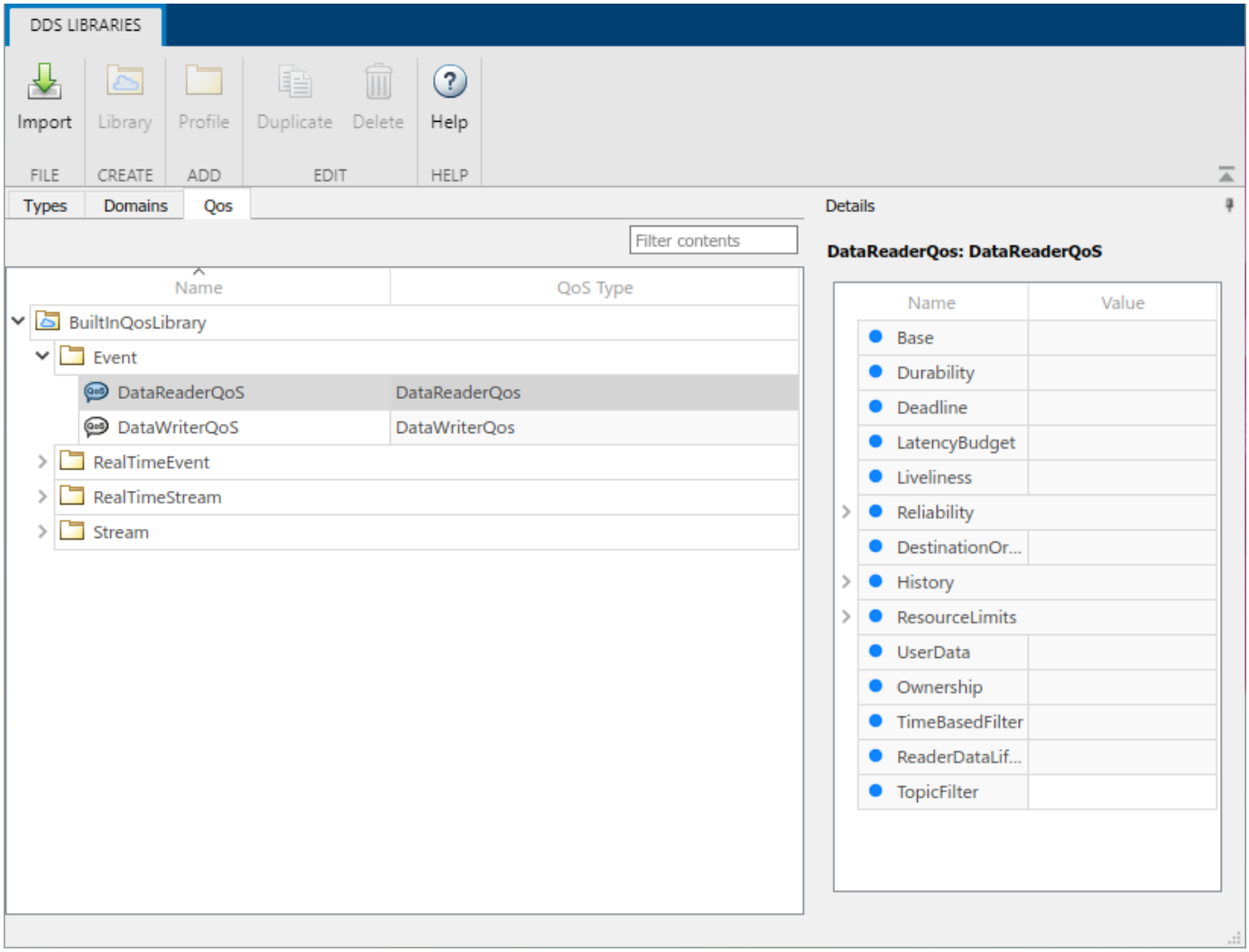


3. View your created default Domains and Topics. Click the **Domains** tab. You can view the created Library (`ShapesDomainLibrary`), Domain (`ShapesDomain`), and Topics (`Circle`, `Square`, `Triangle`). Also note that in the **Details** pane, you can view and set the Registered Type for each Topic.



4 View your created default Quality of Service (QoS) profiles. Click the **QoS** tab. You can view the default Library, `BuiltInQosLibrary`, composed of four QoS profiles specified by application type:

- Event-based applications
- Real-time event-based applications
- Real-time stream applications
- Stream applications



See Also

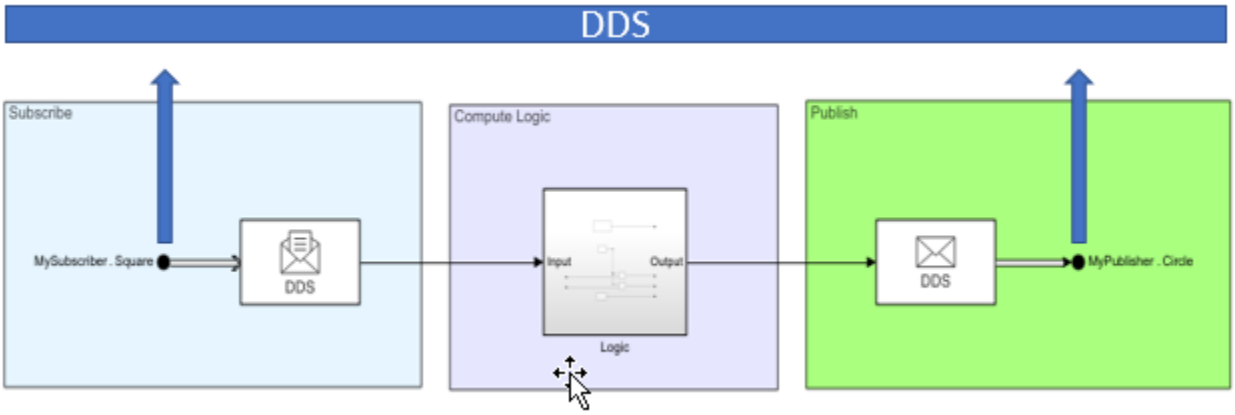
Related Examples

- “Manage DDS Definitions”

Construct Simulink Model as Publisher and Subscriber

To publish and subscribe to the DDS network, the application model for the Shapes Demo has already been configured as a top model that meets the necessary modeling aspects:

1. The model has inports and outports that are set to DDS data types.
2. The model has message blocks, specifically the Write DDS Sample and Take DDS Sample blocks, that send and receive messages from the DDS network. These blocks convert between DDS and Simulink data types to enable the model to publish and subscribe to the DDS network.
3. The model itself contains the DDS application logic. The logic portion of the model is independent of the middleware. It uses the data communicated from the middleware in a Simulink compliant form to run the application algorithm.



To explore the publish and subscribe aspects of the application model:

1. Click the inport and view its DDS data type.
2. Click the Take DDS Sample block that converts the DDS data type to the Simulink data type.
3. Examine the logic component of the application model. The logic component uses the Simulink data types to compute the logic of the DDS application.
4. Click the Write DDS Sample block that converts Simulink data types to a DDS data type.
5. Click the outport and view the DDS data type on the port that sends the data to the DDS network.

See Also

Related Examples

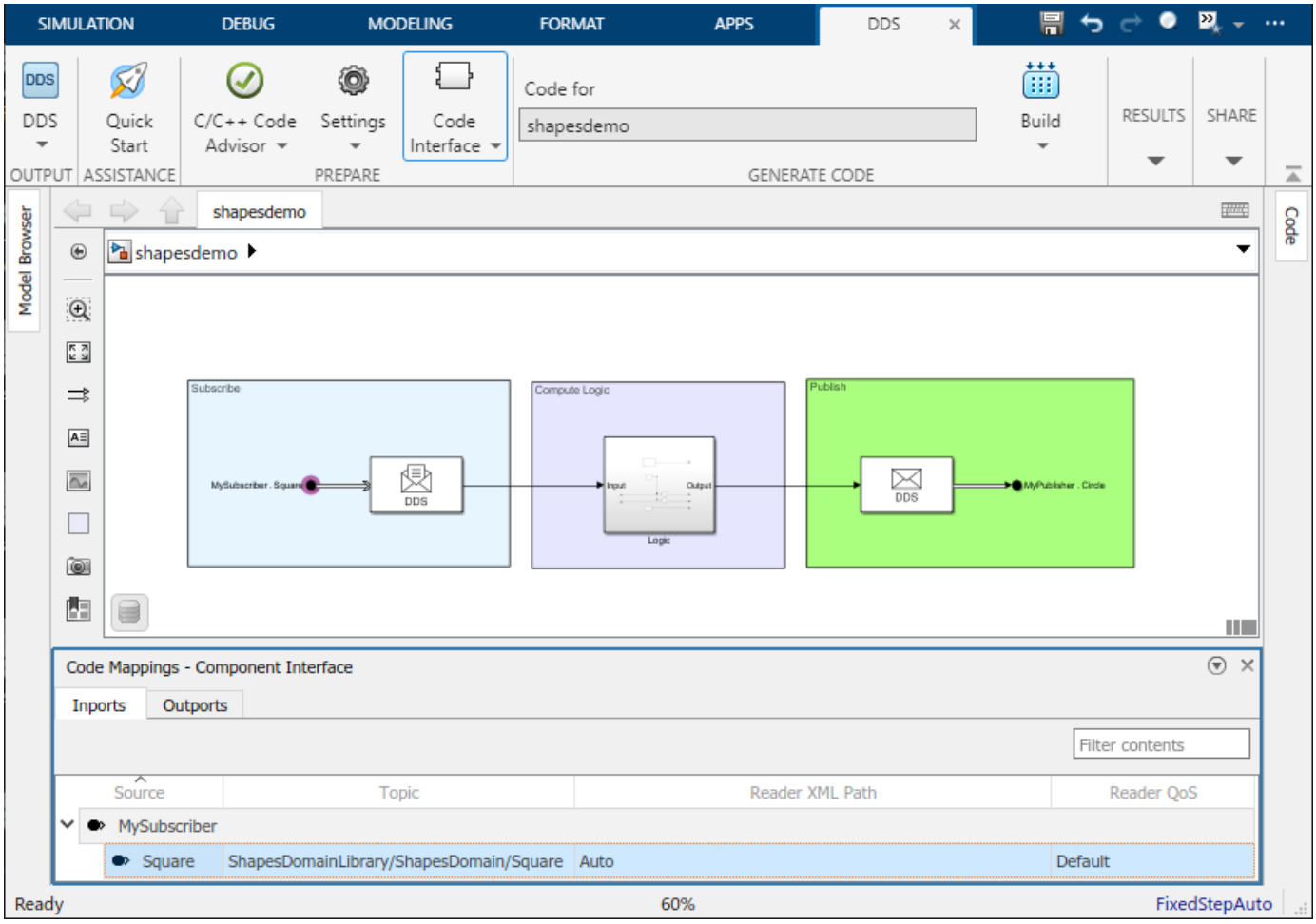
- “Model DDS Applications”

Configure the DDS Interface

The DDS interface is the connection point between your DDS application and the DDS network. Specifically, the connection point for a DDS application is the specification of the Topic and Quality of Service (QoS) for the DataReaders and DataWriters of the application.

To configure the DDS interface for the Shapes Demo:

1. Open the Code Mappings editor. On the **DDS** tab, click **Code Interface** and select **Individual Element Code Mappings**.



2. Configure the Topic, Reader, and QoS for the inport that acts as a DataReader. In the Code Mappings editor, click the **Inports** tab:

   a. Select the Topic. For the **Topic**, select the Square Topic specified in the drop-down options by its path (`ShapesDomainLibrary/ShapesDomain/Square`).

   b. Select the DataReader. For the **Reader XML Path**, select `Auto` to have DDS Blockset automatically generate a reader for the application.

   c. Select the Quality of Service (QoS). For the **Reader QoS**, select from the following list of options:

- You can leave the QoS policy as the default. The default QoS profile is the default provided by your target DDS vendor, eProsima
- You can select from built-in QoS profiles. The built-in QoS profiles automatically appear when you use the Quick Start default dictionary option.
- You can select from QoS profiles imported from XML. Imported QoS profiles also appear as options in this drop-down.

For this example, select `Default`.

3. Configure the Topic and QoS for the outport that acts as a DataWriter. In the Code Mappings editor, click the **Outports** tab:

   a. Select the Topic. For the **Topic**, select the Circle Topic specified in the drop-down options by its path (`ShapesDomainLibrary/ShapesDomain/Circle`).

   b. Select the DataWriter. For the **Writer XML Path**, select `Auto` to have DDS Blockset automatically generate a reader for the application.

   c. Select the Quality of Service (QoS). For the **Writer QoS**, select from the following list of options:

   - You can leave the QoS policy as the default. The default QoS profile is the default provided by your target DDS vendor, eProsima
   - You can select from built-in QoS profiles. The built-in QoS profiles automatically appear when you use the Quick Start default dictionary option.
   - You can select from QoS profiles imported from XML. Imported QoS profiles also appear as options in this drop-down.

   For this example, select `Default`.

See Also

Related Examples

- “Interactively Configure DDS Interface”
- “Manage QoS”

Build and Deploy Application to DDS Network

DDS Blockset enables out-of-the-box support for your previously selected vendor, eProsima. To use this out-of-the-box support, build your application model and deploy your generated executable in your development environment:

1. Ensure that your model is configured correctly. Verify that the model ports are configured and mapped appropriately for DDS.
2. Set up your build environment. Verify that a C++ compiler is appropriately set up for your platform. For eProsima, no additional setup is required (If you had selected RTI, you would need to install RTI Connext).
3. Build the Shapes Demo application model. On the **DDS** tab, click **Build**.
4. Run the generated executable.

See Also

Related Examples

- “Deploy DDS Applications”
- “DDS Blockset System Requirements” on page 1-27

DDS Blockset Limitations and Considerations

DDS Blockset does not support the following import, modeling, simulation, and deployment capabilities:

- Import — DDS Blockset does not directly support IDL import for eProsima. To import IDL for eProsima, you must convert IDL to XML and import the XML equivalent files.
- Modeling — DDS Blockset requires that all ports for an application model must map to DDS. If you have ports that do not map DDS, the model does not simulate or build correctly.
- Simulation — DDS Blockset does not support Rapid Accelerator mode simulation, external mode simulation, software-in-the-loop (SIL) testing, or MAT- file logging for DDS applications.
- Deployment — DDS Blockset does not support compiling the code generated from a DDS application model for a non-DDS application.

See Also

“DDS Blockset System Requirements” on page 1-27

DDS Blockset System Requirements

To create and model DDS applications, DDS Blockset requires:

- MATLAB
- Simulink

To build and deploy DDS applications, DDS Blockset requires Embedded Coder (requires Simulink Coder and MATLAB Coder).

Supported Platforms for DDS Blockset

You can run DDS Blockset on the following platforms:

Platforms

- Windows® — Windows 10 recommended.
- Linux® — Ubuntu® 18.04 recommended.
- Mac OS X

Supported Compilers to Build and Deploy DDS applications

You can build and deploy DDS applications by using a C++ compiler specific to your platform and additional vendor setup:

C++ Compilers

- Windows— Visual Studio 2019+
- Linux— GNU Compiler Collection (GCC) 6.3+
- Mac OS X — Xcode 10+

Additional Vendor Setup

RTI

DDS Blockset supports DDS out-of-the-box for the RTI products RTI Connext and RTI Connext Micro. You can generate code without additional setup for RTI, but to build and compile an executable, you must have RTI Connext 6.1.0+ or RTI Connext Micro 2.0.0+ installed.

To get started by using RTI within MATLAB, you can use the provided file exchange version of RTI Connext compatible with your version of DDS Blockset. To download and install, see RTI Connext® DDS connectivity framework for DDS Blockset - File Exchange .

For product information, see RTI Connext and RTI Connext Micro.

eProsima

DDS Blockset supports DDS out-of-the-box with the eProsima product Fast DDS version 2.6.0+. No additional setup is required. For product information, see Fast DDS.

See Also

“DDS Blockset Limitations and Considerations” on page 1-26 | RTI Connext® DDS connectivity framework for DDS Blockset - File Exchange | RTI Connext | Fast DDS

DDS Blockset Shapes Demo

This example shows how to import, configure, and deploy the Shapes Demo provided by DDS vendors to introduce DDS concepts. In the demo there are various shapes - Circles, Squares, and Triangles - that you can subscribe to watch their location as they move around a canvas. In this version of the demo, DDS Blockset provides an application that subscribes to a Square and then publishes the location of a Circle.

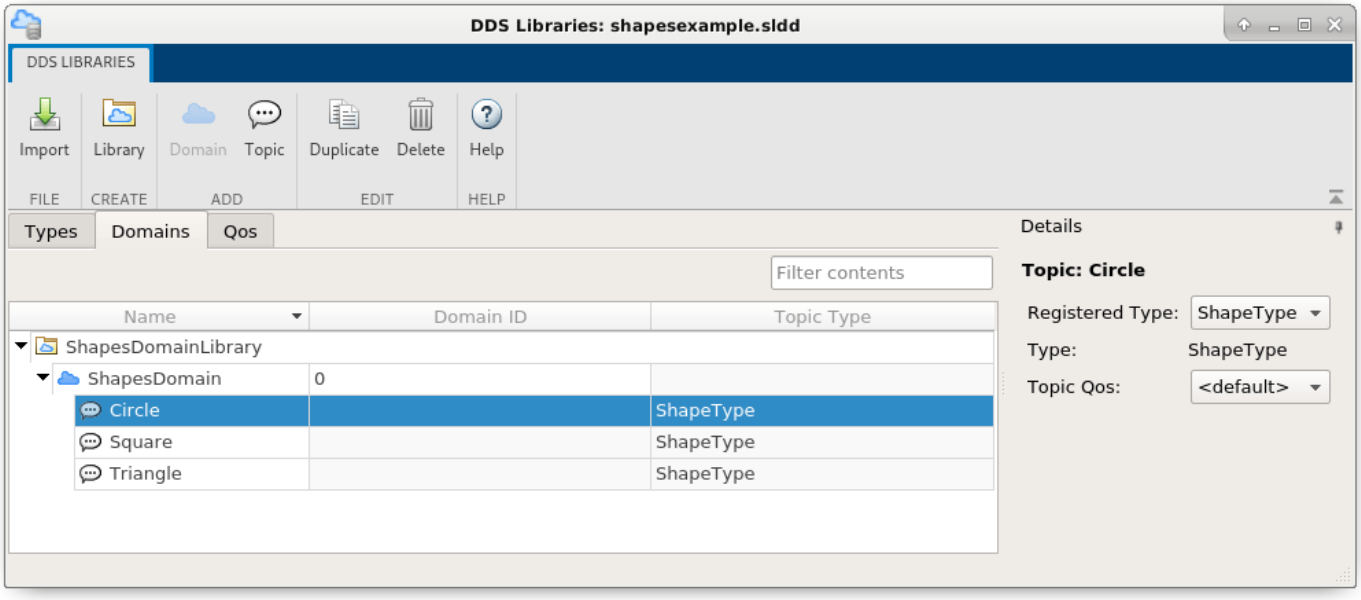
Import DDS Definitions

To get started with the Shapes Demo, the DDS definitions of Domains, Topics, Types, and Quality of Service (QoS) have been imported from XML. To view the XML specifications, open `shapesexample.xml`.

View and Edit Definitions

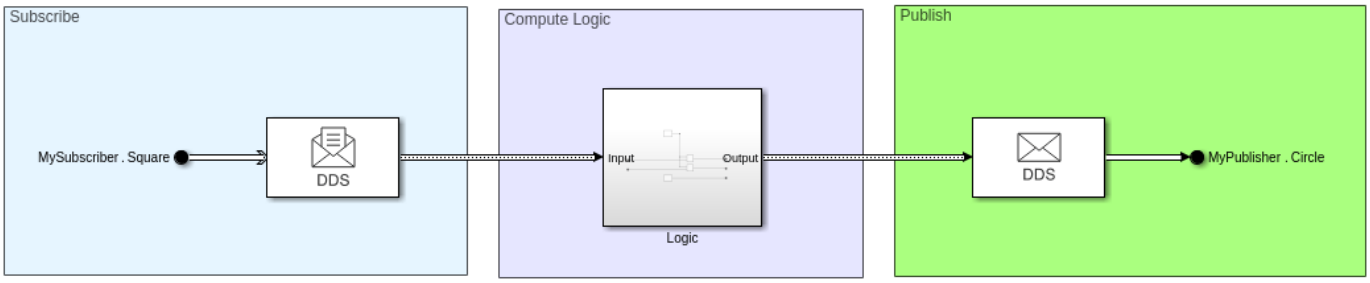
To view these DDS definitions in the DDS Dictionary, open the dictionary and view the **Types**, **Domains**, and **QoS** tabs.

```
open shapesexample.sldd
```



View Publisher and Subscriber Model Construction

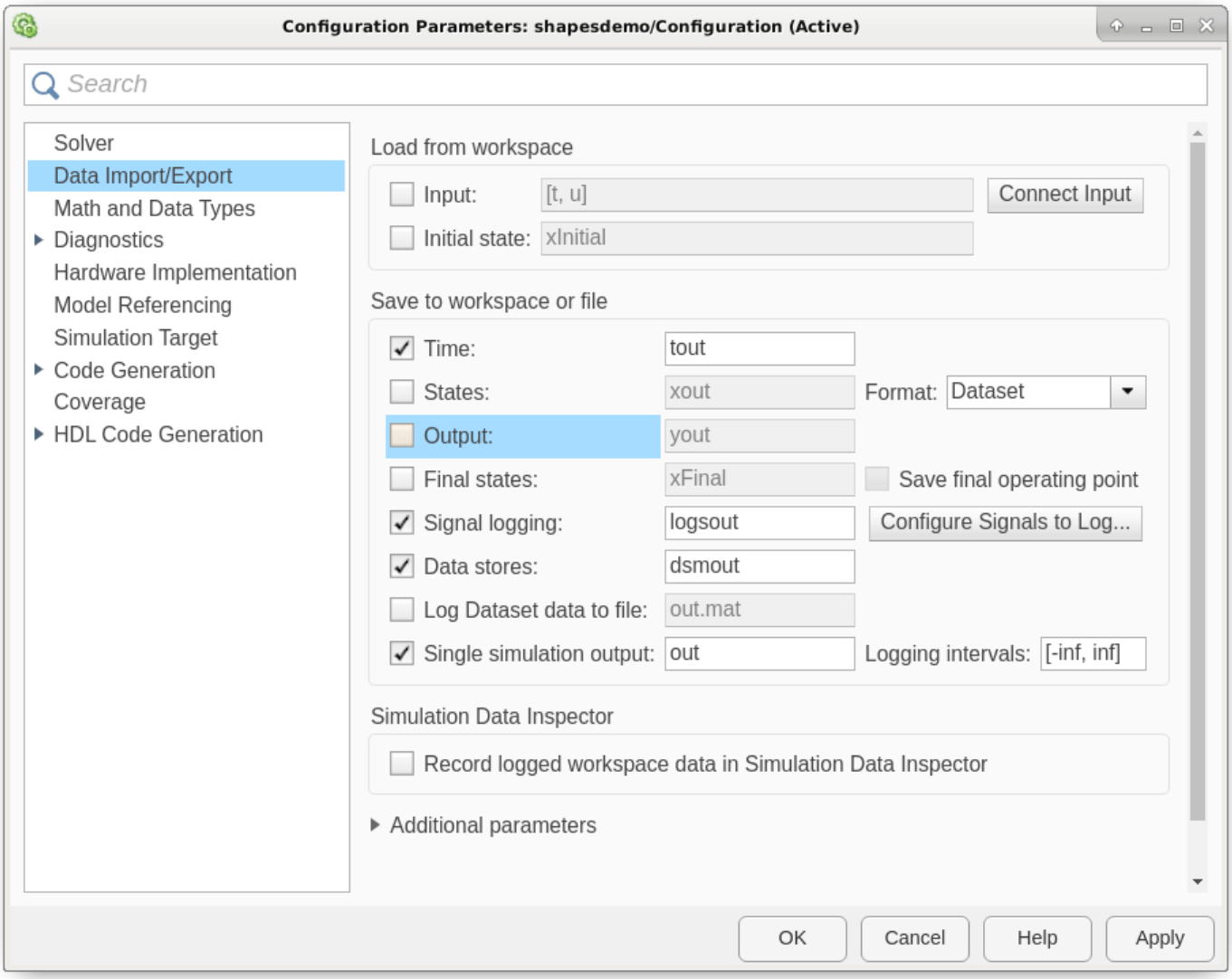
```
open_system('shapesdemo');
```



In the Simulink model for the Shapes Demo, view how the DDS blocks, Take DDS Sample and Write DDS Sample, are used to subscribe to and publish the location of the shape.

1. Click the Bus Element In block and view the DDS data type of the shape.
2. Click the Take DDS Sample block that converts the DDS data type to the Simulink data type.
3. Examine the logic component of the application model. The logic component uses the Simulink data types to compute the logic of the DDS application.
4. Click the Write DDS Sample block that converts the Simulink data types to a DDS data type.
5. Click the Bus Element Out block and view the DDS data type to send the shape position to the DDS network.

Also, note that root level output logging is unset for this model to use Bus Element Out block with virtual bus representing the DDS Publisher.



View Configured DDS Interface

In addition to the modeling of the Shapes Demo, you can view the imported DDS interface configuration of the Topics, DataReaders, and DataWriters for the model:

1. Open the Code Mappings editor. On the toolstrip, click **Code Interface** and select **Individual Elements Code Mappings**.
2. To view the imported Topics and DataReaders, click on the **Inports** tab.
3. To view the imported Topics and DataWriters, click on the **Outports** tab.

Build and Deploy the Shapes Demo

DDS Blockset enables out-of-the-box support for RTI and eProsima. To use this out-of-the-box support, build your application model and deploy your generated executable in your development

environment. For this example, click **Build** to subscribe to a Square and publish the location of a Circle Topic.

See Also

Related Examples

- “DDS Positioning System Application”

External Websites

- RTI Connext Shapes Demo
- eProsima Shapes Demo