DDS Blockset

Reference

MATLAB&SIMULINK®

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

Revision History

Contents

Blocks

Take DDS Sample

Receive Data Samples from DDS network
Library: DDS Blockset

Description

The DDS Blockset provides Take DDS Sample and Write DDS Sample blocks to help model DDS applications. The Take DDS Sample block receives samples from the DDS network and converts them to signals that can be used in the Simulink® environment. The Write DDS Sample block takes signals from the Simulink environment and converts them to samples to send data into the DDS network. These blocks are preconfigured to be compatible with the DDS network and provide type checking functionality.

Ports

Input

Port_1 — Input DDS sample
scalar | vector | matrix

Receives samples from the DDS network. Samples are converted to signals that can be used within the Simulink environment. The conversion preserves the value and data type of the original DDS sample.

Data Types: bus

Output

Port_S — Show whether a DDS sample was received
scalar

Shows whether a sample was received successfully from the DDS network. If a sample was received successfully the block outputs 1, otherwise it outputs 0.

Dependencies

To enable this port, select the check box labeled **Show receive status**.

Data Types: double

Port_1 — Output signal
scalar | vector | matrix

Converted signal that can be used in the Simulink environment.

Data Types: bus

Parameters

Show receive status — Show whether a DDS sample was received
off (default) | on

Select this check box to show whether a sample was received from the DDS network. If a sample was received successfully the block outputs 1, otherwise it outputs 0.

Programmatic Use
Block Parameter: ShowQueueStatus
Type: character vector
Values: 'on' | 'off'
Default: 'off'

Version History
Introduced in R2021a

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also
Write DDS Sample

Write DDS Sample

Send Data Samples to DDS network
Library: DDS Blockset

Description

The DDS Blockset Library provides Take DDS Sample and Write DDS Sample blocks to help model DDS applications. The Take DDS Sample block receives samples from the DDS network and converts them to signals that can be used in the Simulink environment. The Write DDS Sample block converts signals from the Simulink environment to samples to send data into the DDS network. These blocks are preconfigured to be compatible with the DDS network and provide type checking functionality.

Ports

Input

Port_Enable — External enable signal
scalar

Enables the block to send a sample into the DDS network. For input values greater than 0 send is enabled. For input values less than or equal to 0 send is disabled.

Dependencies

To display this port, select the check box labeled **Show enable port**.

Data Types: `double`

Port_1 — Input signal
scalar | vector | matrix

Signal in Simulink environment. Signals are converted to samples that applications use to send data into the DDS network. The conversion preserves the value and data type of the original signal.

Data Types: `bus`

Output

Port_1 — Output DDS sample
scalar | vector | matrix

Sends samples into the DDS network.

Data Types: `bus`

Parameters

Show enable port — Display the enable port
off (default) | on

Select this check box to display enable port.

Programmatic Use
Block Parameter: ShowEnablePort
Type: character vector
Values: 'on' | 'off'
Default: 'off'

Version History
Introduced in R2021a

Extended Capabilities

C/C++ Code Generation
Generate C and C++ code using Simulink® Coder™.

See Also
Take DDS Sample

Tools

DDS Dictionary

Manage DDS aspects of an application

Description

To manage the DDS aspects of your applications in the Simulink environment, you can use the DDS Dictionary, which is a section of the Simulink data dictionary that contains DDS properties. The DDS Dictionary is an intuitive graphical interface that enables you to quickly create and edit the DDS Domains, Topics, Data Samples, and Quality of Service (QoS).
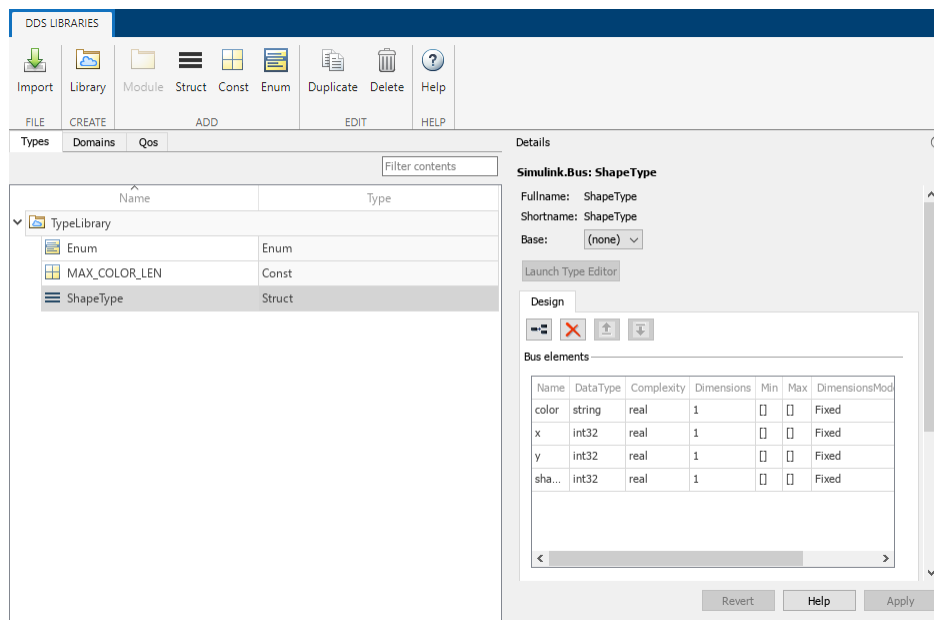
The DDS Dictionary is composed of three management tabs.

Types

The **Types** tab manages the DDS data types that are used to publish and subscribe to Data Samples in the DDS network. The tab can also be used to view and edit the Simulink equivalent data types that are used to compute the application logic in the Simulink environment.
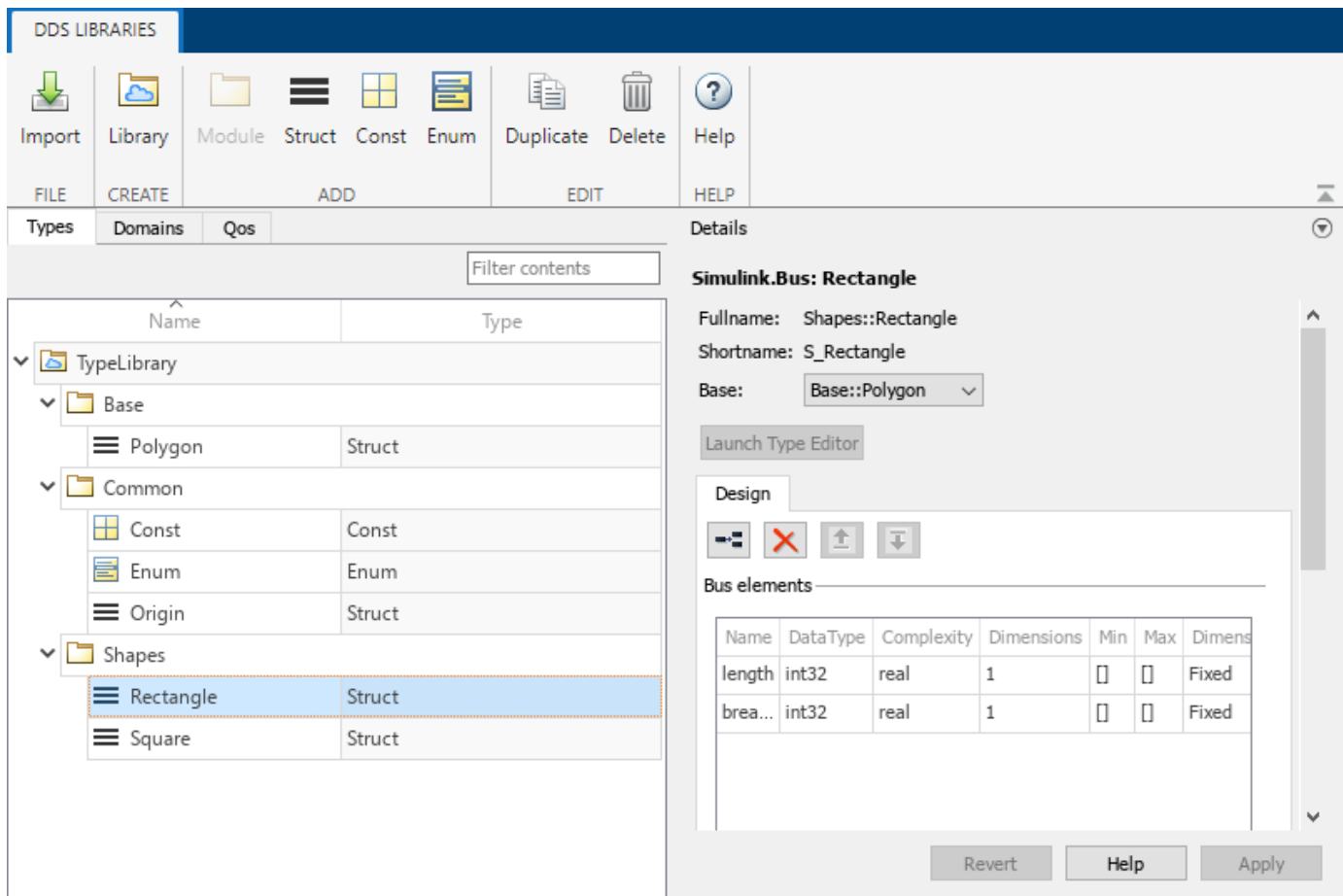
See "Type" on page 2-0   for more information, including details about full names and short names.

Domains

The **Domains** tab manages the Domains, Topics, and Registered Types so that applications can be configured to publish and subscribe to Topics.
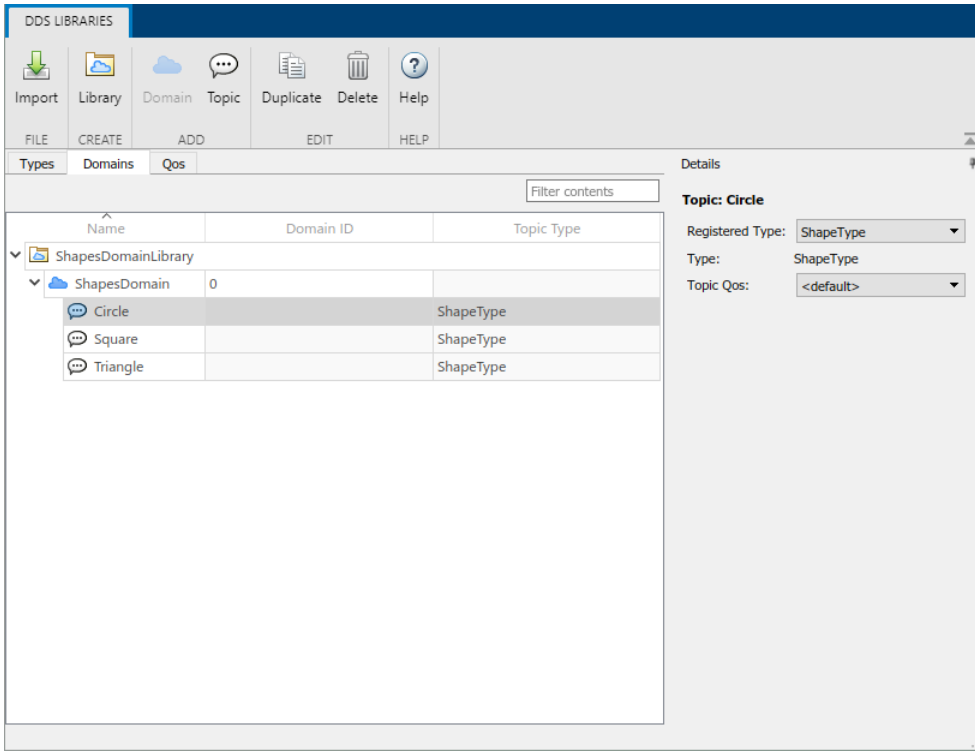
See "Topic Type" on page 2-0   for more information.

QoS

The **QoS** tab manages the Quality of Service (QoS) available for a DDS application.
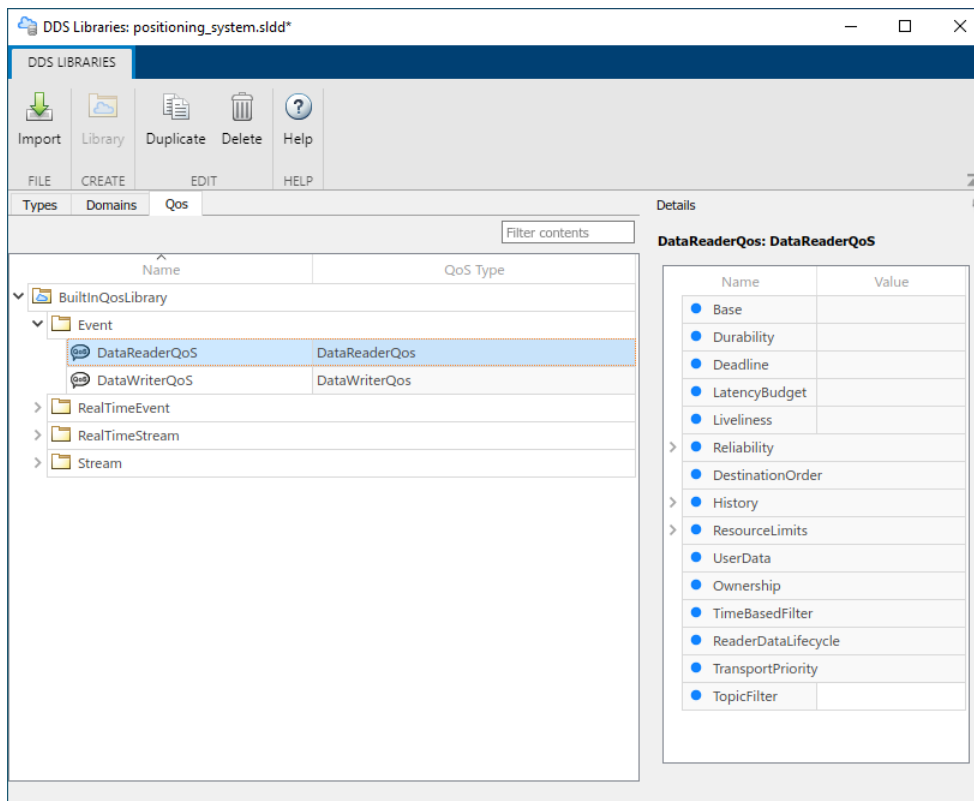
See "QoS Type" on page 2-0   for more information.

Open the DDS Dictionary

- Open a model in the DDS Blockset app. On the **DDS** tab, click **Code Interface** and select **DDS Dictionary**.
- In a Simulink data dictionary, if DDS definitions are available, a **DDS Libraries** node appears in the dictionary. If you open the **DDS Libraries** section of the Simulink data dictionary, in the **Details** pane, you can click **Open DDS Libraries** to open the graphical interface for the DDS Dictionary.

Examples

Import XML Files

This example shows how to import additional XML files to include more DDS definitions in your DDS Dictionary.

1. Open the DDS Dictionary.
2. Click **Import**.
3. Select additional XML files and click **Open**.

Configure Types

This example shows how to create and configure new DDS data types and their equivalent Simulink data types.

1 Open the DDS Dictionary.
2 Create a new DDS data type library. On the DDS Dictionary toolstrip, click **Library**.
3 Create DDS data types. Select the new library and create the following new DDS data types:

- Create a DDS Struct data type. On the DDS Dictionary toolstrip, click **Struct**. Select the new DDS type and in the **Details** pane, view its Simulink equivalent data type, a `Simulink.Bus` object.

- Create a DDS Const data type. On the DDS Dictionary toolstrip, click **Const**. Select the new DDS type and in the **Details** pane, view its Simulink equivalent data type, a numeric MATLAB® variable.

- Create a DDS Enum data type. On the DDS Dictionary toolstrip, click **Enum**. Select the new DDS type and in the **Details** pane, view its Simulink equivalent data type, a Simulink Enumerated data type.

Configure Domains and Topics

This example shows how to create and configure a new Domain and Topic.

1 Open the DDS Dictionary. Click the **Domain** tab.
2 Create a new Domain. On the DDS Dictionary toolstrip, click **Domain**.

- Edit the Domain name. For the new Domain, in the **Name** column, edit the name directly in the spreadsheet.

- Edit the Domain ID. For the new Domain, in the **Domain ID** column, edit the Domain ID directly in the spreadsheet.

3 Create a Registered Type. To create a new Topic you must first create a Registered Type for your new Domain. Select the new Domain, in the **Details** pane, click the link icon to create a Registered Type.
4 Create a new Topic. On the DDS Dictionary toolstrip, click **Topic**.

- Set the Registered Type for a Topic. Select the new Topic. In the **Details** pane, from the **Registered Type** drop-down, select a Registered Type from the listed options.

- Set the Quality of Service (QoS) for a Topic. Select the new Topic. In the **Details** pane, from the **Topic QoS** drop-down, select a QoS profile from listed options.

Configure Quality of Service (QoS)

This example shows how to import, view, and edit the Quality of Service (QoS) for your DDS application.

1 Open the DDS Dictionary.
2 Import QoS from XML.
3 In the DDS Dictionary, click the **QoS** tab and view QoS profiles and policies. In the **Details** pane, set QoS policy values as needed.

4   (Optional) Duplicate or delete QoS profiles in your dictionary. To duplicate a QoS profile, select a profile and click **Duplicate**. To delete a QoS profile, select a profile and click **Delete**

Parameters

Types

Name — Names of DDS data type libraries and data types
character vector

The names of DDS data type libraries or DDS data types available in the DDS Dictionary.

Example: TypeLibrary, ShapeType

Type — DDS data type
character vector

The DDS data type in the DDS Dictionary. Supported DDS data types are Const, Enum, and Struct.

To view the equivalent Simulink data type for a DDS data type, select a DDS data type to open the **Details** pane and review the corresponding Simulink data type information.

DDS Data Type	Simulink Data Type
Const	Numeric MATLAB variable
Enum	Enumeration
Struct	`Simulink.Bus` object

The **Details** pane displays the full name and short name of a DDS data type. `Fullname` is a concatenation of the full path of the nested module names defined in the namespace hierarchy for the data type. `Shortname` is a version of the long name that is shortened by using a unique identifier representing the namespace hierarchy.

For Struct types, the **Details** pane also displays `Base`, which is the base of the data type. If you specify a base for a Struct type, the Simulink bus object contains elements of the base type in addition to the elements defined for the data type. DDS Blockset checks for name conflicts and prevents the addition of elements with the same name in the hierarchy.

Type names are represented in Simulink and generated code as follows:

- For new DDS dictionaries created in R2022b, data type names are the short names.
- For DDS dictionaries created in R2022a or earlier, data type names are the full names.

For example, the data type name for a Simulink object in R2022a is represented as the full name, which is the module name appended to the instance name: `Shapes_Rectangle`, `Base_Polygon`, `Common_Origin`. In R2022b, these data types are represented as the short names, `S_Rectangle`, `B_Polygon`, `C_Origin`.

For DDS dictionaries created in R2022a or earlier and moved to R2022b, data type names are represented as the full names for backward compatibility.

Example: Const, Enum, Struct

Domains

Name — Names of Domain libraries, modules, Domains, and Topics
character vector

The names of Domain Libraries, Domain Modules, Domains, and Topics available in the DDS Dictionary.

Example: ShapesDomainLibrary, ShapesLibrary, Circle

Domain ID — ID of a Domain
character vector

Domain ID you can set to specify a Domain.

Example: 1

Topic Type — DDS data type specified for a Topic
character vector

The DDS data type that a Topic uses for its Data Samples.

Example: ShapeType

QoS

Name — Names of QoS libraries and profiles
character vector

The names of Quality of Service (QoS) libraries and profiles.

Example: BuiltInQosLibrary, Event, DataReaderQoS

QoS Type — Specifies the type of QoS profile
character vector

The QoS Type is a read-only field that shows if a QoS profile is specified for DataReaders or DataWriters.

Example: DataReaderQoS

Version History
Introduced in R2021a

Support for shortened version of DDS data type names

The DDS Dictionary now includes a shortened version of DDS data type names for use in Simulink and in the generated code.

Before R2022b, a data type name was represented in Simulink and generated code as the full name, which is a concatenation of the full path of module names defined in the namespace hierarchy for the type. This representation sometimes resulted in object names that exceeded the 63 character limit. The short name for a data type is a version of the long name that is shortened by using a unique identifier representing the namespace hierarchy.

- For new DDS dictionaries created in R2022b, the dictionary displays the full name, short name, and base of a DDS data type. Simulink and code generation use the short name for Simulink.Bus objects, enumerated types, and constants, with appropriate C++ aliases for the namespace and full names in the generated code.

- For DDS dictionaries from previous releases moved to R2022b, Simulink and code generation use the full name for backward compatibility.

See Also
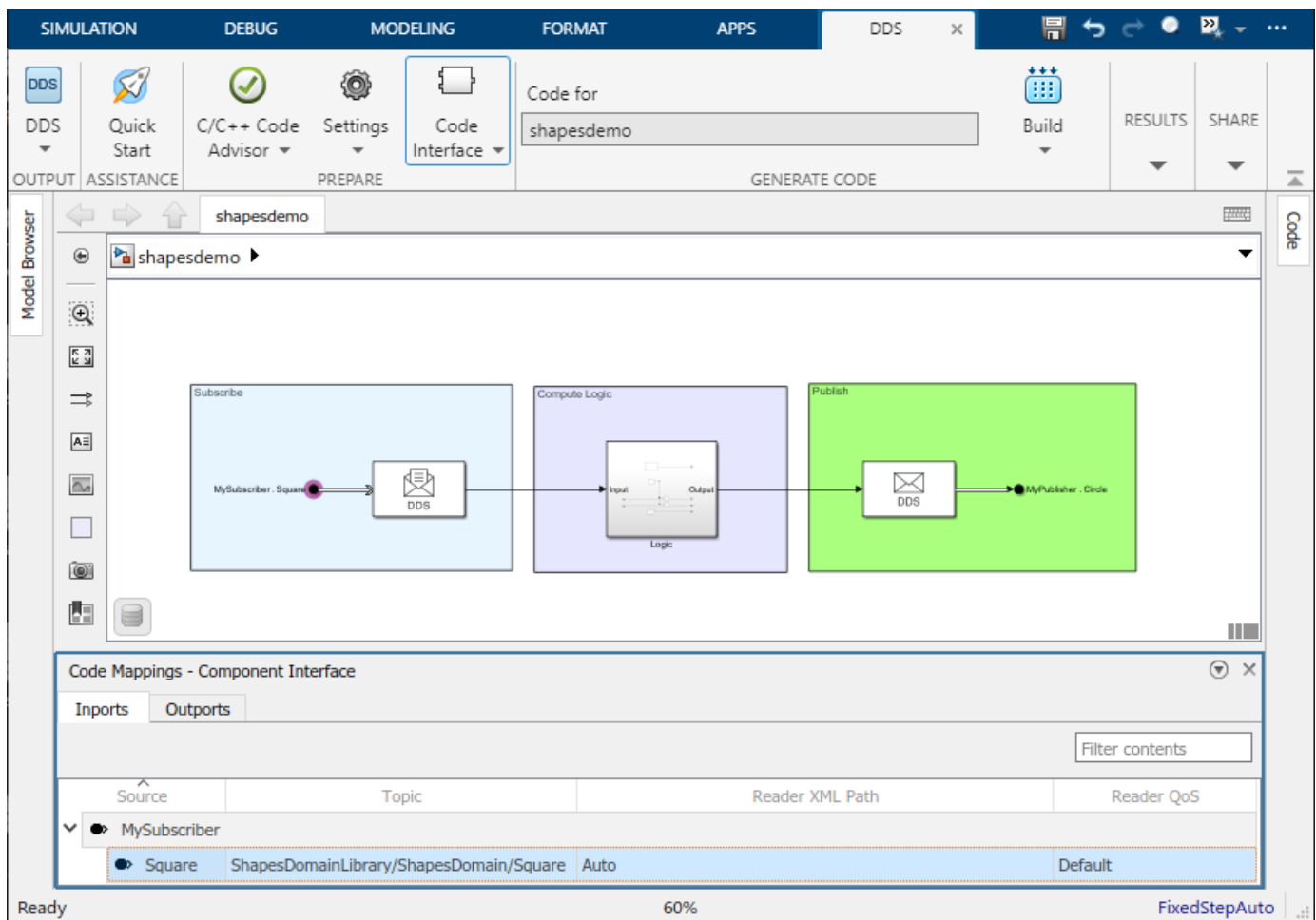**Code Mappings Editor**

Code Mappings Editor

Configure DDS application inports and outports as DataReaders and DataWriters

Description

The Code Mappings editor is a graphical interface where you can configure inports and outports as DataReaders and DataWriters in a DDS application.

To configure the inports and outports, use the Code Mappings editor panes:

- **Inports**
- **Outports**

Open the Code Mappings Editor

- Open a model in the DDS Blockset app. On the **DDS** tab, click **Code Interface** and select **Individual Element Code Mappings**.

- Open a model in the DDS Blockset app. The Code Mappings editor appears as a minimized tab directly below the model canvas. Click the tab to expand the editor.

Examples

Configure DataReaders and DataWriters by Using Default Behavior

This example uses the Shapes Demo example model to show how to configure the DataReaders and DataWriters for a DDS application by using default behavior.

Create a DDS application

1. Open the Shapes Demo example model, `shapesdemo`.
2. Create DDS definitions. In the DDS Application Quick Start, on the second page, select **Create and use default dictionary**.

Configure the DDS Interface

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

2. Configure the inports. The inports represent DataReaders in a DDS application.

- Select the Topic. Set the **Topic** to the Square Topic specified in the drop-down options by its path (ShapesDomainLibrary/ShapesDomain/Square).
- Select the DataReader. Set the **Reader XML Path** to Auto to have DDS Blockset automatically generate a reader for the application.
- Select the Quality of Service (QoS). Set the **Reader QoS** to Default to have the reader use the vendor provided QoS.

3 Configure the outports. The outports represent DataWriters in a DDS application.

- Select the Topic. Set the **Topic** to the Circle Topic specified in the drop-down options by its path (ShapesDomainLibrary/ShapesDomain/Circle).
- Select the DataWriter. Set the **Writer XML Path** to Auto to have DDS Blockset automatically generate a reader for the application.
- Select the Quality of Service (QoS). Set the **Writer QoS** to Default to have the reader use the vendor provided QoS.

Deploy the DDS application

1 Set up the build environment for your target vendor.
2 On the **DDS** tab, click **Build**.
3 Deploy the application executable on DDS.

### Configure DataReaders and DataWriters by Using XML Definitions

This example uses the Shapes Demo example model to show how to configure the DataReaders and DataWriters for a DDS application by using definitions previously specified in imported XML definitions.

Create a DDS application

1 Open the Shapes Demo example model, shapesdemo.
2 Create DDS definitions. In the DDS Application Quick Start, select **Import from XML**, and select the shapesexample_qos XML file. When you close the Quick Start, the DDS DataReader and DataWriter definitions from the shapesexample_qos XML file become available in the DDS Dictionary for the example application.

Configure the DDS Interface

1 Open the Code Mappings editor. On the **DDS** tab, click **Code Interface** and select **Individual Element Code Mappings**.
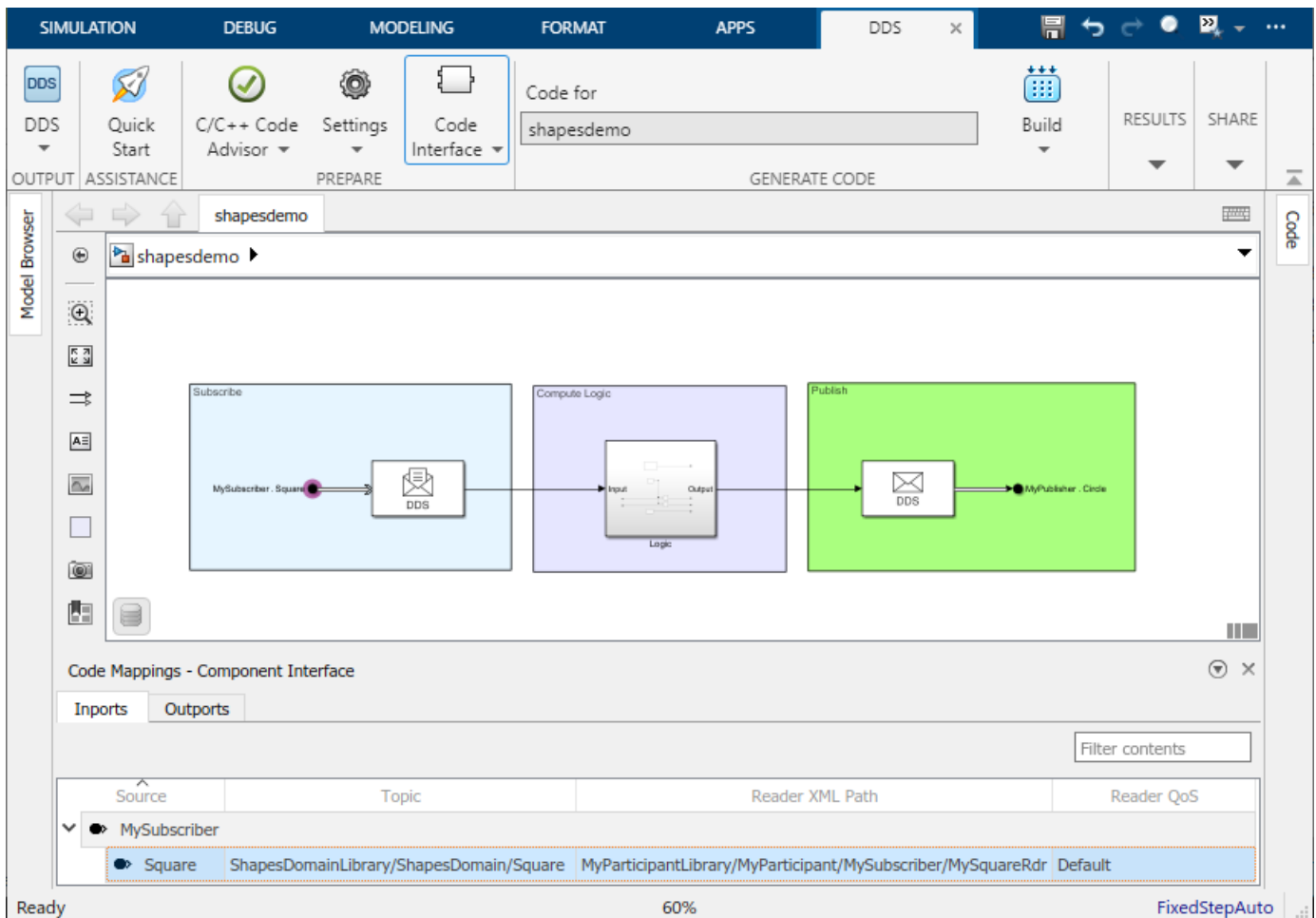
2   Configure the inports. The inports represent DataReaders in a DDS application. The Topic, DataReader, and QoS for each inport appears in the editor based on the imported XML definitions.

- Verify the Topic. For the **Topic**, verify that the Square Topic specified in the drop-down options by its path (`ShapesDomainLibrary/ShapesDomain/Circle`) is shown in the editor.

- Verify the DataReader. For the **Reader XML Path**, verify that the DataReader, `MySquareRdr`, specified by its path (`myParticipant/MySubscriber/MySquareRdr`).

- Configure the Quality of Service (QoS). Set the **Reader QoS** to `Default` to have the reader use the vendor provided QoS.

3   Configure the outports. The outports represent DataWriters in a DDS application. The Topic, DataWriter, and QoS for each outport appears in the editor based on the imported XML definitions.

- Verify the Topic. For the **Topic**, verify that the Circle Topic is specified in the drop-down options by its path (`ShapesDomainLibrary/ShapesDomain/Circle`).

- Verify the DataWriter. For the **Writer XML Path**, verify that the DataWriter, `MyCircleWtr`, specified by its path (`myParticipant/MySubscriber/MyCircleWtr`).

- Configure the Quality of Service (QoS). Set the **Writer QoS** to `Default` to have the writer use the vendor provided QoS.

Deploy the DDS application

1 Set up the build environment for your target vendor.
2 On the **DDS** tab, click **Build**.
3 Deploy the application executable on DDS.

Parameters

Inports

`Source` — Root-level Inport block or bus element
character vector

Identifies a root-level Inport block or an element of an In Bus Element block in the model. Inports map to DataReaders in DDS applications.

`Topic` — Specifies the Topic that a DDS application subscribes to
character vector

Select a Topic for each DataReader.

`Reader XML Path` — Specifies the DataReader
character vector

Select or generate a DataReader for each inport.

`Reader QoS` — Specifies the QoS for the DataReader
character vector

Select the QoS for the DataReader for each inport.

Outports

`Source` — Name of root Outport block or bus element
character vector

Identifies a root-level DDS Outport block or an element of an Out Bus Element block in the model. Outports map to DataWriters in DDS applications.

`Topic` — Specifies the Topic that a DDS application publishes
character vector

Select a Topic for each DataWriter.

`Writer XML Path` — Specifies the DataWriter
character vector

Select or generate a DataWriter for each outport.

`Writer QoS` — Specifies the QoS for the DataWriter
character vector

Select the QoS for the DataWriter for each outport.

Version History
Introduced in R2021a

See Also
DDS Dictionary

Topics
"DDS Blockset Shapes Demo"