

RTI System Designer

Getting Started Guide

Version 7.2.0



Contents

1	Introduction	1
1.1	Prerequisites	2
2	Using System Designer	3
2.1	Starting System Designer	3
2.2	Create a New Project	4
2.3	Workflow	5
2.4	Exploring the Views	6
2.4.1	Working with Types	6
2.4.2	Configuring Quality of Service (QoS)	11
2.4.3	Handling Domains	22
2.4.4	Define Participants and Entities	27
2.4.5	Define Plugins	31
2.4.6	Define Routing Services	34
2.4.7	Define User Environment variables	46
2.4.8	Using Fully Qualified Names when Referring to Objects	48
2.4.9	Configuring Security Plugins	49
2.4.10	EZ-Secure	52
2.5	Toolbar	53
2.6	Project Management	54
2.6.1	Create a New Project	54
2.6.2	Open an Existing Project	56
2.6.3	Open a Recent Project	58
2.6.4	Import a Project from a Zip File	60
2.6.5	Export a Project to a Zip File	63
2.6.6	Remove a Project from the List of Recent Projects	65
2.6.7	Validate Project	67
2.6.8	Project Properties	69
2.7	External XML Files	72
2.7.1	Creating a New File	73
2.7.2	Open an Existing File	74
2.7.3	Import a File	76
2.8	XML Validation	79
2.8.1	Code Generator	79
2.8.2	Other Methods	80
2.9	Online Help	81
2.10	Info Dialog	81

3	Tutorials	82
3.1	Defining Your Data Type	82
3.2	Defining an XML Application Creation Configuration File	85
3.3	Defining a Routing Service Configuration File	91
4	Release Notes	95
4.1	Supported Platforms and Browsers	95
4.2	Compatibility	95
4.3	What's New in 7.2.0	95
4.3.1	Specify the time period to check for Identity Certificate and CRL file changes	95
4.3.2	System Designer now accepts fully qualified names when searching types	96
4.3.3	New option disables type checking to allow using undefined types	96
4.3.4	"Add.." button is now available when a QoS Profile is selected	97
4.3.5	Last Modified Time project property no longer updates on open	97
4.3.6	QoS Profiles can now be edited directly from their panel	97
4.3.7	QoS Profiles defined in the open project display first when working with inheritance .	97
4.3.8	IDL files can now be imported	97
4.3.9	Disable @key option when a member type contains an optional member	98
4.3.10	Edit IDL files directly in System Designer	98
4.3.11	Display common profiles only when creating a QoS inheritance	98
4.3.12	Specify maximum number of removed participants to track	98
4.4	What's Fixed in 7.2.0	98
4.4.1	Negative values were not allowed when setting the minimum, maximum, and default member values	98
4.4.2	Invalid use of the Unix path separator in Windows	99
4.4.3	System Designer incorrectly displayed the <base_name> tag after removing a QoS Snippet	99
4.4.4	System Designer allowed inheritance from types defined below the current type . . .	99
4.5	Previous Releases	99
4.5.1	What's New in 7.1.0	99
	New Types View option enables adding, editing, and removing #include directives . .	99
	New Routing Service XML configuration views	100
	New field to configure RTPS PSK Protection	100
	New field to configure notification when a security certificate is about to expire	100
	Copy directive (@copy) now available for C# and Python	100
	Support for the OBSERVABILITY_COLLECTOR_SERVICE_QOS kind category .	100
	New fields to configure the Participant Factory QoS Monitoring Tag	101
	Third-party software upgrades	101
4.5.2	What's Fixed in 7.1.0	101
	System Designer did not select newly created files	101
	The Add more... command did not generate a new member name when adding an Enumerator or Struct member	101
	System Designer failed to create string and wstring constants	101
	System Designer did not open new projects in a new window	102
	System Designer did not create a new XML file when creating a new element	102
	System Designer failed to create a Typedef when using @default, @min or @max annotations	102
	System Designer failed to change the Directive Kind in an existing directive	102

	Fixes Related to Vulnerabilities	102
	Fixes Related to Generated Code	104
4.5.3	What's New in 7.0.0	104
	Support for the DISCOVERY logging category	104
	New field to configure Domain tag when defining Participant QoS	104
	Improvements to .rtisdproj file to be human-readable and contain more metadata	106
	New field to configure Partition QoS when defining a DomainParticipant	106
	QoS snippets can now be composed of other QoS snippets	107
	Automatic validation of XML files when opening a project or adding XML files to a project	107
	Improved usability of Representation QoS Policy editor	108
	Updated Shared Secret Algorithm field in Configure Security dialog	108
	New security properties in Global Security settings	109
	Third-party software upgrades	109
4.5.4	What's Fixed in 7.0.0	109
	Unsupported QoS elements and policies appeared in Configuration	109
	Unconfigurable QoS XML elements and policies appeared in Configuration	110
	Renamed items were not updated in System Designer's UI	111
	Incomplete error messages when importing an XML file in some web browsers	111
	Deleting project files from disk may have left some empty directories	111
	System Designer incorrectly allowed XML and IDL to be edited in XML and IDL views	111
	Discrepancy between range defined by System Designer and by API for certain elements	111
	System Designer unable to edit Domain after detecting invalid Domain ID	112
	IDL and XML text editors appeared empty on network-isolated system	112
	System Designer failed to import multiple XML files	112
	Unable to add default case to Union member when adding more than one case value	114
	Types tab did not show #include directives from XML file	114
	Unable to move an element up or down in Types tab, in some cases	114
	Failure in Types tab when trying to add module with same name as another	115
	Could not create new file in Types tab if project contained read-only file	115
	Fixes related to vulnerabilities	116
4.6	Known Issues	116
4.6.1	Modules defined in multiple files, with different annotations	116

5 Copyrights and Notices

117

Chapter 1

Introduction

Note: This document assumes you have a basic understanding of *RTI® Connex® DDS* application development and concepts, such as a DDS Domains, *DomainParticipants*, *Topics*, *DataWriters*, and *DataReaders*. For an overview of these concepts, please see the [RTI Connex DDS Getting Started Guide](#).

RTI System Designer is based on RTI XML-Based Application Creation, a mechanism to simplify the development of *Connex DDS* applications, and on *RTI Routing Service*, an out-of-the-box solution that allows developers to rapidly scale and integrate real-time systems that are disparate or geographically dispersed. For an overview of their functionality, please see the following:

- [RTI Connex DDS Core Libraries XML-Based Application Creation Getting Started Guide](#)
- [RTI Routing Service User's Manual](#)

System Designer is a tool that allows you to graphically design and configure all the aspects of a *Connex DDS* distributed system. Think of it as a user interface (UI) to XML-Based Application Creation, a technology that allows you to specify in XML format all the aspects of a DDS system:

- The Data Types that will be used to communicate information in the system
- The *Topics* that will be used in the domain, associating each *Topic* with a *Data Type*
- The *DomainParticipant* that can potentially be used, giving each a participant-name
- The *DataWriters* and *DataReaders* within each *DomainParticipant*, each associated with its corresponding *Topic*
- The Qualities of Service (QoS) that all the entities can use

Your application code simply indicates the participant configuration name of the *DomainParticipant* that it wants to create. The XML-Based Application Creation infrastructure takes care of the rest: creating the *DomainParticipant*, registering the types and *Topics*, and populating all the configured Entities.

There is also a user interface to *Routing Service*, allowing you to configure it in XML.

Writing this XML can be quite challenging. Using a good editor with autocomplete capabilities along with the provided XML Schema can be of great help. But it is still quite complicated, especially as the project gets

bigger and involves multiple files. The *System Designer* UI tool can help ease this process and manage the complexity of a large system.

This *Getting Started Guide* is not a manual for *System Designer* and it does not cover all the functionality. The

tool has a lot of inline help. Look for the  symbol for more help while using the tool.

Tip: You can access a live version of *System Designer* without installing it by visiting: <https://community.rti.com/downloads/experimental/system-designer>

This URL allows you to test a demo version of *System Designer* and learn more before you download and install it.

1.1 Prerequisites

If you want to validate the XML as described in *Validate Project*, **xmllint** must be installed before running *System Designer*. **xmllint** is available for the major operating systems here: <http://xmlsoft.org/>.

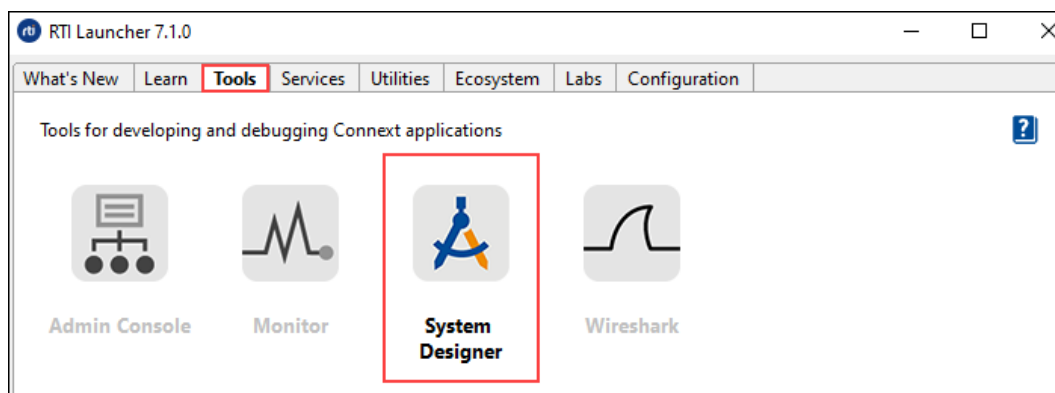
Chapter 2

Using System Designer

2.1 Starting System Designer

You can run *System Designer* from *Launcher* or from the distributed command-line executable.

In *Launcher*, go to the **Tools** tab and click the *System Designer* button:



You can also start *System Designer* with the `rtisystemdesigner[.bat]` script in the `bin` directory of your *Connex DDS* installation. The *System Designer* script will start the tool, try to open your default browser, and point to the right address.

If the browser won't start, open a new webpage and go to:

<http://localhost:5000/index.html>

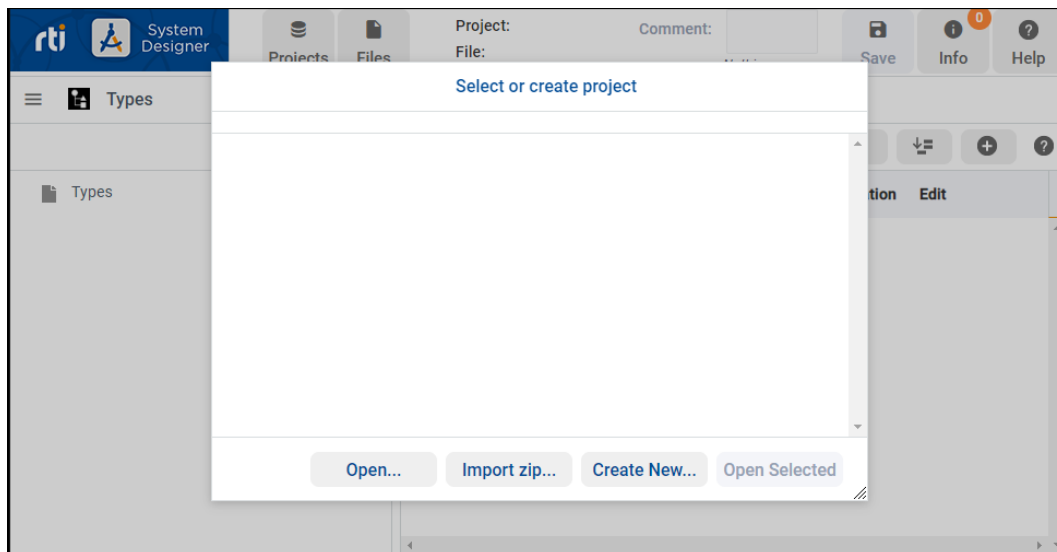
Tip: By default, *System Designer* uses port 5000. If that port is not available, you can change it by setting `RTI_SD_PORT` to be a valid port for your system.

Note: *System Designer* automatically shuts down the server if there hasn't been a connection in the last five minutes. This is important when running *System Designer* from *Launcher*, since clicking the *System Designer* button again won't create a new server instance until the five-minute period has passed. (Clicking the *System*

Designer button again while the previous instance is running will just open your default browser.) To open a new *System Designer* instance from *Launcher*, wait at least five minutes or kill the running instance.

When you start *System Designer*, the application will allow you to:

- **Open...** Browse the filesystem for an existing *System Designer* project (.rtisproj)
- **Import Zip...** Browse the filesystem to import a *System Designer* project that was previously exported and sent to you (.zip)
- **Create New...** Create a new project
- **Open Selected...** Open one of the listed Recent Projects (the list is empty right after the first execution)



2.2 Create a New Project

After you select **Create New...**, you will see this dialog:

You can choose a name and select the Project Path by clicking



. Press **OK**, and *System Designer* will create a new project and present you with the main UI where you can start designing your system.

When you create a new project, *System Designer* creates a project (`.rtisdproj`) file and if you checked the option when creating the project, an empty XML file. You can use *System Designer* to add components to the file.

You can work on your project wherever you want. *System Designer* finds default profiles and files in your *Connex DDS* installation for you.

2.3 Workflow

The *System Designer* main UI offers seven views, which represent the tasks in XML-Based Application Creation and Routing Service:

- Types Definition
- QoS Specification
- Domain Definition
- Participant Definition
- Plugins Definition
- Routing Service Definition

- User Environment Variables Definition

We suggest you start by defining a Type and proceed with the definition of the QoS. Then move to the Domain view, register your types, and declare your Topic. Finally, use the Participant view to define your participant, *DataReaders*, and *DataWriters*.

Of course, you can always (and probably will) go back and forth among these views to tweak your system.

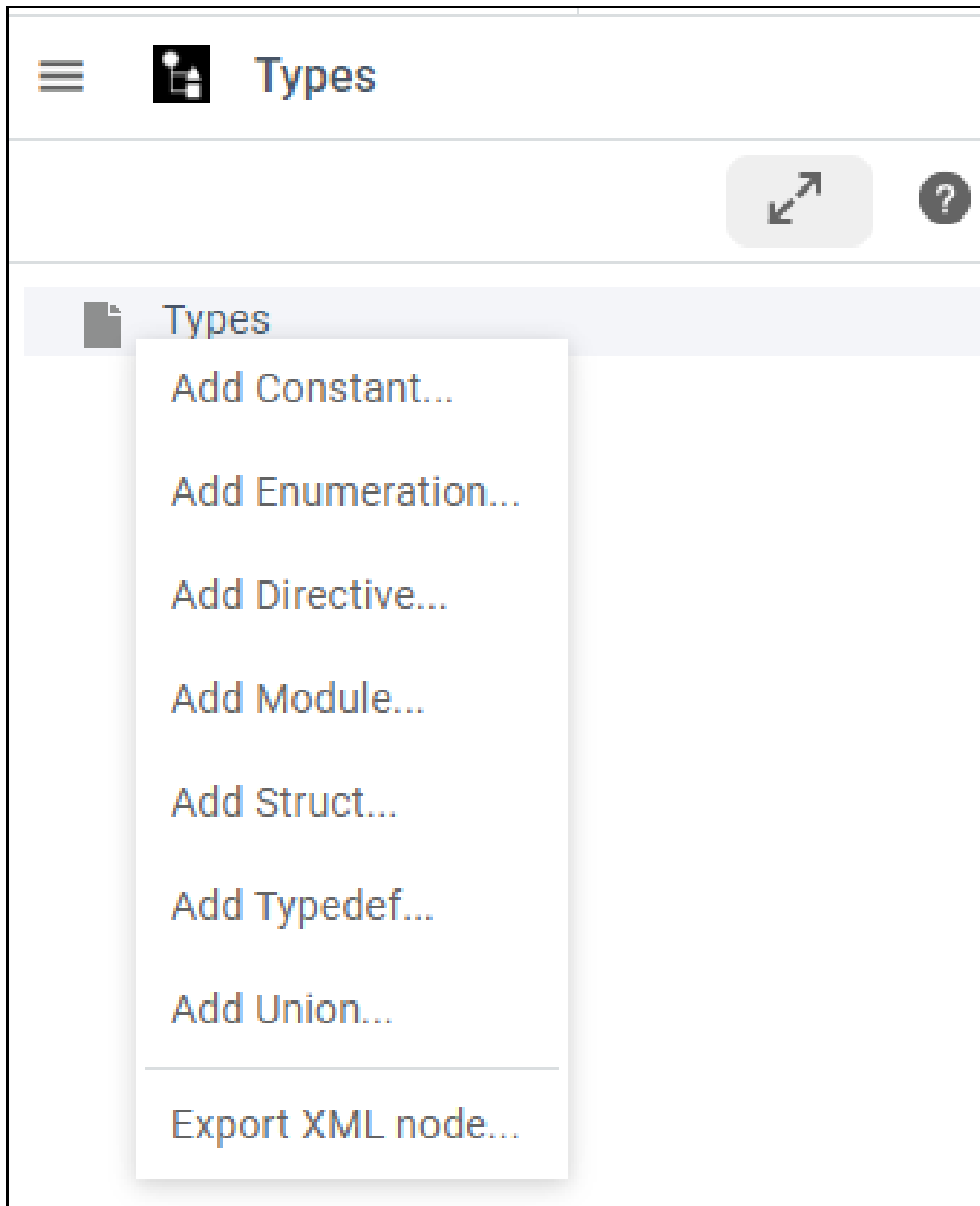
If you are defining a *Routing Service* configuration file, also move to the Plugins view, register the plugins, and then declare your *Routing Service* configuration in the Routing Services view.

System Designer automatically saves your projects and changes. (You can disable Autosave from the Projects menu: select **Properties**.)

2.4 Exploring the Views

2.4.1 Working with Types

To introduce new types in your project, right-click on the **Types** node in the Tree View and add one of the top-level types:



For more information on data types, see [Introduction to the Type System, in the RTI Connext DDS Core Libraries User's Manual](#).

In this example, we are adding two structures, NewStruct1 and NewStruct2:

Add Struct

Name:

Base Type:

Press [Key Down] or start typing for suggestions

☒ @resolve-name

☐ @nested

☐ @extensibility

☐ @autoid

☐ @transfer_mode

☐ @language_binding

☐ @allowed_data_representation

File:


Struct Declaration

Structs are aggregated type that contains named members, each with a corresponding member type. Structs can be used directly as the type published on a Topic or become types nested into another structured or container type. Structs can inherit from other structs.

- Base type can be only another struct or a typedef that resolves to a valuetype or struct.
- @resolve-name** tells rtdsgen to attempt to resolve the types related to this struct (on by default). When checked, if a particular type cannot be resolved during code generation rtdsgen will report an error. If unchecked, generation will proceed (but you might get a compilation error). This annotation is normally used on architectures where there is no preprocessor that can resolve #includes. Unchecking it at this level, affects all the types of the members as well as the base type.
- The annotation **@nested** tells rtdsgen that this struct cannot be used to represent top-level data that will be published or subscribed. If checked the code generated won't have the requested type support functionality required by types that can be published on a Topic. Check this only if this struct is always used nested into another container.
- @extensibility** lets you define the extensible type annotation to use for the given struct. Extensibility can be specified only if the struct does not have a base class.

The pop-up shown above will guide you in creating a new type. You can choose the Name and, optionally, the Base Type. You can also select one of the annotations presented on the screen.

By default, the new object will be added to the last XML file used, but you can use the drop-down menu to select any XML file or use the **New...** button to add new XML files to your *System Designer* project.

Once you have added the top level, go to the Structured view and populate the new type using the  button:



A pop-up will help you select the types and the right values. In this example, we are adding:

- A uint8 named 'member1' and a bounded (255) keyed string named 'member2' to the NewStruct1 structure.
- An octet named 'member1' and a char8 named 'member2' to the NewStruct2 structure.

Add Struct Member

Name:

Type:

String Maximum:

☐ Sequence

☐ Array dimension

☐ @key

☐ @optional

☐ @external

☒ @resolve-name

☐ @hashid

☐ @id

☐ @default

☐ @min

☐ @max

Member Declaration


Here you define an individual member of your struct.


- **@key** lets you specify that this member is part of a key used in your type.
- **@external** tells the middleware that the target data should be stored, not inline within its containing type, but external to it (i.e. using a pointer).
- **@resolve-name** tells rtiddsgen to attempt to resolve the type of this member (on by default). When checked, if a particular type cannot be resolved during code generation rtiddsgen will report an error. If unchecked, generation will proceed (but you might get a compilation error). It is normally used on architectures where there is no preprocessor that can resolve #includes. Applies only if non-primitive types are used for this member.
- **@optional** lets you to set the value NULL when writing to avoid sending any value for this field.
- **@hashid** lets you manually specify the hash ID that will be used to identify this member.
- **@id** is optionally used to specify the ordering of this member relative to the other members of the struct.
- **@visibility** lets you specify whether this member is private or public (valuetypes only).

The **Types** view allows you to explore the types in three ways:


- Structured view
- XML view
- IDL view

Tip: The [RTI Connex DDS Core Libraries Extensible Types Guide](#) has a mapping of type names in IDL 4.2. (For example, you may want to know that a `long` is called `int32` in IDL 4.2).

The Structured view is the simplest way to explore and modify the types in your system. The  button allows you to go as deep as needed when exploring types.

- In the **Item** column, you will see the name and type of a specific element of your type.
- In the **Annotations** column, you can see things like the ID of a member; you can see if the member is a key, or if it is an optional or extensible member.
- The **Location** column will indicate in which file the member is defined, in case you have a project with multiple XML files.
- The **Edit** column contains buttons to alter the selected item. Container objects such as structs, unions, and modules have a  button to add a child to them.



You can use  to rearrange all the objects.

Currently you cannot move elements in or out of a container (e.g., moving a struct out of a module is not currently supported). To do that, you can always switch to the XML view and modify the XML manually.

Structured XML IDL			
<input type="text"/> <input type="button" value="Search"/> <input type="button" value="↕"/> <input type="button" value="↕"/> <input type="button" value="↕"/> <input type="button" value="+"/> <input type="button" value="?"/>			
Item	Annotations	Location	Edit
▼ struct NewStruct1		ShapesDemo.x	
uint8 member1		ShapesDemo.x	
string<255> member2	@key	ShapesDemo.x	

The XML view shows the types in XML format. You can highlight a specific member by selecting it in the tree on the left panel:

Types

- Types
 - struct NewStruct1
 - struct NewStruct2

Structured XML IDL


XML File: C:\Users\loadholt\...

```

1 <types>
2   <struct name="NewStruct1">
3     <member name="member1" type="uint8"/>
4     <member name="member2" type="string" stringMaxLength="255" key="true"/>
5   </struct>
6   <struct name="NewStruct2">
7     <member name="member1" type="octet"/>
8     <member name="member2" type="char8"/>
9   </struct>
10 </types>
11

```

 **Edit XML...**

You can also click on  **Edit XML...** to edit your XML file manually. All the changes will be reflected in the Structured view.

Finally, the IDL view shows you the IDL definition of the type:

Types

- Types
 - struct NewStruct1
 - struct NewStruct2

Structured XML IDL

IDL File: NewProject.idl


```

1 struct NewStruct1 {
2   uint8 member1;
3   @key
4   string<255> member2;
5 };
6
7 struct NewStruct2 {
8   octet member1;
9   char member2;
10 };
11

```


You can use the IDL view to:

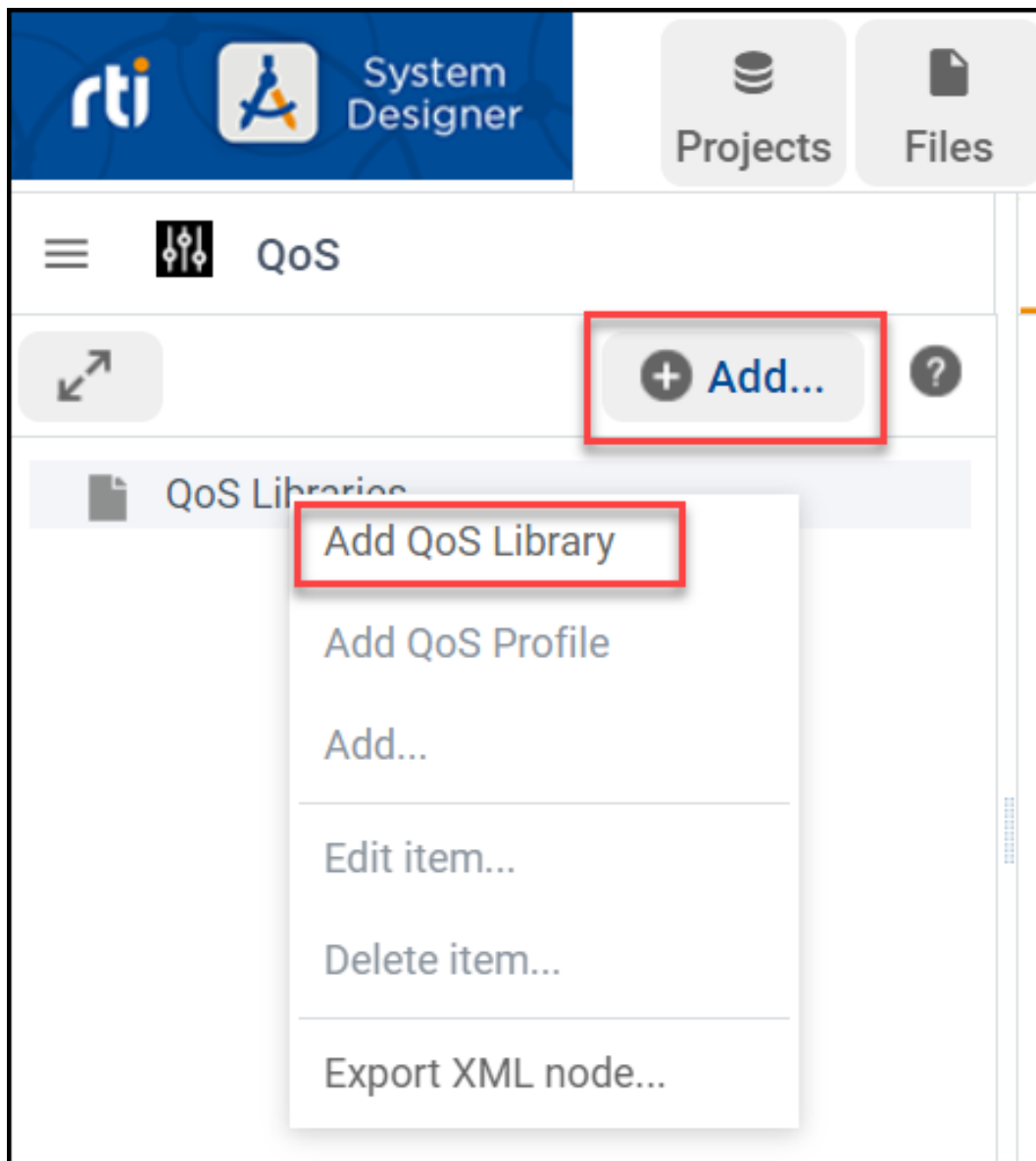
- Edit the IDL file manually by clicking the  **Edit IDL...**. All the changes will be reflected in the Structured view.

- Download the IDL type using the  **Download IDL...** button. (After you click this button, a dialog box opens; click the “Download” link at the bottom.)

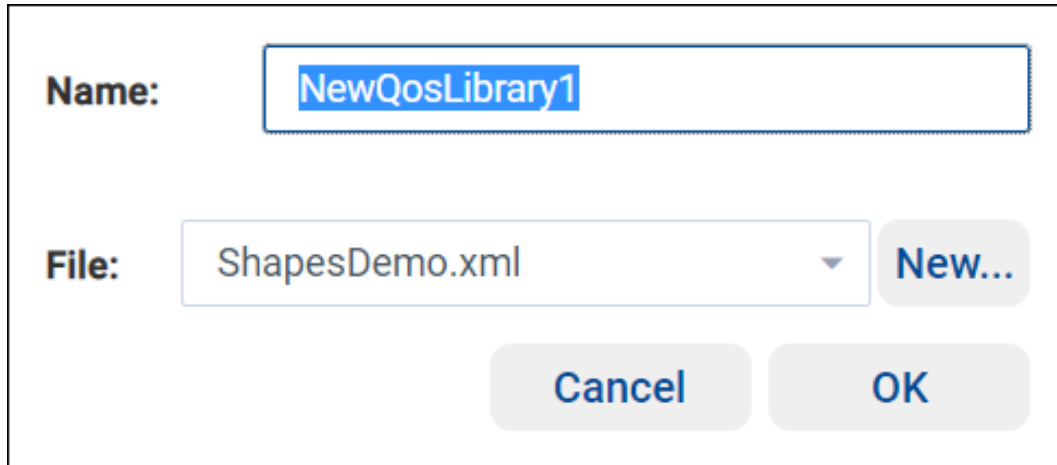
2.4.2 Configuring Quality of Service (QoS)

The QoS view allows you to add Qualities of Service and organize them in QoS Libraries and QoS Profiles.

To add new QoS to your model, click  **Add...** in the QoS view or right-click **QoS Libraries** in the tree:

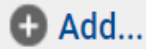


To add a QoS Library, enter the Library Name and the destination file:

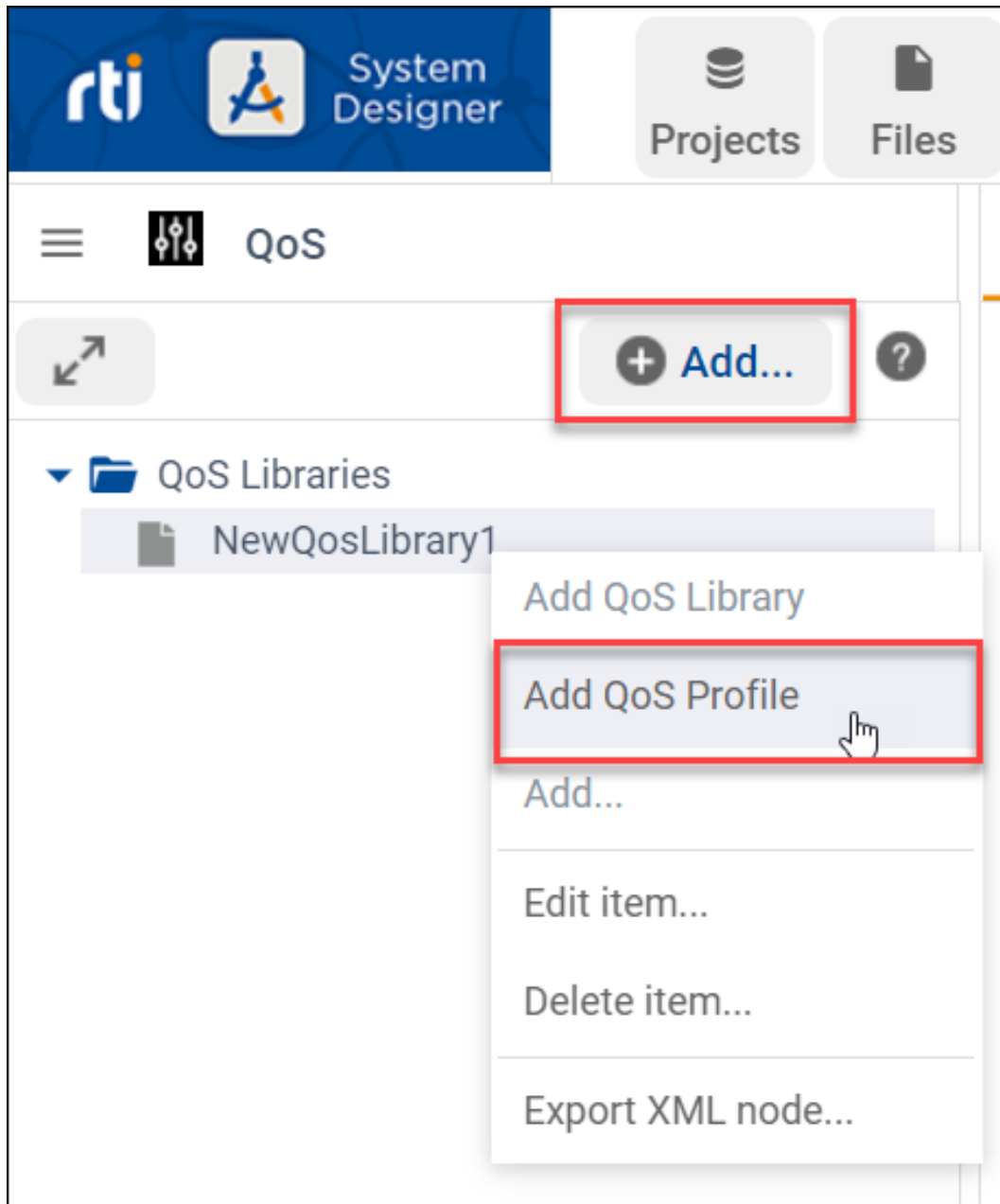


The dialog box has a title bar. Inside, there are two main sections. The first section is labeled 'Name:' and contains a text input field with the text 'NewQosLibrary1'. The second section is labeled 'File:' and contains a dropdown menu with 'ShapesDemo.xml' and a 'New...' button. At the bottom of the dialog are two buttons: 'Cancel' and 'OK'.

You can add QoS Profiles to that library by using




in the main panel while a QoS Library is selected or by right-clicking on the QoS Library:





A pop-up window guides you to select the profile name, the destination file, the profile you want to inherit from, and other attributes as shown in the screenshot below:


Name:


☐ Profile is a QoS Snippet 



Inheritance Chain:



☐ Default Profile 

☐ Default Participant Factory Profile 

☐ Single Entity QoS Profile 

File:  

C:\Users\loadholt

To inherit values from another QoS Profile or a QoS Snippet, click  next to **Inheritance Chain**.

The screenshot shows a dialog box for configuring a profile. At the top, there is a checkbox labeled "Show Common Profiles". Below this is a section labeled "Base Profile:" followed by a dropdown menu. Underneath is a section labeled "QoS Snippets:" with a help icon (a question mark in a circle) to its right. The "QoS Snippets:" section contains a list area with the text "double-click to add a new item". At the bottom right of the dialog are two buttons: "Cancel" and "OK".

To inherit values from another profile, select the appropriate profile in the **Base Profile** field. To narrow the list of available profiles to those most commonly used, select **Show Common Profiles**. These common profiles include:

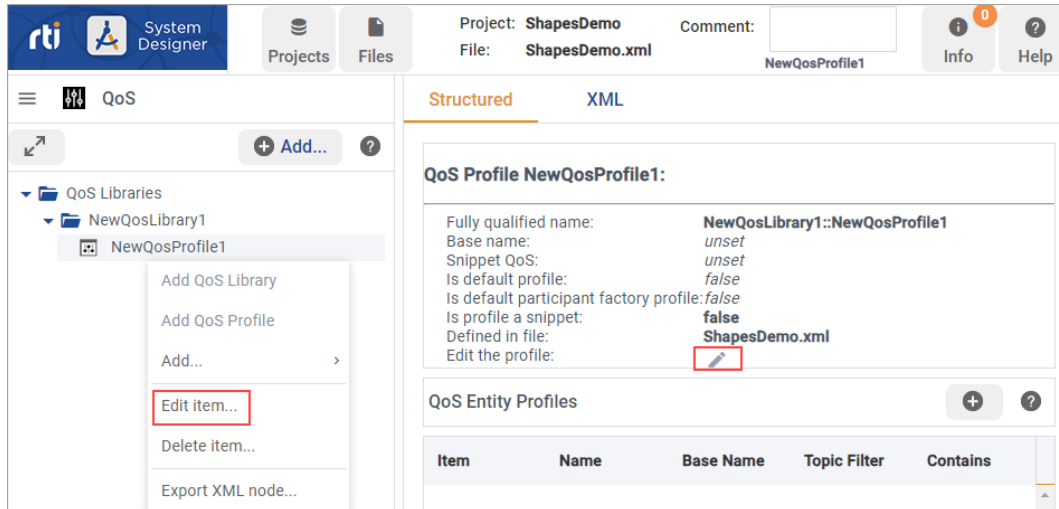
- Builtin profiles starting with BuiltinQosLib::Pattern
- QoS Profiles defined in the open project

To inherit values from a Snippet, double-click in the **QoS Snippets** field to select one or more Snippets. Note that some profiles cannot be defined as Snippets because they inherit from a base profile.

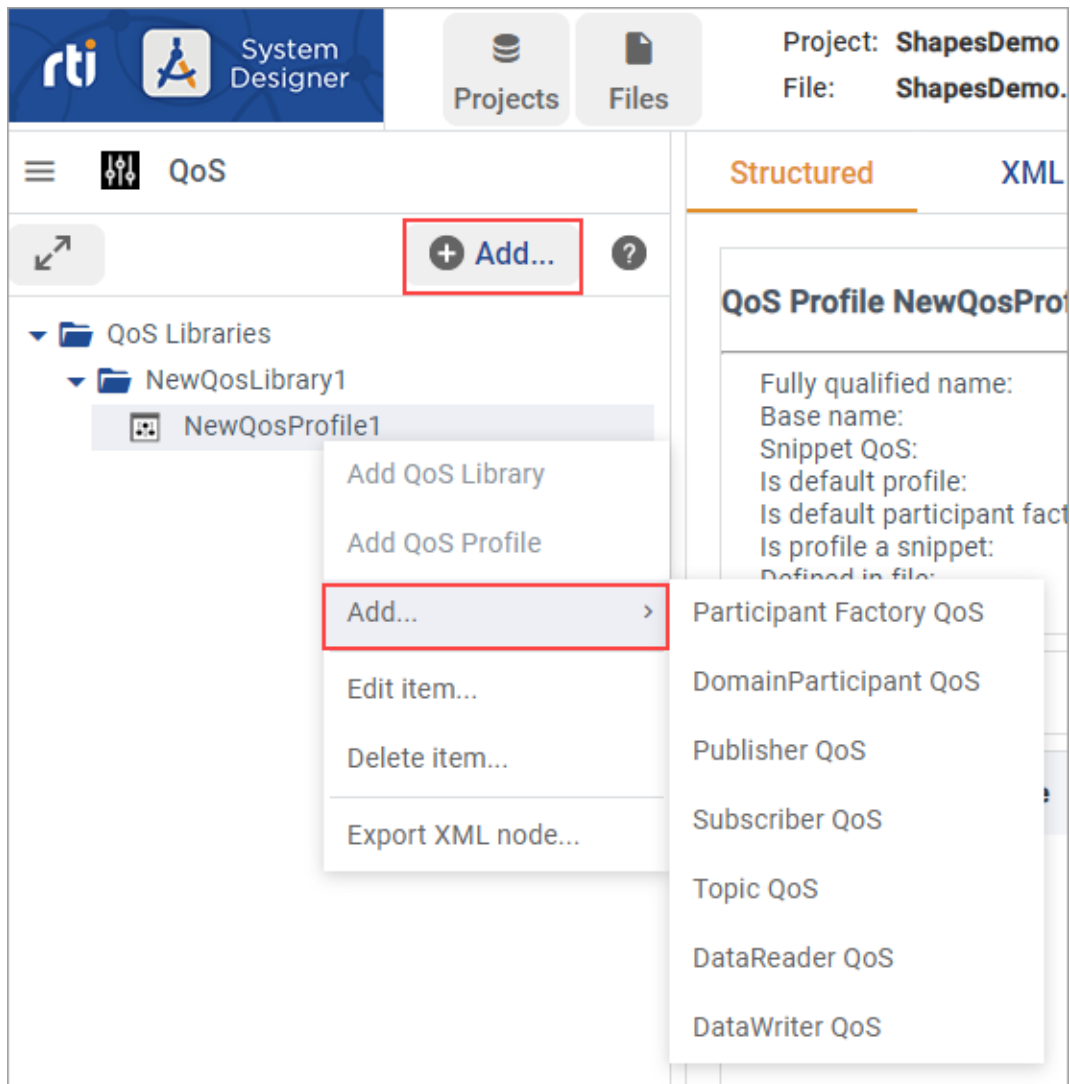
For more information on Entities and Profiles, see these sections of the *RTI Connex DDS Core Libraries User's Manual*:

- [QoS Profile Inheritance and Composition](#)
- [DDS Entities](#)
- [Fundamentals of DDS Domains and DomainParticipants](#)
- [DomainParticipantFactory](#)

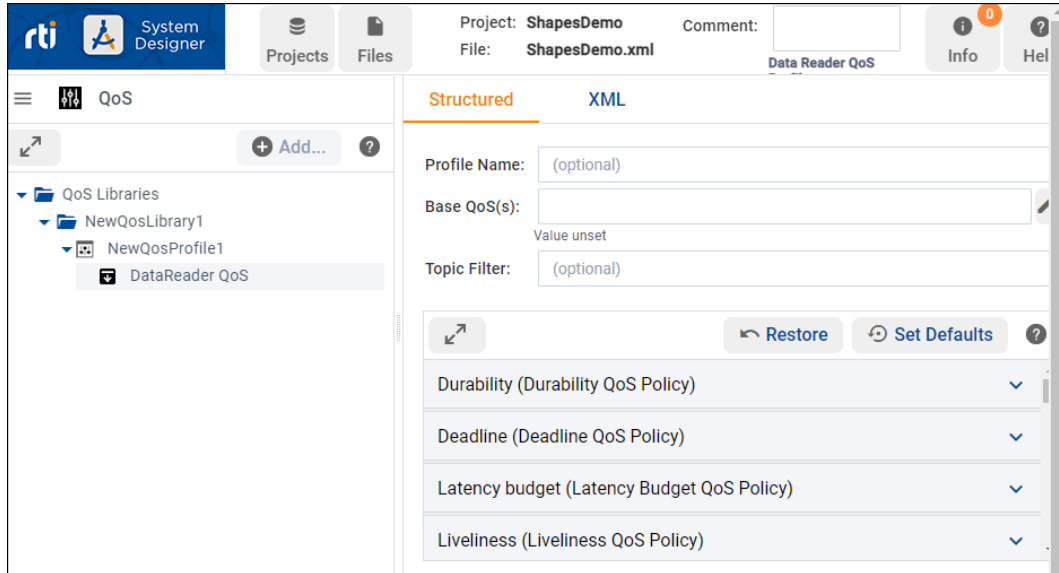
After creating a profile, you can edit it by right-clicking the profile in the tree or clicking the **Edit the profile** command in the QoS Profile view:



Once you have your profiles configured, you can select one of them and add QoS for a specific Entity:

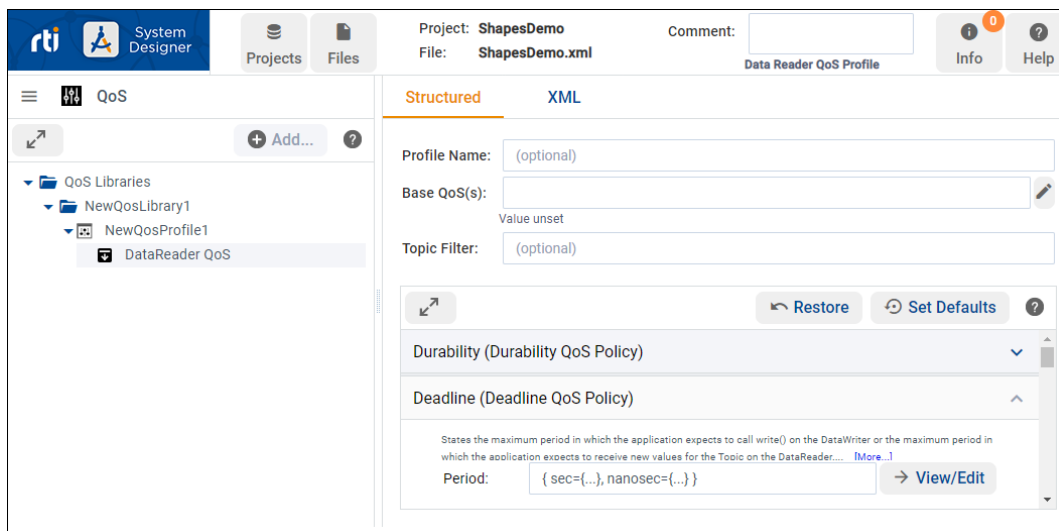


When an Entity is selected, you can set all the QoS related to that Entity in the Structured view:



See [QoS Policies](#) in the *RTI Connext DDS Core Libraries User's Manual* for more information on the QoS available for each entity.

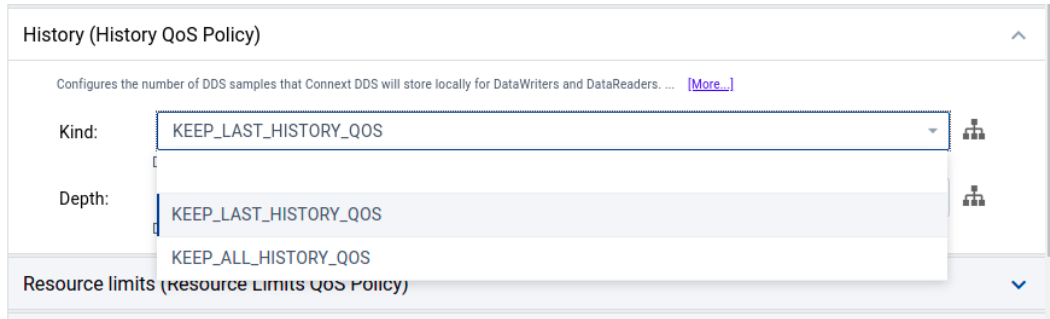
You can browse all of the QoS policies and select the  button to set and get more information on a specific setting within a policy:



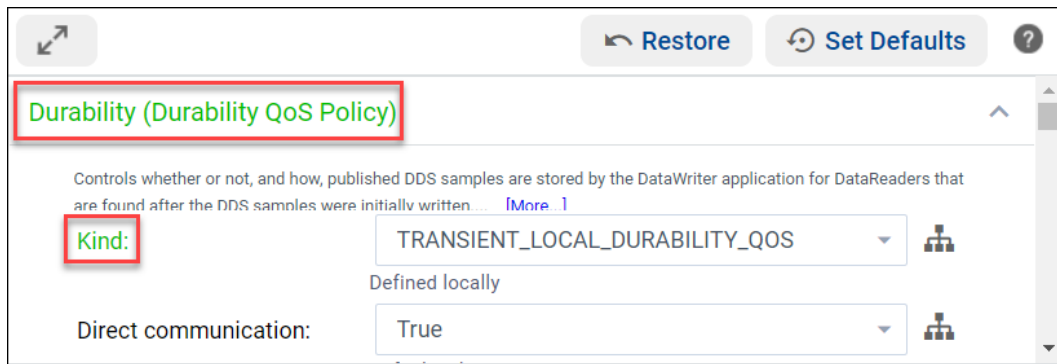
If you don't create a QoS Profile or policy for an Entity, the default QoS is used for that Entity.


A short explanation is shown, and a link to **[More...]** information for that specific policy is provided.

For each QoS policy, suggestions and checks are provided:




Once the QoS is set, it will be highlighted in green to indicate that it is being set locally:



If your profile is inheriting from another profile, System Designer highlights the values that are inherited from the base profile in blue. To create or edit inherited values, click  next to the **Base QoS(s)** field, and then select the QoS Profile to inherit from:

Structured **XML**

Profile Name: (optional)

Base QoS(s): 

Value unset

☐ Show Common Profiles

Base Profile:

BuiltinQoSLib::Baseline.Root


BuiltinQoSLib::Baseline.5.0.0

BuiltinQoSLib::Baseline.5.1.0

BuiltinQoSLib::Baseline.5.2.0


BuiltinQoSLib::Baseline.5.3.0

BuiltinQoSLib::Baseline.6.0.0

BuiltinQoSLib::Baseline.6.1.0 

BuiltinQoSLib::Baseline.7.0.0

BuiltinQoSLib::Baseline.7.1.0

You can see where a value is coming from by selecting  next to the QoS value:

Structured XML

Profile Name: (optional)

Base QoS(s): BuiltinQoSLib::Baseline.6.1.0 ←

Defined locally

Topic Filter: (optional)

Restore Set Defaults ?

Durability (Durability QoS Policy)

Controls whether or not, and how, published DDS samples are stored by the DataWriter application for DataReaders that are found after the DDS samples were initially written. [More](#)

Kind: TRANSIENT_LOCAL_DURABILITY_QOS

Defined locally

Direct communication: True

A summary displays the inheritance chain:

FileDurability → Kind:

Inheritance Chain: (profiles on top have higher priority)

This profile: value=TRANSIENT_LOCAL_DURABILITY_QOS

- Profile 'BuiltinQoSLib::Baseline.6.1.0': value unset
- Profile 'BuiltinQoSLib::Baseline.6.0.0': value unset
- Profile 'BuiltinQoSLib::Baseline.5.3.0': value unset
- Profile 'BuiltinQoSLib::Baseline.5.2.0': value unset
- Profile 'BuiltinQoSLib::Baseline.5.1.0': value unset
- Profile 'BuiltinQoSLib::Baseline.5.0.0': value unset
- Profile 'BuiltinQoSLib::Baseline.Root': value=VOLATILE_DURABILITY_QOS

Info Help

Structured XML

Profile

Base QoS(s)

Topic Filter

Restore Set Defaults ?

Durability (Durability QoS Policy)

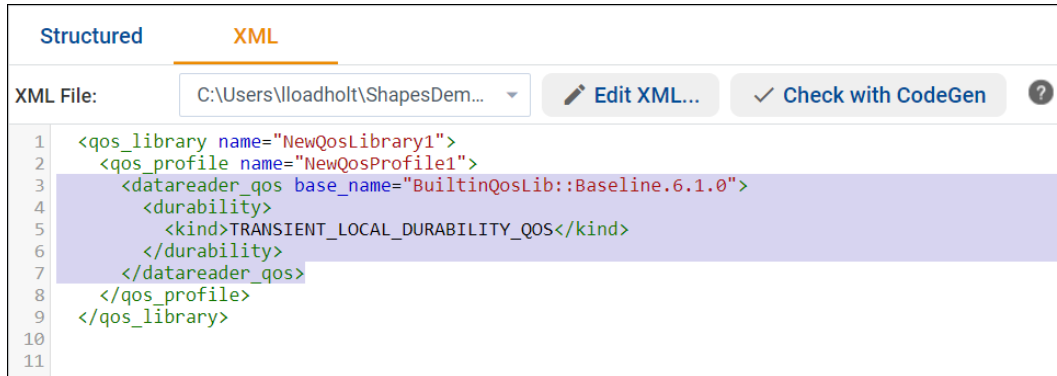
Controls whether or not, and how, published DDS samples are stored by the DataWriter application for DataReaders that are found after the DDS samples were initially written. [More](#)

Kind: TRANSIENT_LOCAL_DURABILITY_QOS

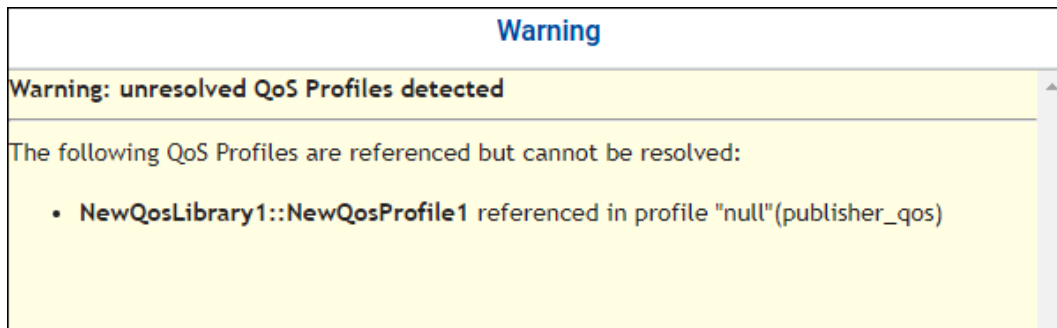
Defined locally

Direct communication: True

The XML view allows you to check the XML results and manually edit each XML file:

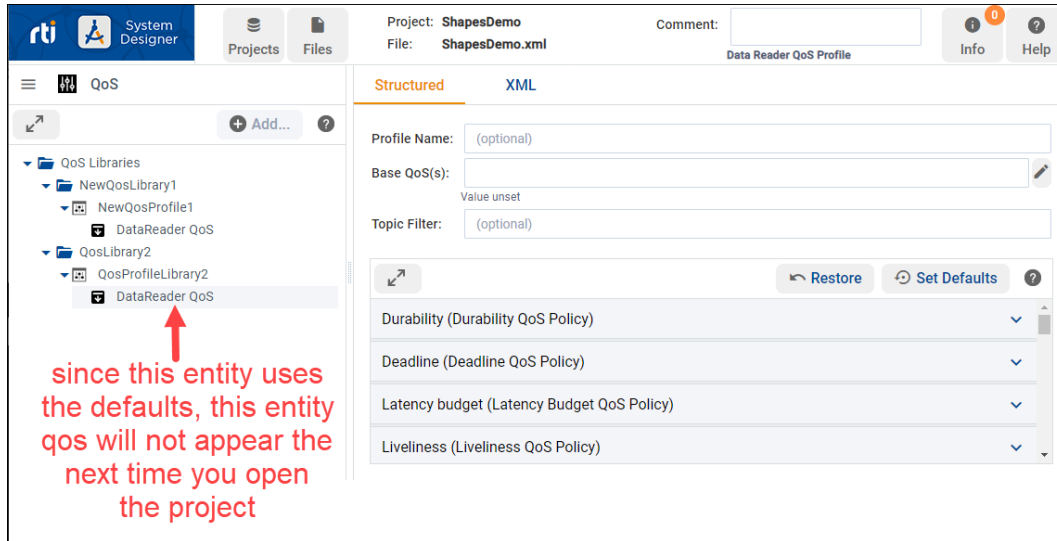


Note: If you later edit a profile name that's already being used in an entity (such as in a `<datawriter_qos>`), the entity does not get updated with your change. You will get a warning when this occurs:



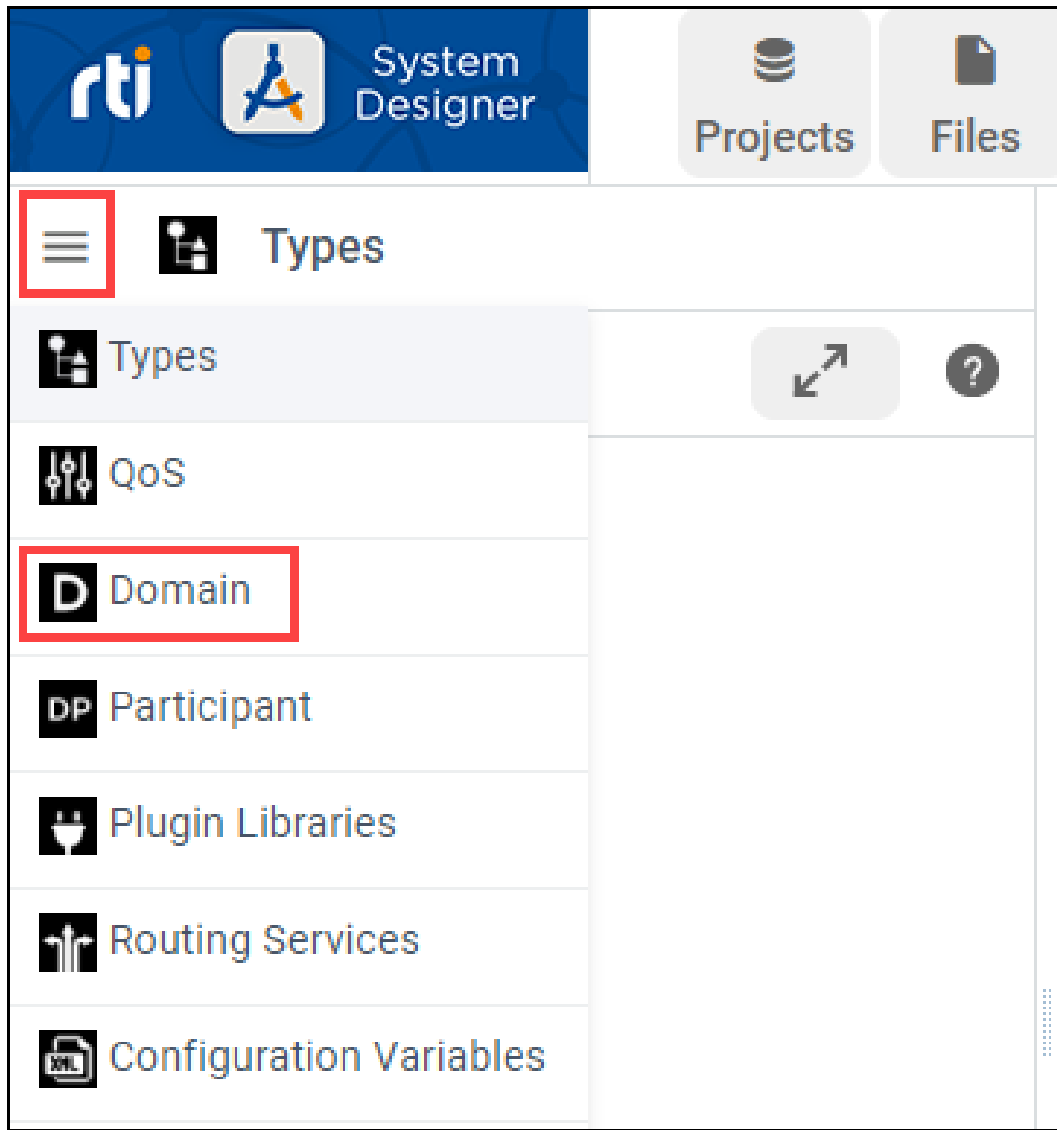
You can select and edit the Entity QoS in the Structured view to use the new name, or edit the XML file directly to use the new name. *System Designer* allows you to change an already used name to give you flexibility in trying out your model. For instance, you may want to use your newly named profile in different entities. Entities that use the old (now unrecognized) name will revert to the default QoS.

Note: About using default QoS: If you create an Entity QoS without modifying its default settings, it will not display the next time you open the project. Entities with no changes to the default QoS use the default settings and are not displayed.



2.4.3 Handling Domains

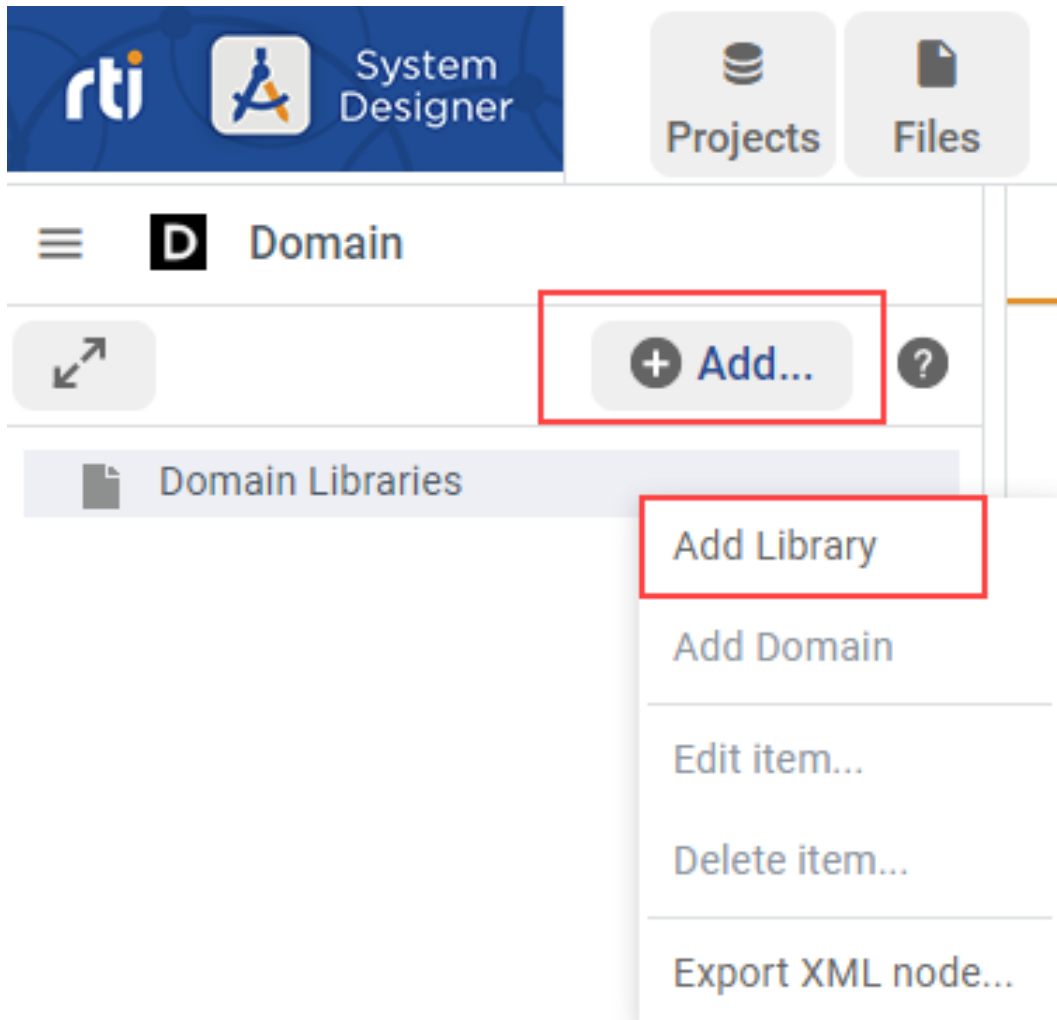
In the Domain view, you can add all your domains and Topics. You can organize them into Domain Libraries.



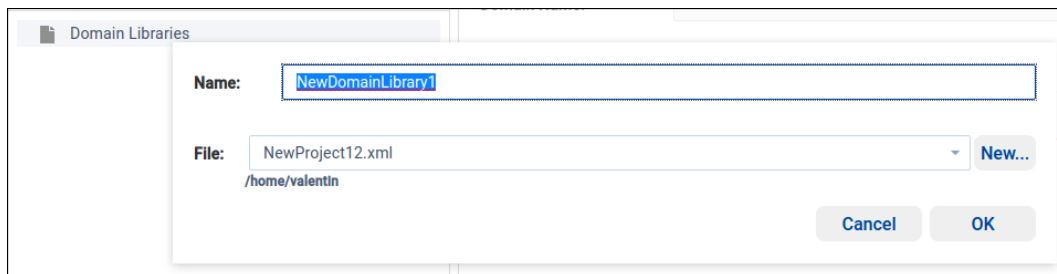
A button with a plus sign icon and the text 'Add...'.


To start a new Domain Library, use the right-click Domain Libraries in the tree:

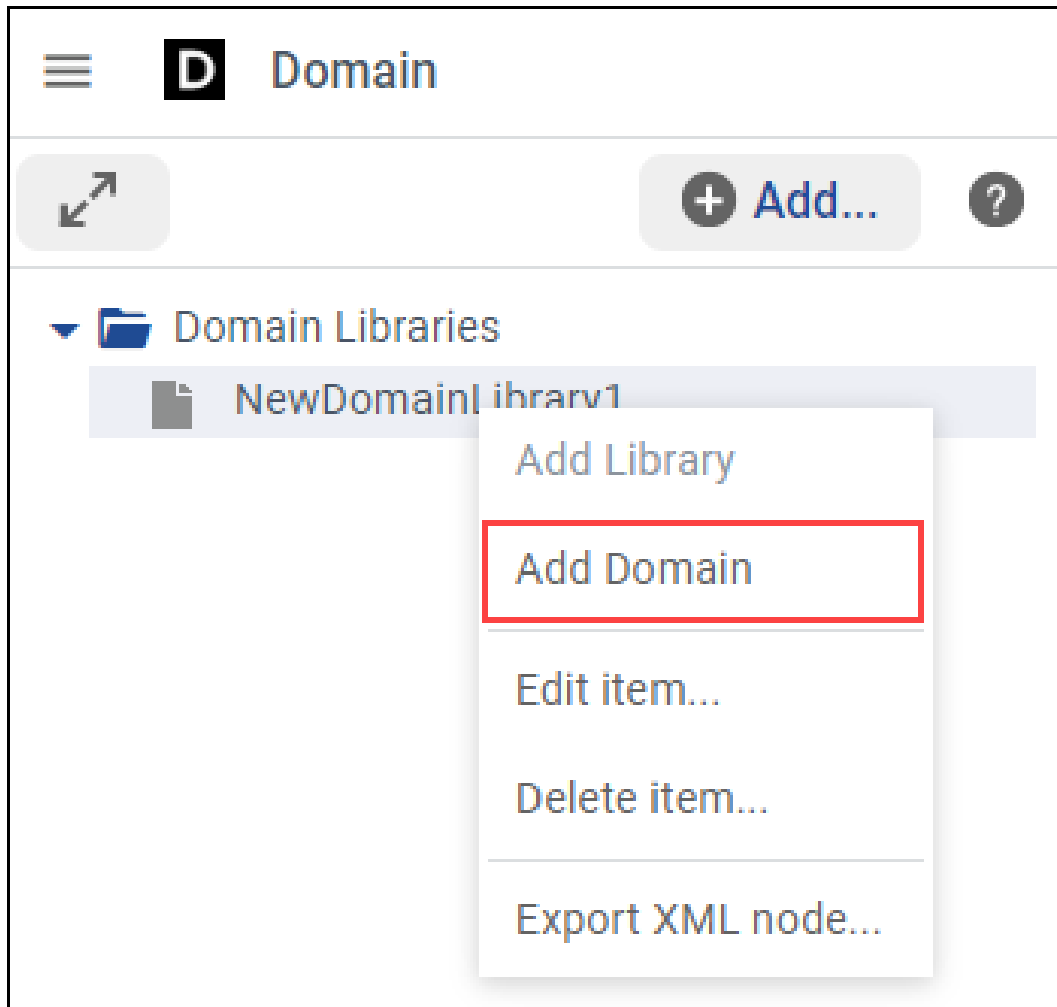
button while Domain Libraries is selected or



To add a Domain Library, enter the Library Name and the destination file:



Then you can add a domain to the new library by using the  button while a Domain Library is selected or by right-clicking the Domain Library:



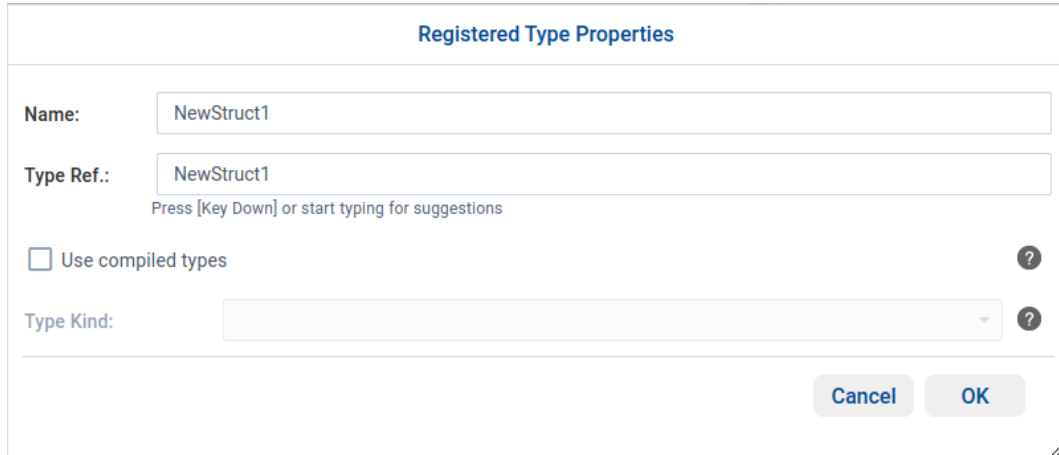
A pop-up window will guide you to enter the Domain Name, the Domain ID, and the destination file:

A screenshot of the 'Domain Properties' dialog box. It has a title bar with the text 'Domain Properties'. Inside, there are three input fields: 'Name:' with the value 'NewDomain1', 'Domain Id:' which is empty, and 'File:' with the value 'NewProject12.xml'. Below the 'File:' field is a file path 'C:\Users\rkorte\' and a 'New...' button. At the bottom right are 'Cancel' and 'OK' buttons.

Once the domain is created, you can register types with the domain and add a new Topic in the Structured view.

To register a new type, select  **Add...** in the **Registered types** table; a pop-up will allow you to choose

a Type Name and Type Reference:

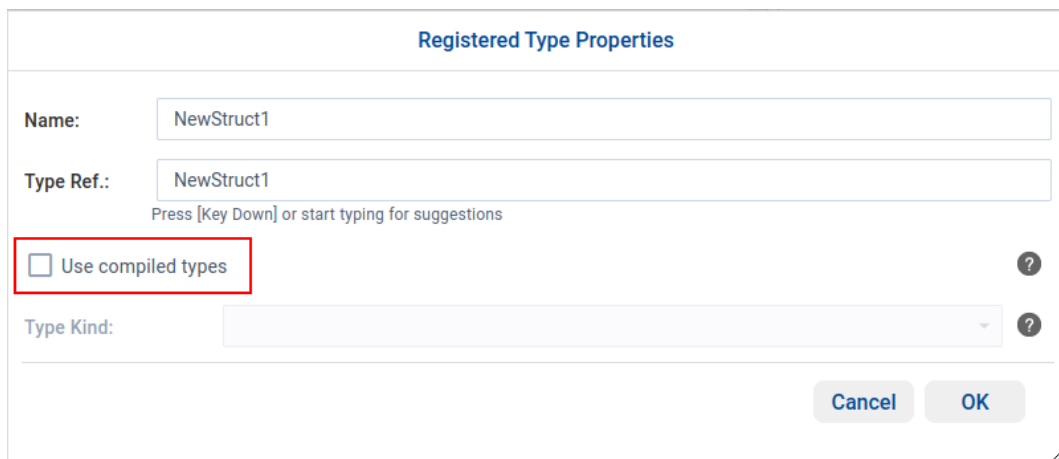


The dialog box is titled "Registered Type Properties". It contains the following fields and controls:

- Name:** A text box containing "NewStruct1".
- Type Ref.:** A text box containing "NewStruct1". Below it is a hint: "Press [Key Down] or start typing for suggestions".
- Use compiled types:** A checkbox that is currently unchecked. To its right is a question mark icon.
- Type Kind:** A dropdown menu that is currently empty. To its right is a question mark icon.
- Buttons:** "Cancel" and "OK" buttons at the bottom right.

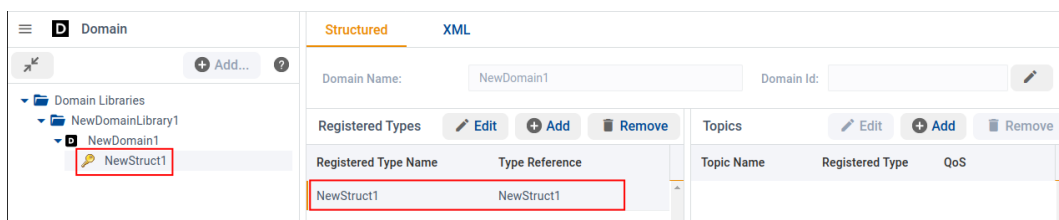
Note: The Type Kind is **deprecated** and you cannot set this value from the dialog. For legacy configurations, if the imported XML contains this attribute, it will be preserved and show in the dialog.

By default, *System Designer* will check if the referenced type is defined in the Types view. If you are planning to use Compiled Types, select the **User compiled types** checkbox and *System Designer* will not check if the referenced type is defined in the Types view.



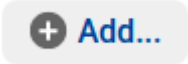
This image shows the same "Registered Type Properties" dialog box as above, but with a red rectangle highlighting the "Use compiled types" checkbox, which remains unchecked. All other fields and controls are identical to the previous image.

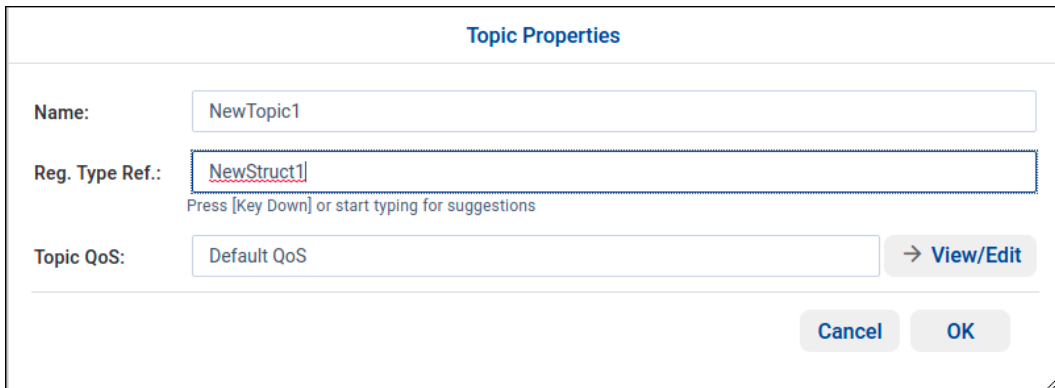
After you click **OK**, the new registered type will appear in the left tree and in the Registered types table:



The image shows the System Designer interface with two main panes:

- Left Pane (Domain Tree):** Shows a hierarchy of domains. Under "Domain Libraries", there is "NewDomainLibrary1", which contains "NewDomain1", which in turn contains "NewStruct1". The "NewStruct1" item is highlighted with a red rectangle.
- Right Pane (Structured/XML View):**
 - Domain Name:** "NewDomain1"
 - Domain Id:** (empty)
 - Registered Types:** A table with two columns: "Registered Type Name" and "Type Reference". It contains one entry: "NewStruct1" mapped to "NewStruct1". This entry is highlighted with a red rectangle.
 - Topics:** A table with three columns: "Topic Name", "Registered Type", and "QoS". It is currently empty.

To register a new Topic, click  in the Topics table. A pop-up will open:

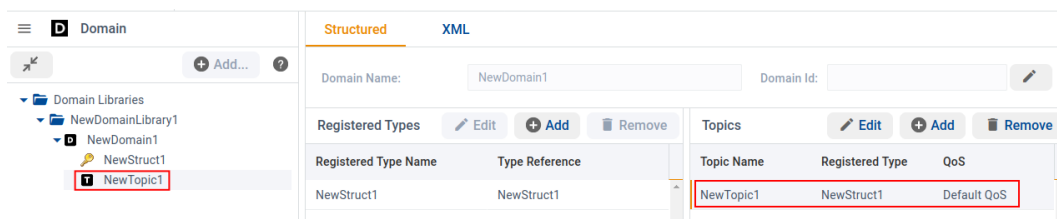


The **Topic Properties** dialog box contains the following fields and controls:

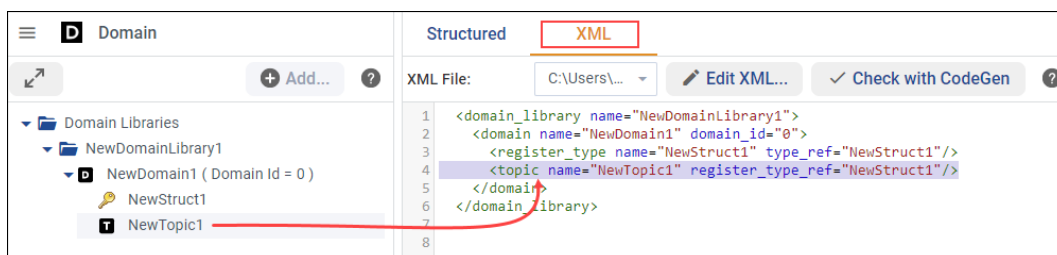
- Name:** A text field containing "NewTopic1".
- Reg. Type Ref.:** A text field containing "NewStruct1". Below it, a hint text says "Press [Key Down] or start typing for suggestions".
- Topic QoS:** A text field containing "Default QoS". To its right is a **View/Edit** button.
- At the bottom right are **Cancel** and **OK** buttons.

Enter your topic name and select one of the available Registered type names from the drop-down menu.

After you click **OK**, the new topic will appear in the left tree and in the main panel:




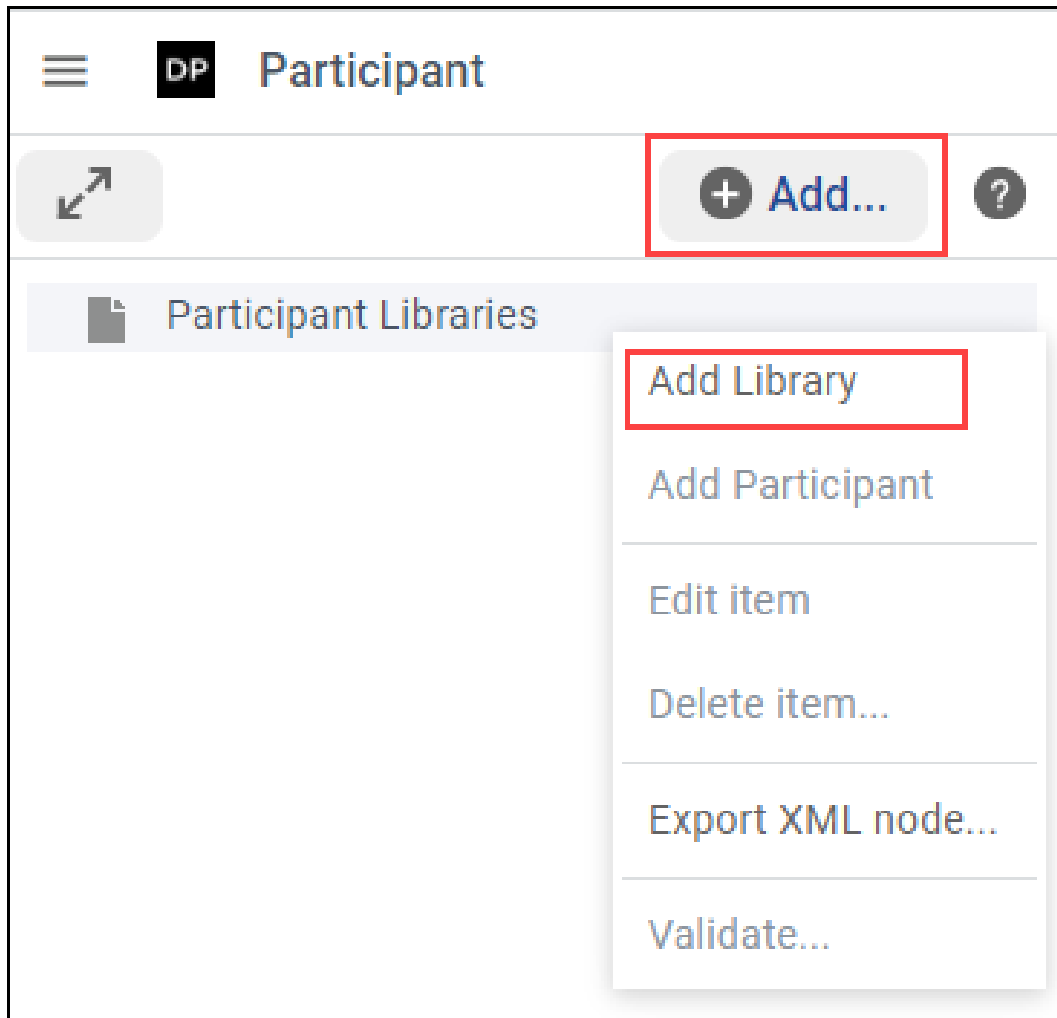
Of course, you can also see the XML definition in the XML view:



2.4.4 Define Participants and Entities

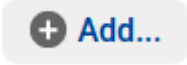
The **Participant** view allows you to add Participants, DataReaders, and DataWriters. You have to organize them into Participant Libraries.

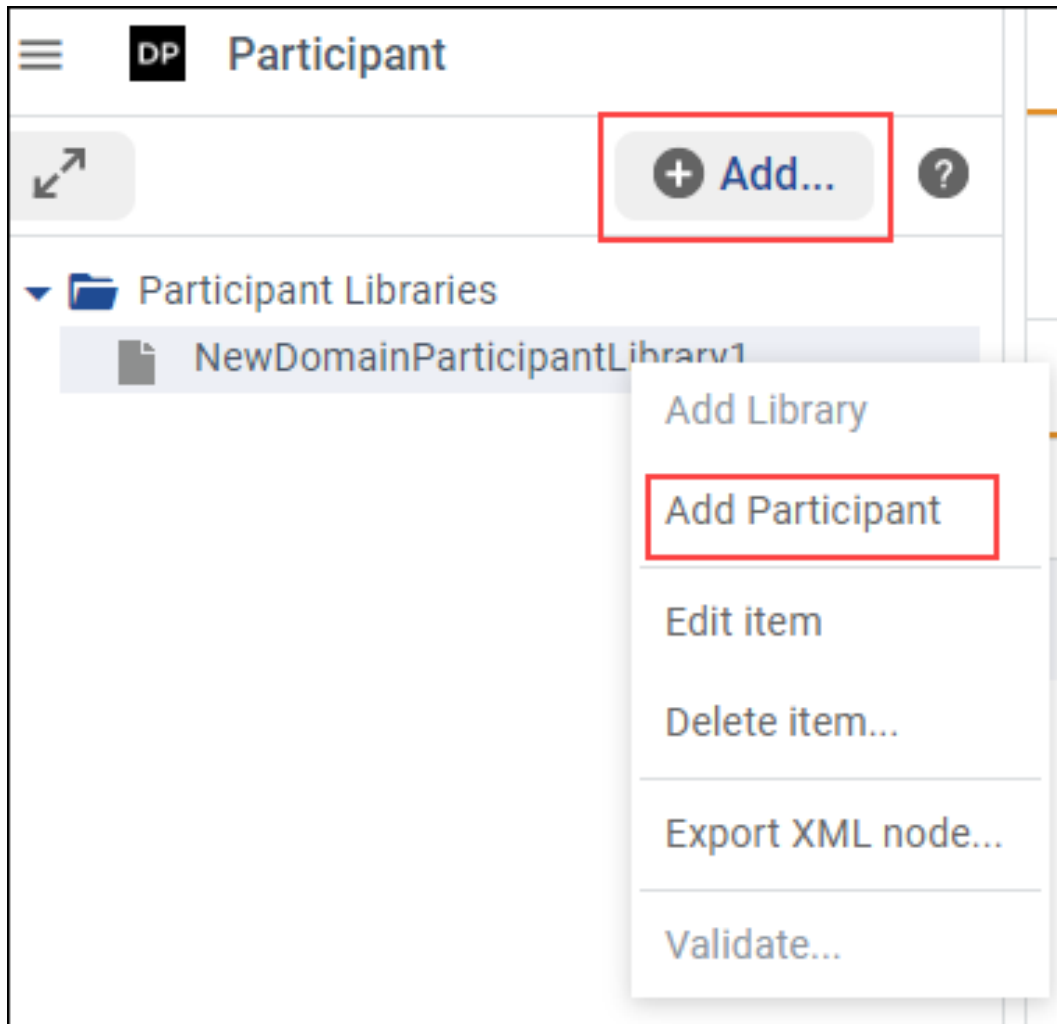
To add a new Participant Library to your model, click the  button while Participant Libraries is selected or right-click Participant Libraries in the tree and select **Add Library**:



To add a Participant Library, enter the Library Name and the destination file:

The image shows a dialog box for adding a library. It has two main input fields: 'Name' and 'File'. The 'Name' field contains the text 'NewDomainParticipantLibrary1'. The 'File' field contains 'ShapesDemo.xml' and has a dropdown arrow and a 'New...' button next to it. At the bottom of the dialog, there are 'Cancel' and 'OK' buttons.

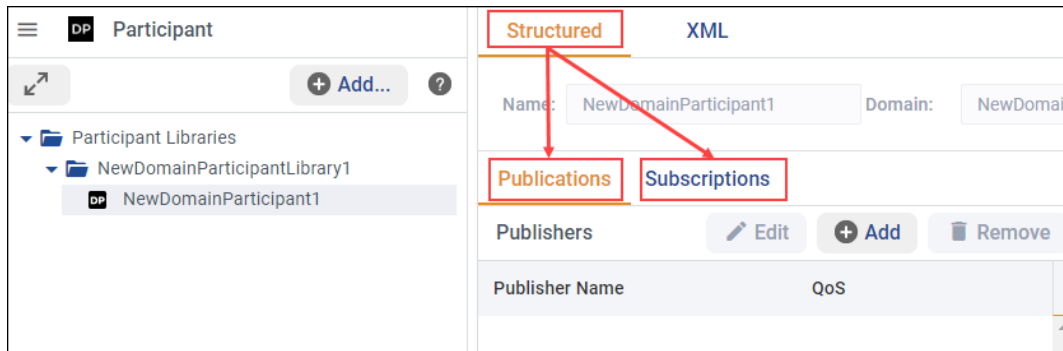
You can then add a Participant to the newly created Library by using the  button or by right-clicking on the Participant Library in the tree and selecting **Add Participant**:



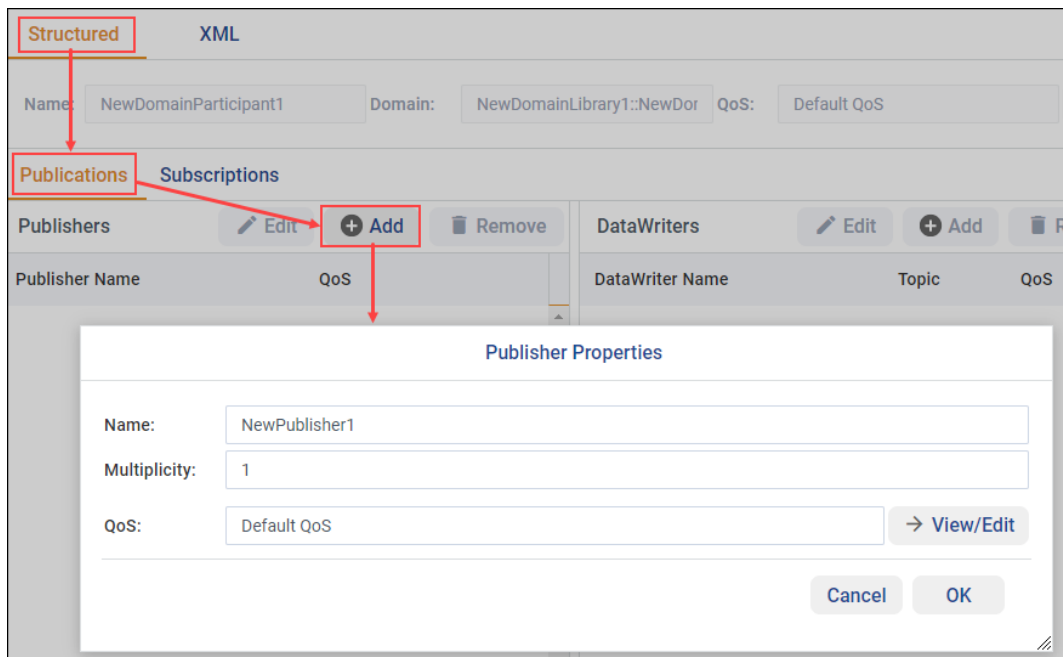
A dialog will guide you to pick a name and select one of the existing domains as the Domain Reference. You can also view and edit the QoS and select a file where the new participant will be stored:

The screenshot shows the 'DomainParticipant Properties' dialog box. It has a title bar with the text 'DomainParticipant Properties'. Inside the dialog, there are four main sections: 'Name:' with a text field containing 'NewDomainParticipant1'; 'Domain Ref.:' with a text field containing 'NewDomainLibrary1::NewDomain1' and a hint 'Press [Key Down] or start typing for suggestions' below it; 'QoS:' with a text field containing 'Default QoS' and a 'View/Edit' button to its right; and 'File:' with a text field containing 'ShapesDemo.xml' and a 'New...' button to its right. Below the 'File:' field, the path 'C:\Users\loadholt' is visible. At the bottom right of the dialog, there are 'Cancel' and 'OK' buttons.

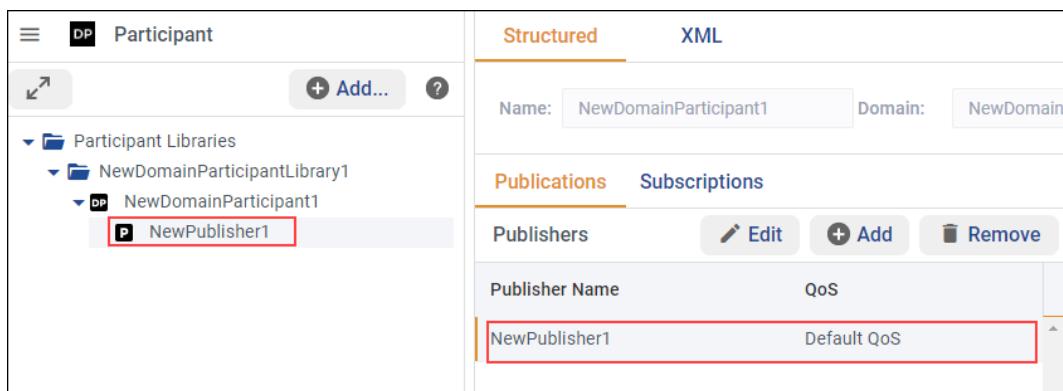
Now you can configure Publications and Subscriptions in the structured view:



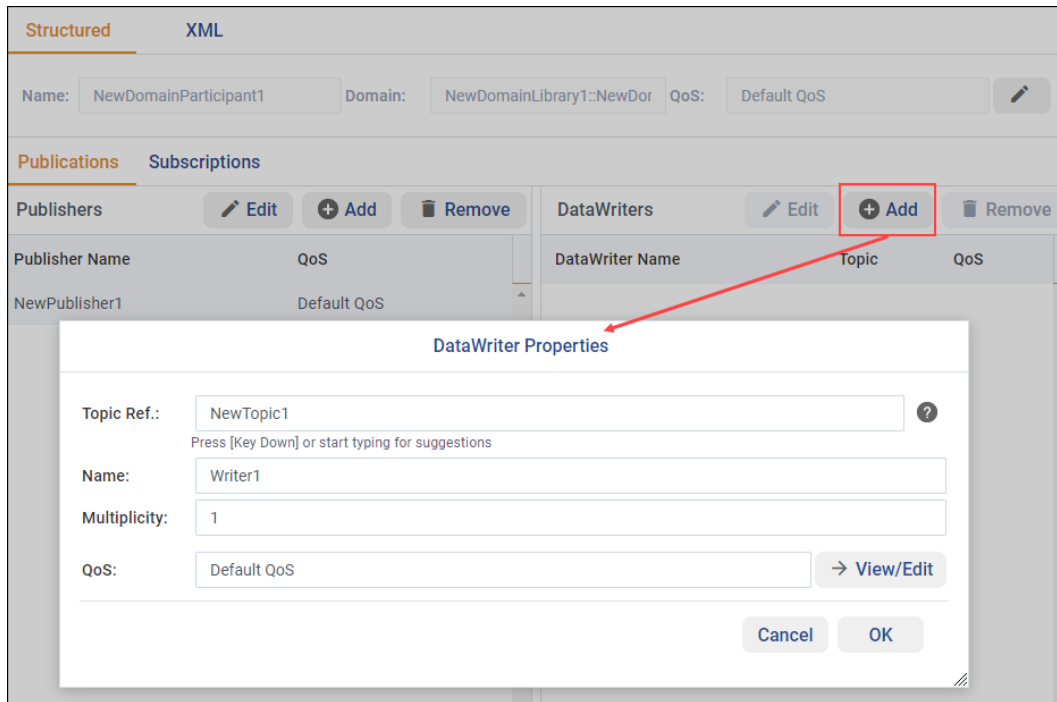
To add a new Publisher, select the **+ Add...** button in the Publishers table. You can then select Name, Multiplicity, and QoS Policies in the presented dialog:



Once you click **OK**, the new Publisher will be available in the Structured view and in the tree on the left:

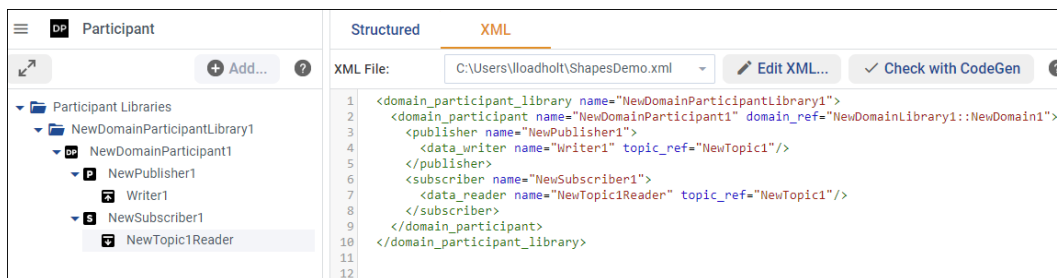


Now you can add DataWriters to the selected Publisher. Use the  **Add...** button and in the pop-up window, select the topic reference, the name of the DataWriter, Multiplicity, and QoS:



Follow the same logic to add Subscribers and DataReaders in the Subscriptions view.

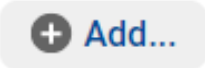
As always, you can see and modify the XML file directly in the XML view:

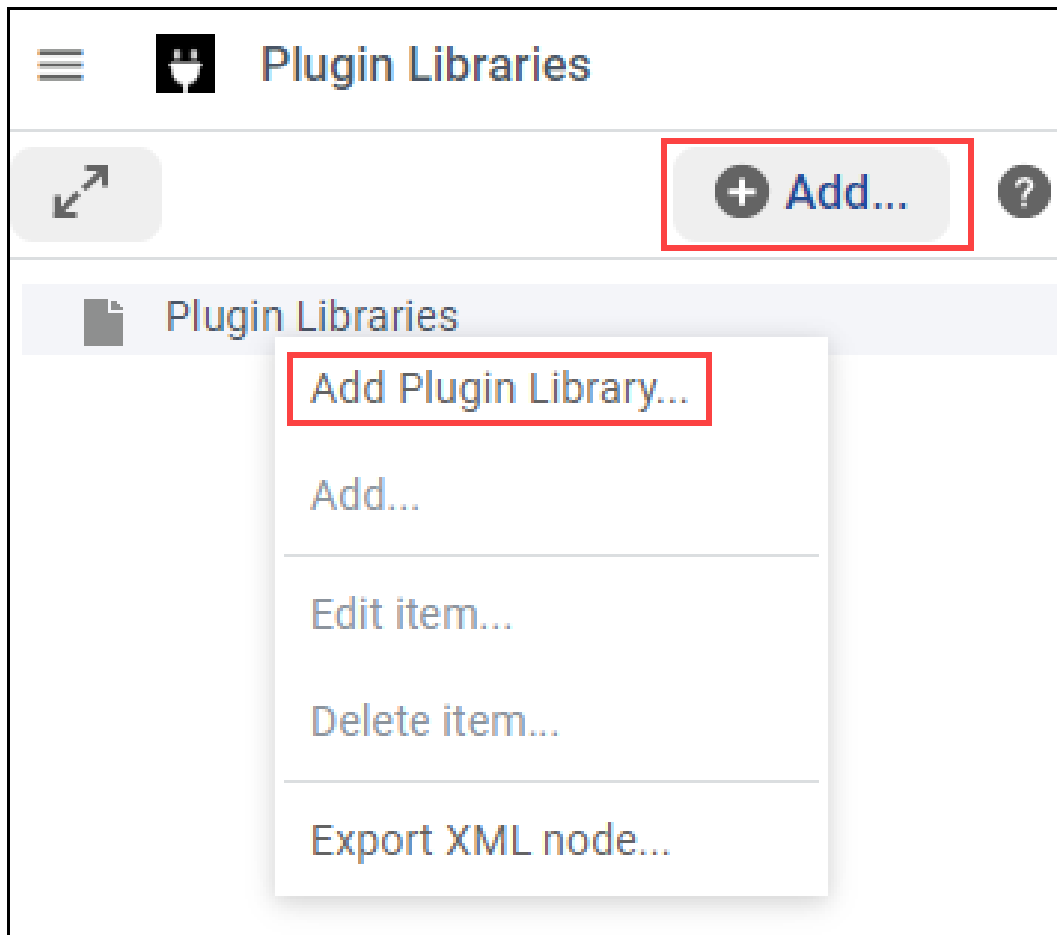


2.4.5 Define Plugins

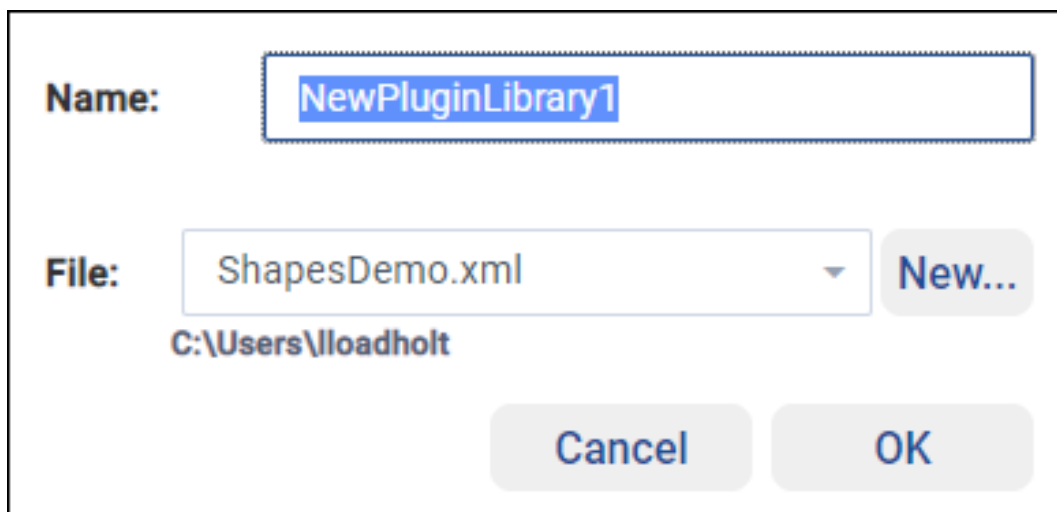
The **Plugin** view allows you to define the pluggable components specific to a Routing Service. These plugins are:

- Organized in Plugin Libraries.
- Categorized and configured based on the source language. Routing Service supports C/C++ and Java plugins.

To add a new Plugin Library to your model, click the  button while Plugin Libraries is selected or right-click Plugin Libraries in the tree and select **Add Plugin Library**:

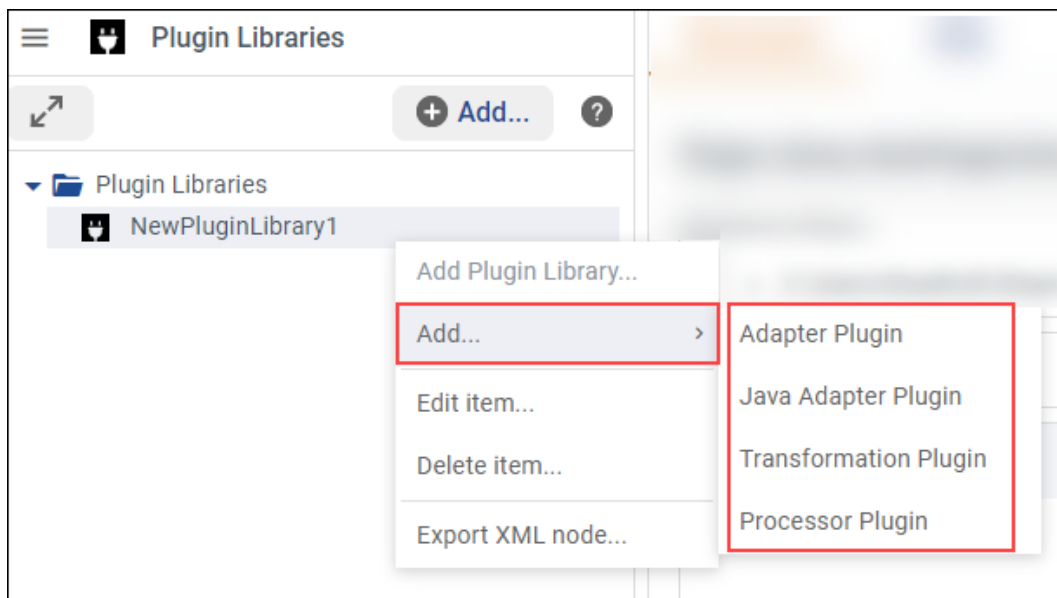
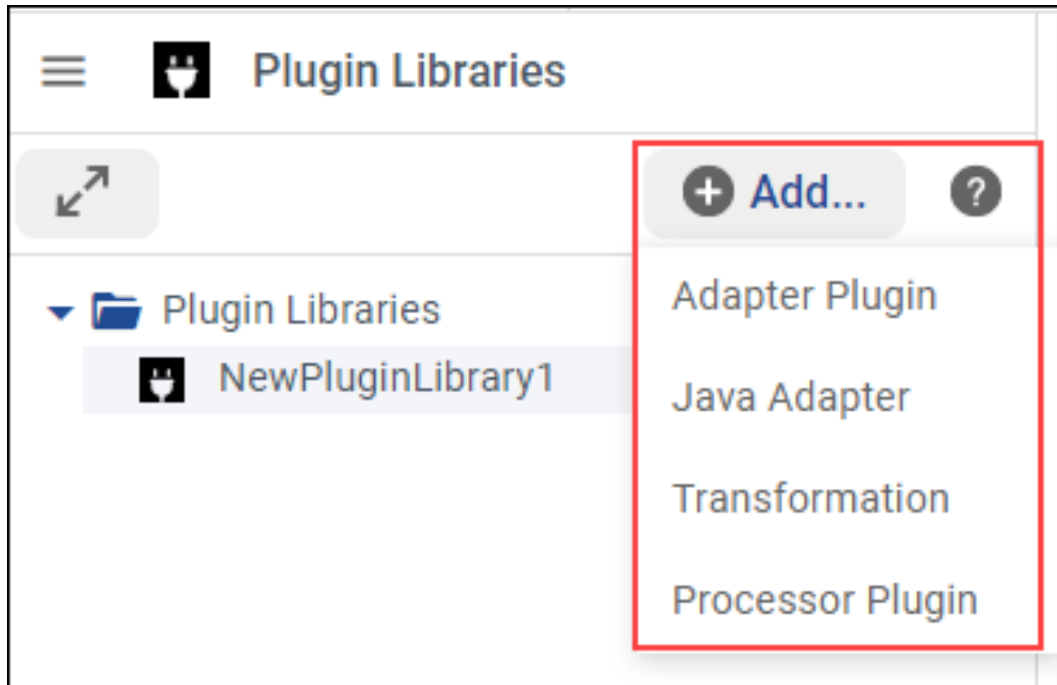


To add a Plugin Library, enter the Library Name and the destination file:

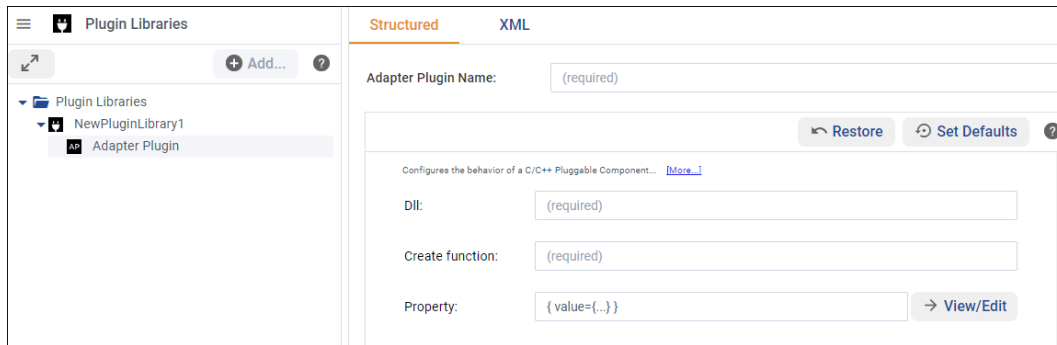


 **Add...**

Then you can add any of the available plugins to the newly created Library by using the button or by right-clicking on the Plugin Library in the tree and selecting **Add**:



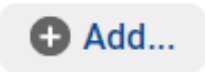
When a Plugin is selected, you can set all the configuration of each different plugin language in the Structured view:

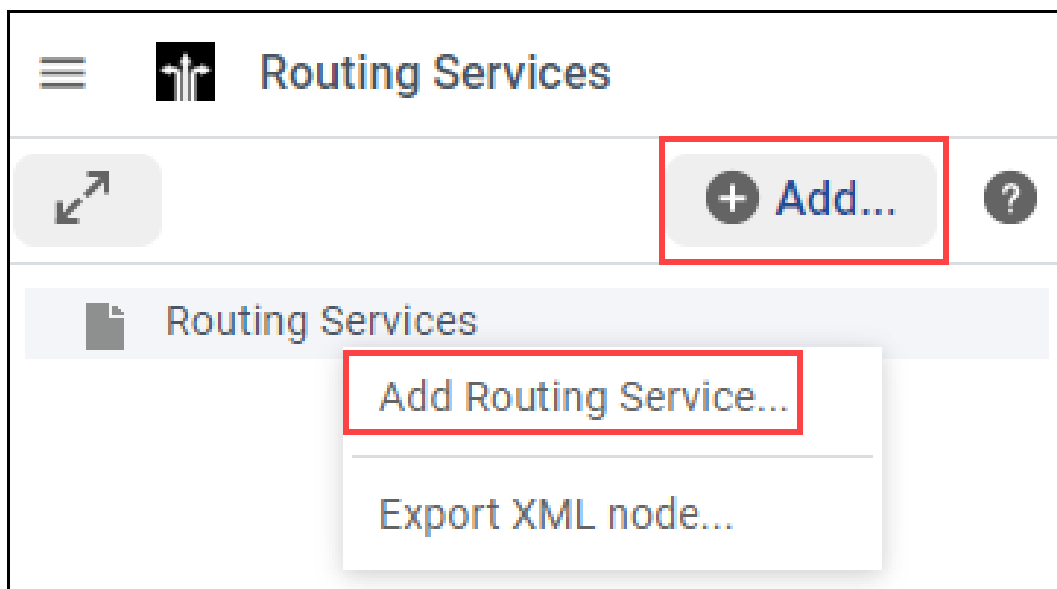


See [Plugin Management \(Configuration\)](#) in the *RTI Routing Service User's Manual* for more information on the configuration available for each plugin language.

2.4.6 Define Routing Services

The **Routing Service** view allows you to define the configuration of an execution of *Routing Service*.

To add a new Routing Service to your model, click the  **Add...** button while Routing Services is selected or right-click Routing Services in the tree and select **Add Routing Service...**:



A pop-up window will appear, where you can enter the Routing Service name, and the file where the Routing Service is going to be saved:

Name:

File: New...

C:\Users\loadholt

Cancel OK

When a Routing Service is selected, you can set all the different aspects of the configurations of a Routing Service in the Structured view:

Routing Services

Structured **XML**

Name:

Enabled:

Group Name:

File:

Restore **Set Defaults** **?**

Annotation (Annotation Configuration)

Administration (Administration Configuration)

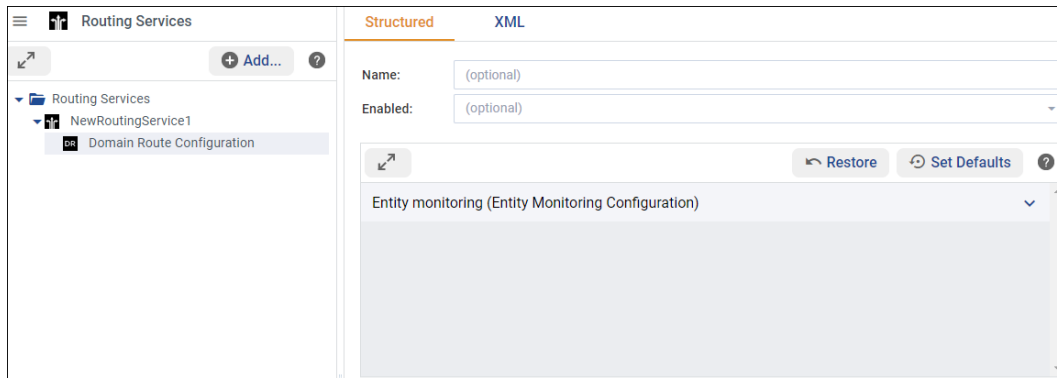
Monitoring (Monitoring Configuration)

Domain Routes **+** **?**

Item	Name	Contains modified content



See [Routing Service Tag](#) in the *RTI Routing Service User's Manual* for more information on the configuration available for an execution of Routing Service.

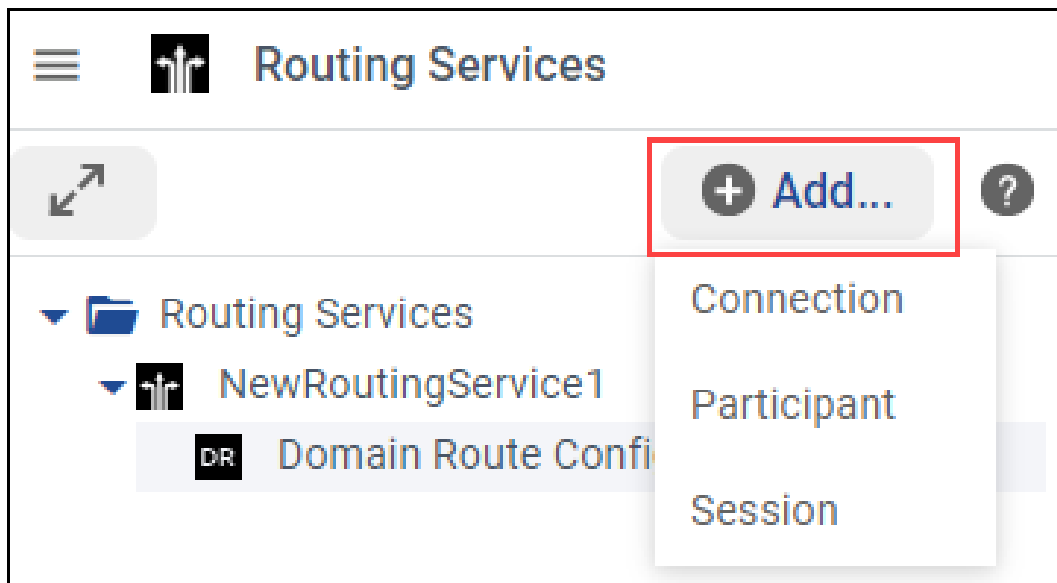
Once you have your Routing Service defined, you can select one of them and add a Domain Route. A Domain Route defines a mapping between different data domains, data available in any of these data domains can be routed to other data domains.



See [Domain Route](#) in the *RTI Routing Service User's Manual* for more information on the configuration available for a mapping between different data domains.

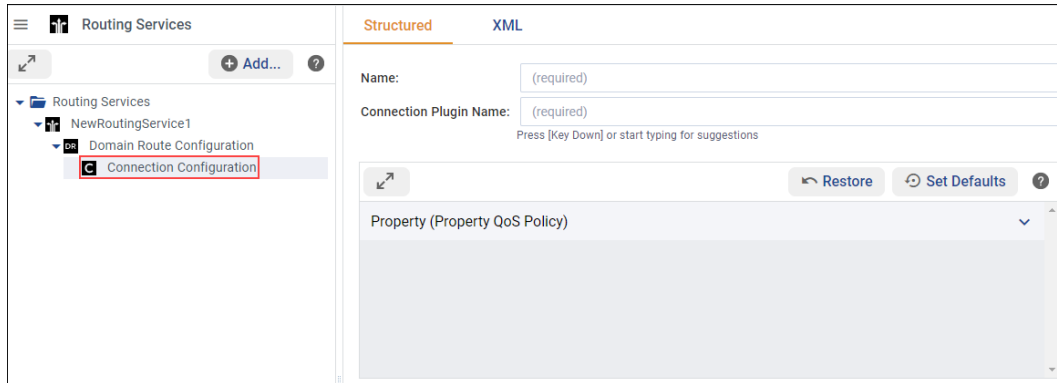
When a Domain Route is selected, you can define Connections, Participants, and/or Sessions by clicking on

the  **Add...** button while the Domain Route is selected, or click on the  in the Connections, Participants & Sessions table:

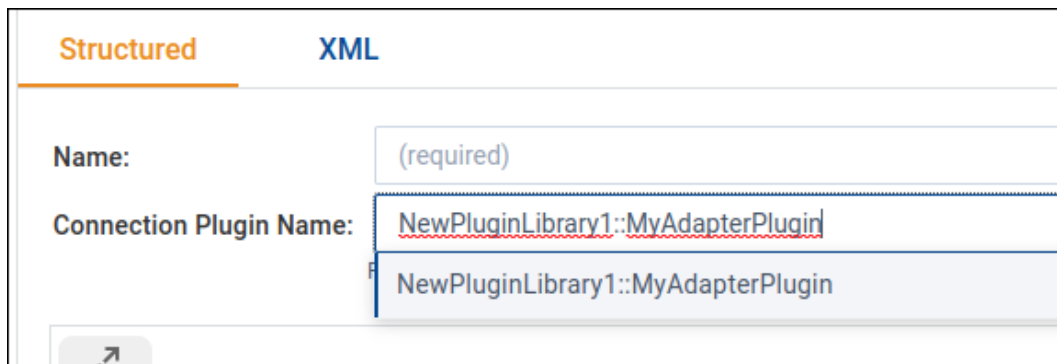


Next, we'll explain the elements that are configurable inside a Domain Route:

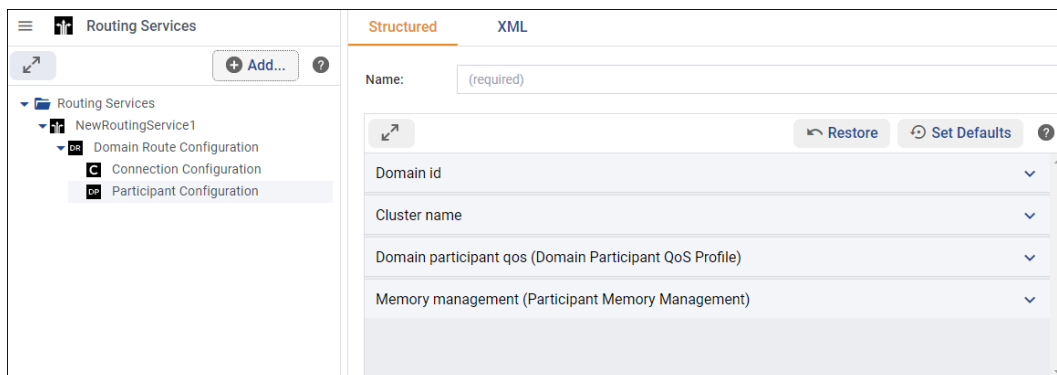
- **Connections:** A Connection requires a *name*, which will be used by the **Route** to select input and output domains, and a *plugin_name*, which will be used to associate a **Connection** with an adapter plugin defined in the Plugin Library view.



System Designer will display the adapter plugin defined in the Plugin Library view if you press the Key Down in the *plugin_name* field, or you start typing for suggestions.



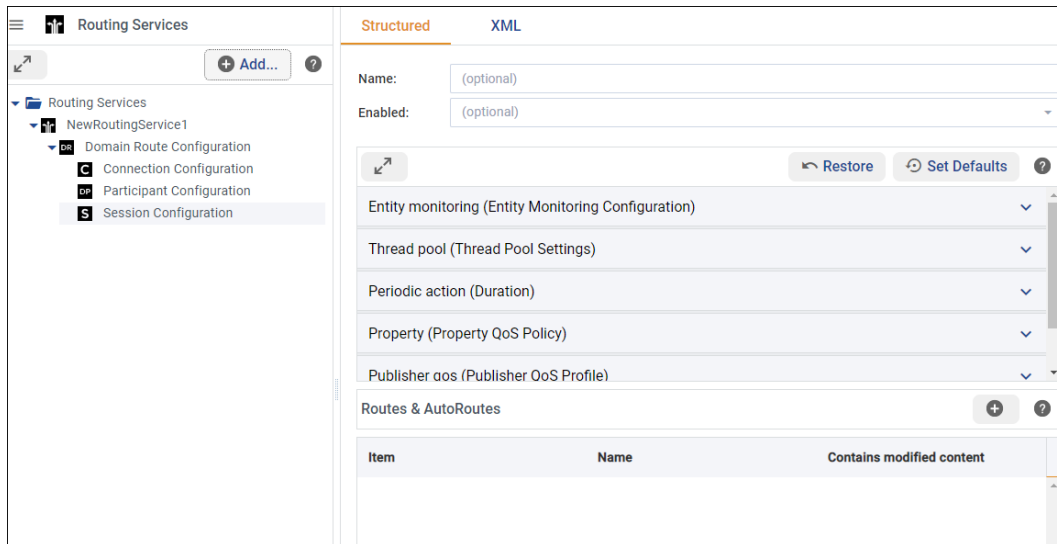
- **Participants:** Routing Service comes with a builtin implementation of a DDS adapter. A Participant corresponds to exactly one DomainParticipant.





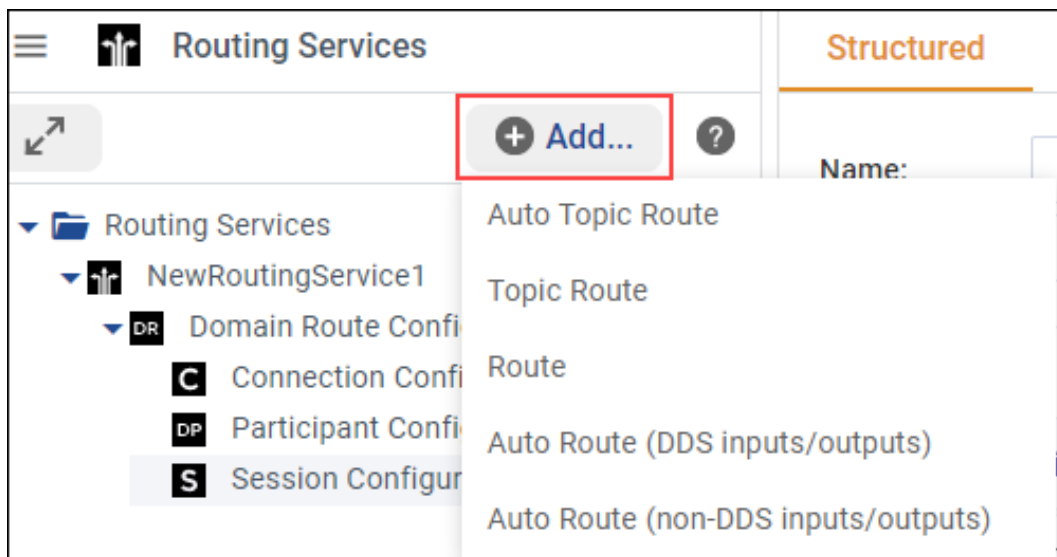
- **Sessions:** A Session defines a multi-threaded context for route processing, including data forwarding. The data is routed according to specified **Routes** and **AutoRoutes**.

Each Session will have an associated thread pool to process Routes concurrently, preserving Route safety. Multiple Routes can be processed concurrently, but a single Route can be processed only by one thread at a time. By default, the session thread pool has a single thread, which serializes the processing of all the Routes.

Sessions that bridge domains will create a Publisher and a Subscriber from the DomainParticipants associated with the domains.

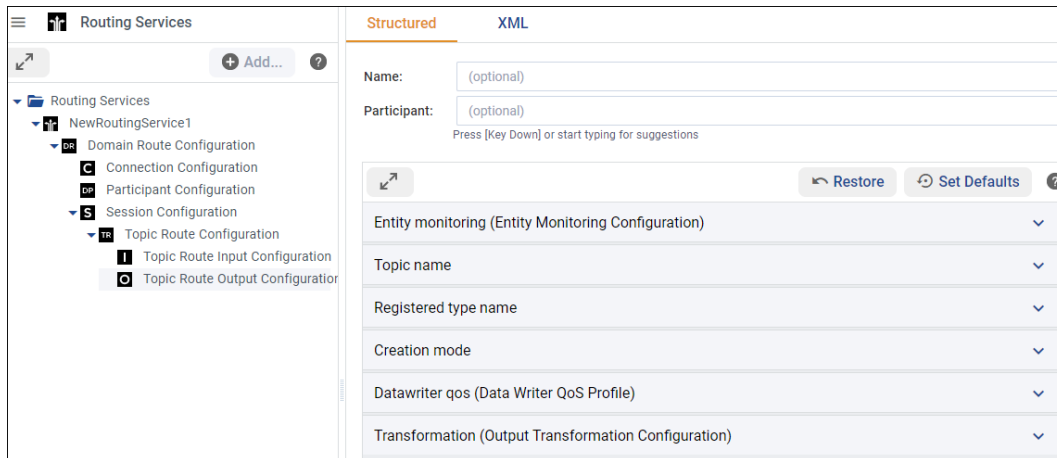


When a Session is selected, you can define Routes or Auto Routes by clicking on the  **Add...** button while the Session is selected, or click on the  in the Routes & AutoRoutes table:

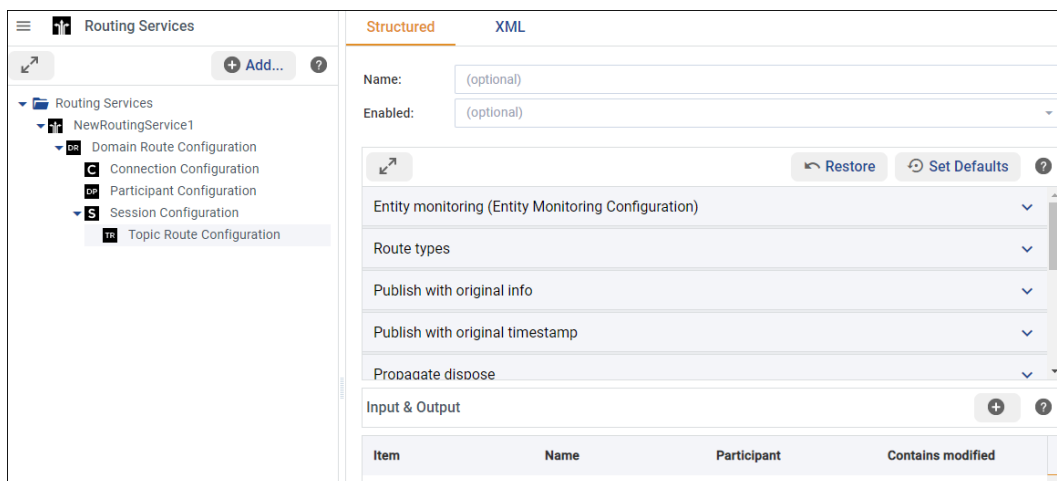


- **Routes & Topic Routes:** A Route explicitly defines a mapping between one or more input data streams and one or more output data streams. The input and output streams may belong to different data domains.

Route events are processed in the context of the thread belonging to the parent Session. Route event processing includes, among others, calls to the StreamReader read and StreamWriter write operations.

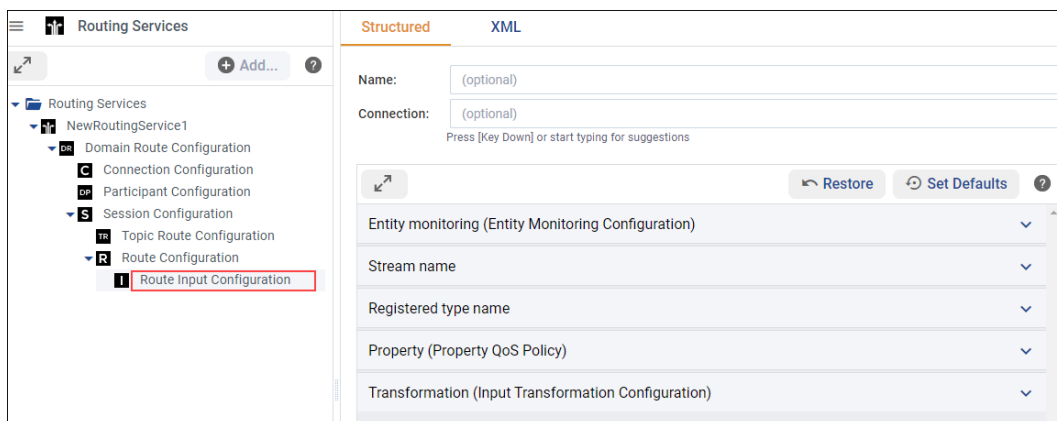
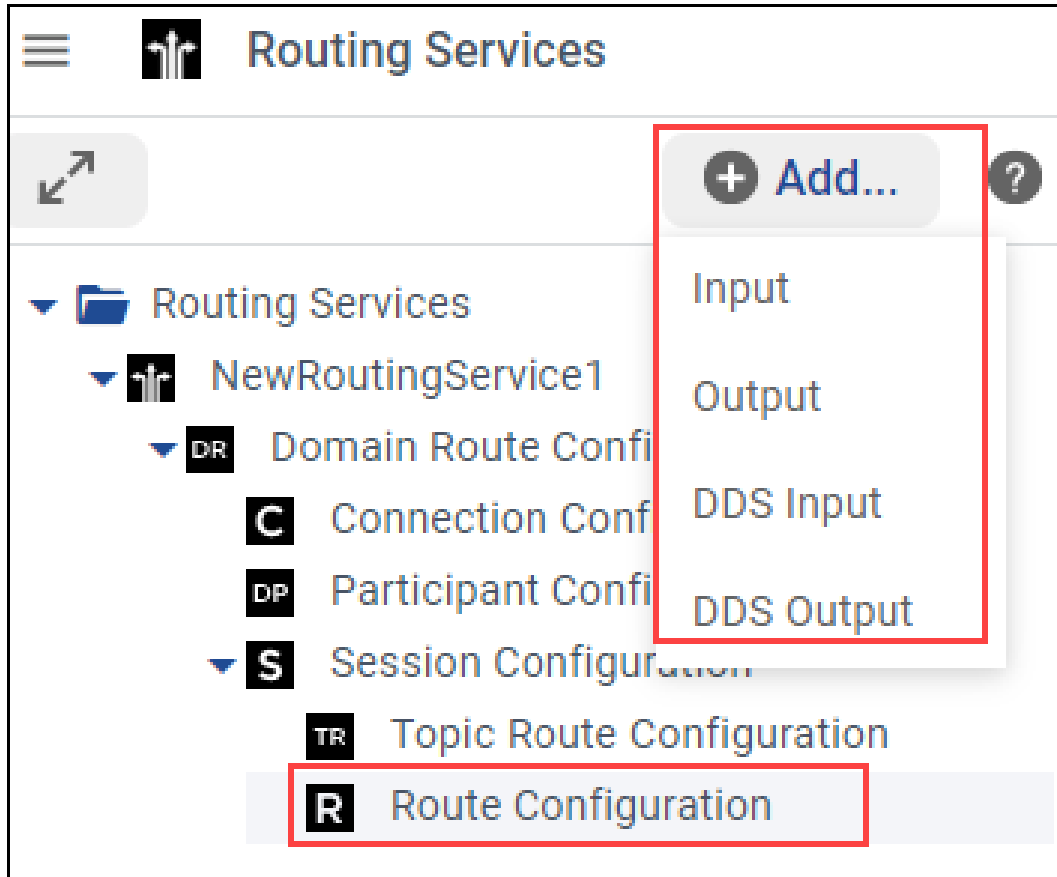


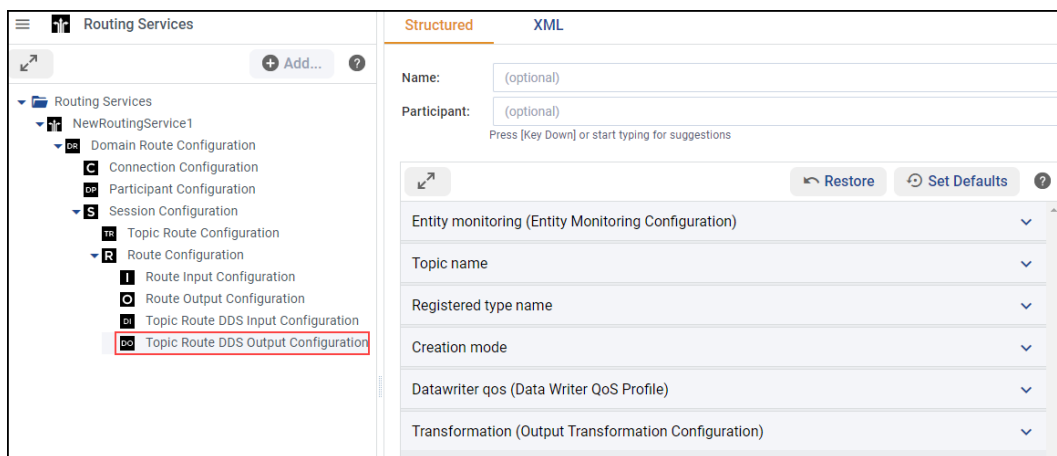
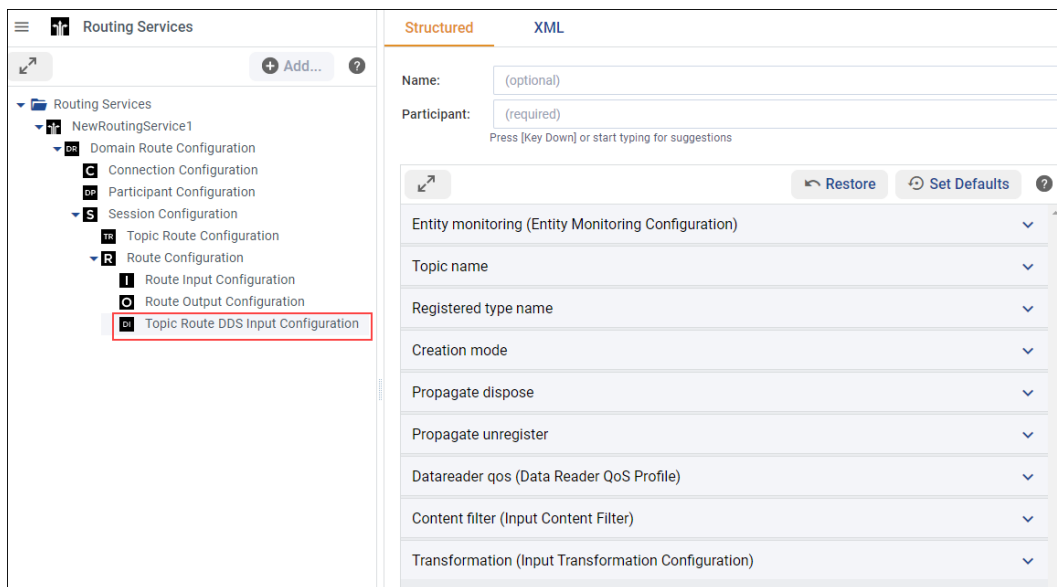
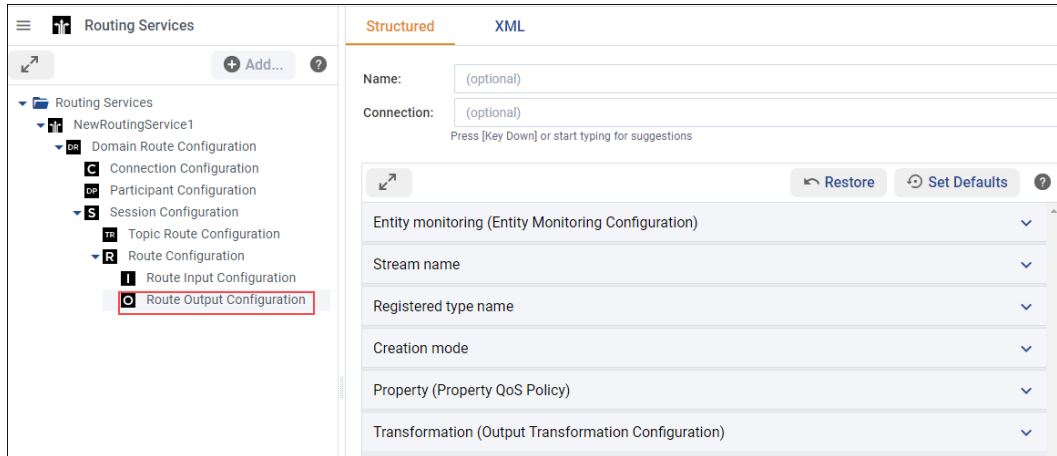
See [Route](#) in the *RTI Routing Service User's Manual* for more information on the configuration available for a Route.



See [Topic Route](#) in the *RTI Routing Service User's Manual* for more information on the configuration available for a Topic Route.

- * **Input/Output:** Inputs and outputs in a Route or TopicRoute have an associated StreamReader and StreamWriter, respectively. For DDS domains, the StreamReader will contain a DataReader and the StreamWriter will contain a DataWriter. The DataReaders and DataWriters belong to the corresponding Session Subscriber and Publisher.



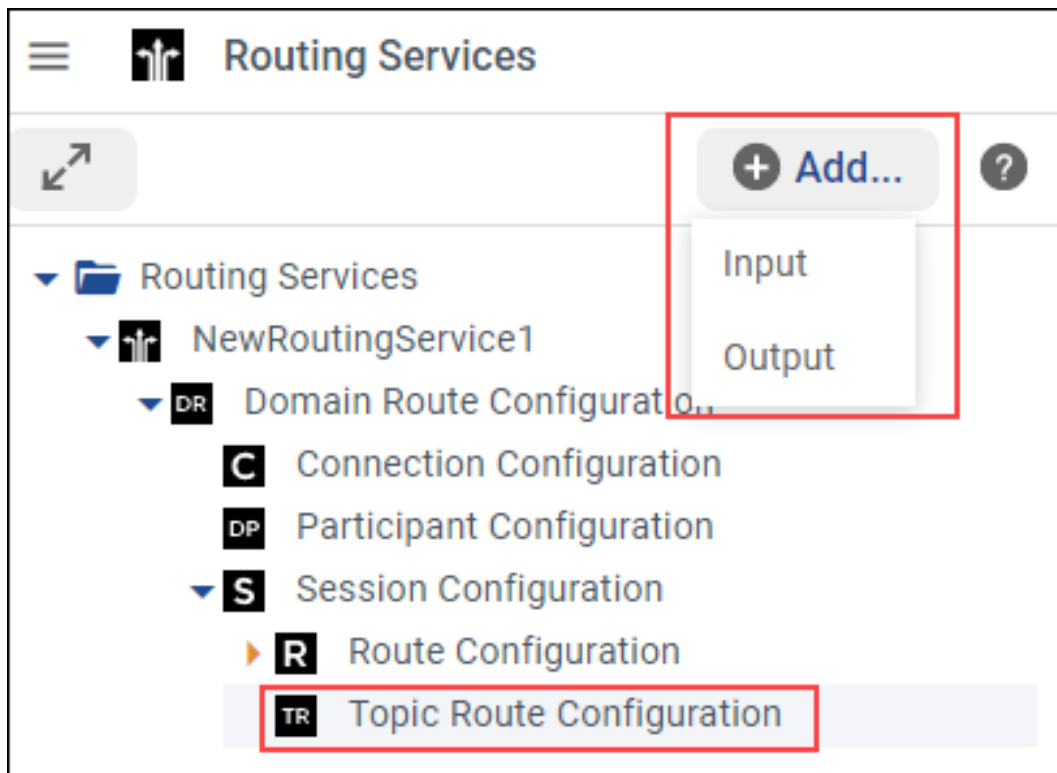


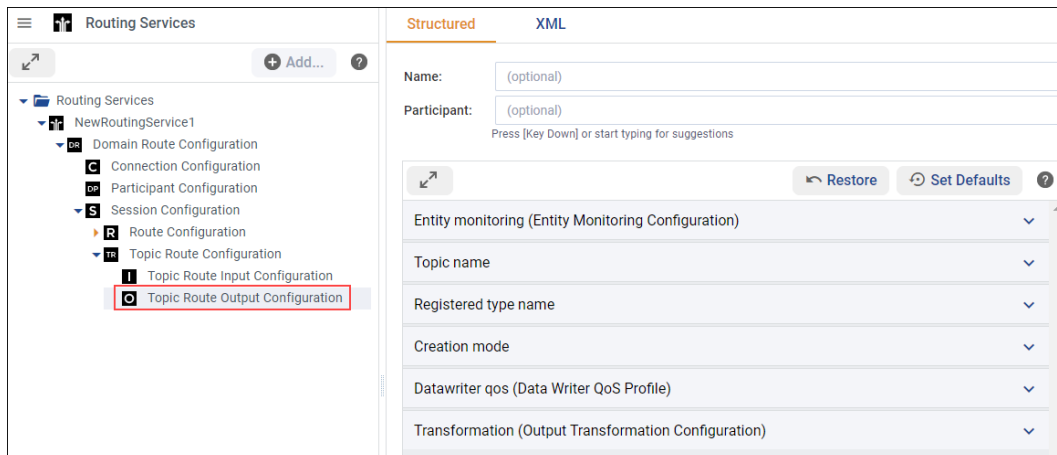
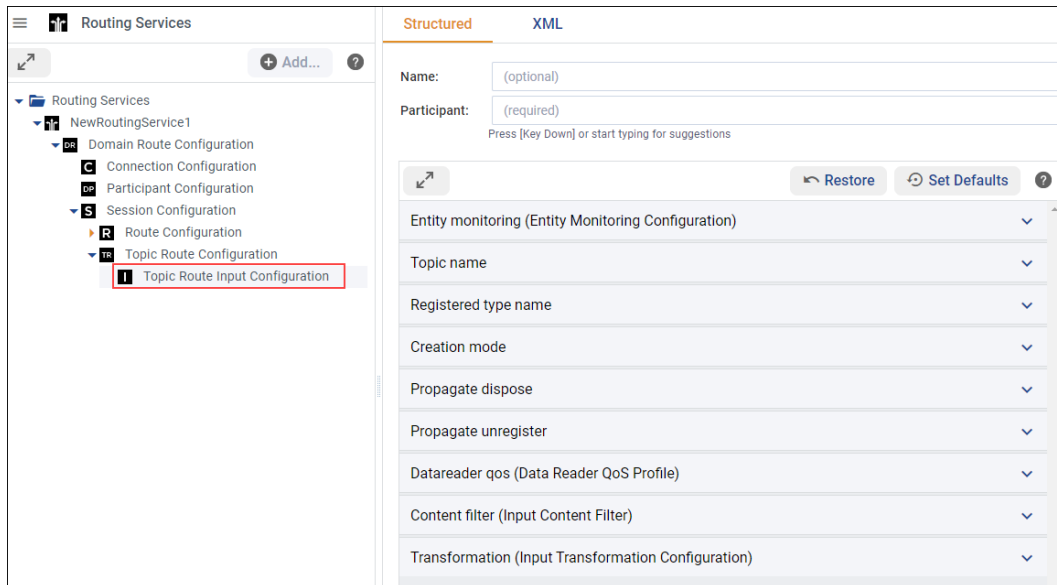
System Designer will display the Connection defined in the Routing Service view if you press the Key Down in the *connection* field, or you start typing for suggestions.

Structured	XML
Name:	<input type="text" value="RouteInput"/>
Connection:	<input type="text" value="NewRoutingService1::DomainRoute::MyConnection"/>
	<input type="text" value="NewRoutingService1::DomainRoute::MyConnection"/>

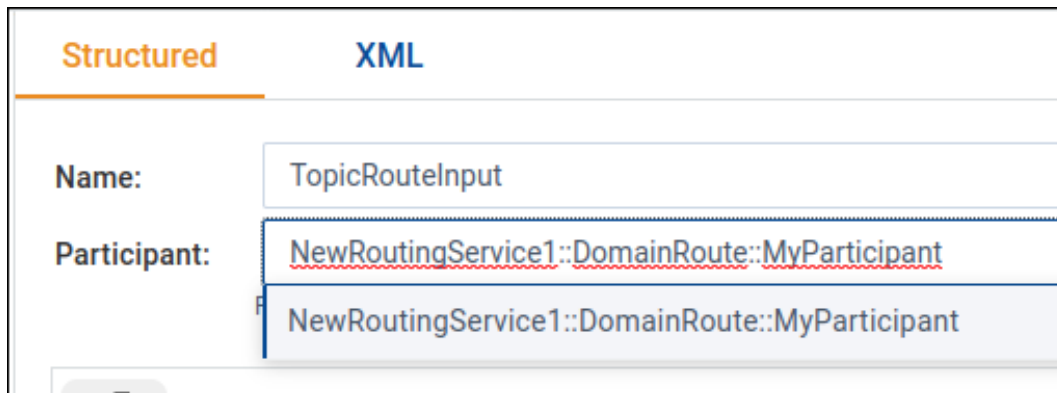
See [Route Inputs/Outputs](#) in the *RTI Routing Service User's Manual* for more information on the configuration available for a Route Inputs/Outputs.

A TopicRoute is a special kind of Route that allows defining mapping between DDS topics only.





System Designer will display the Participants defined in the Routing Service view if you press the Key Down in the *participant* field, or you start typing for suggestions.



See [Topic Route Inputs/Outputs](#) in the *RTI Routing Service User's Manual* for more

information on the configuration available for a Topic Route Inputs/Outputs.

- **Auto Routes & Auto Topic Routes:** An Auto Route defines a set of potential **Routes**, with single input and output, both with the same registered type and stream name. A Route can eventually be instantiated when a new stream is discovered with a type name and a stream name that match the filters in the **AutoRoute**. When this happens, a Route is created with the configuration defined by the AutoRoute.

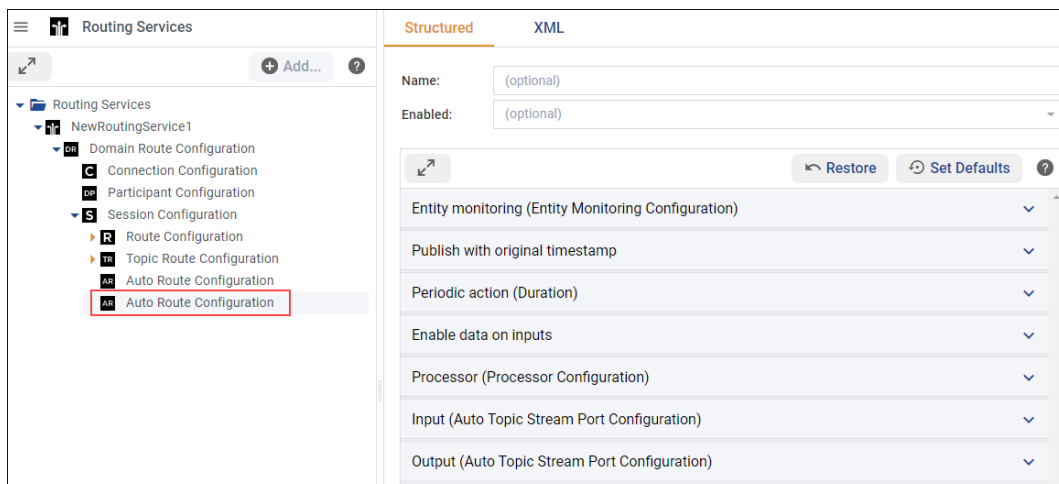
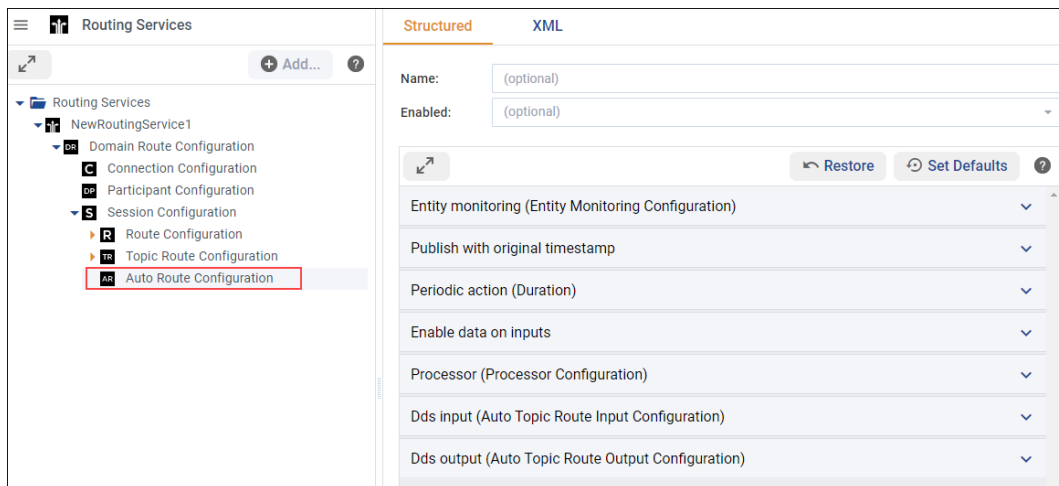
The generated Route has a name constructed as follows:

```
[auto_route_name]@[stream_name]
```

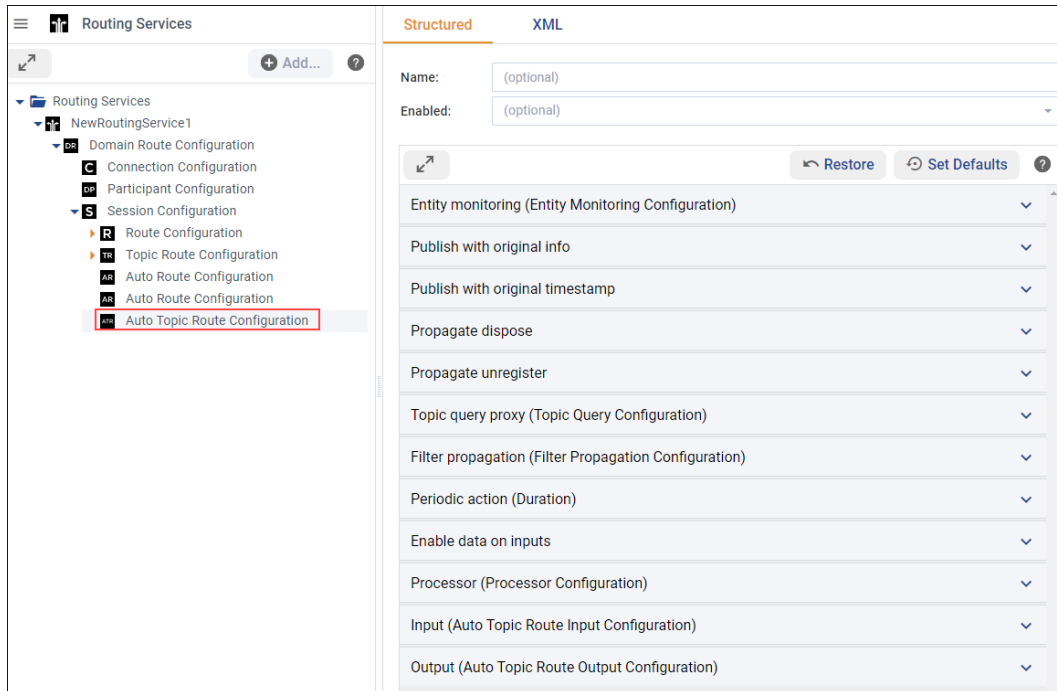
where *[auto_route_name]* represents the name of the AutoRoute and *[stream_name]* the name of the matching stream.

DDS inputs and outputs within an AutoRoute are defined by clicking on the **Auto Route (DDS inputs/outputs)**, whereas Input and outputs from other data domains are defined by clicking on the **Auto Route (non-DDS inputs/outputs)**.

An **AutoTopicRoute** is a special kind of **AutoRoute** that defines a mapping between two DDS domains.

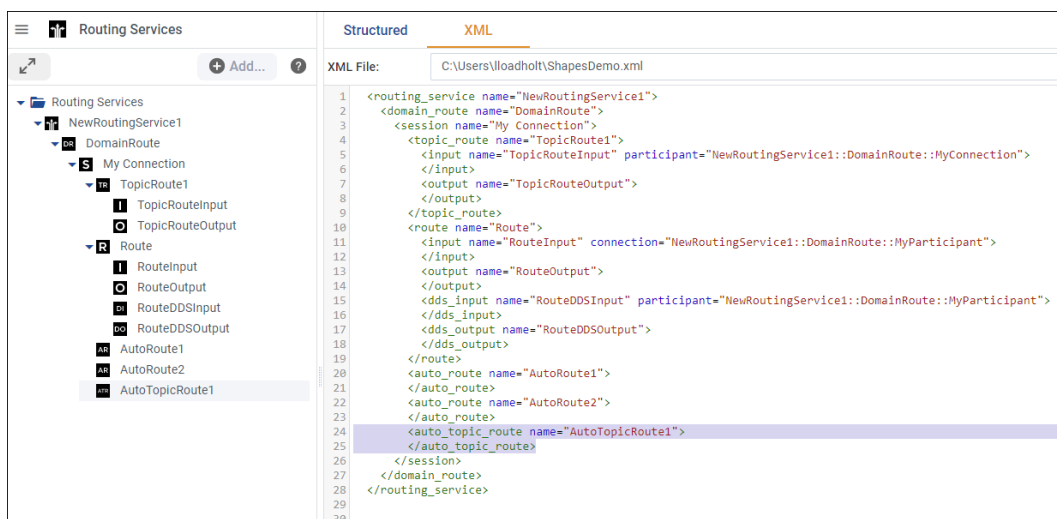


See [Auto Route](#) in the *RTI Routing Service User's Manual* for more information on the configuration available for an Auto Route.



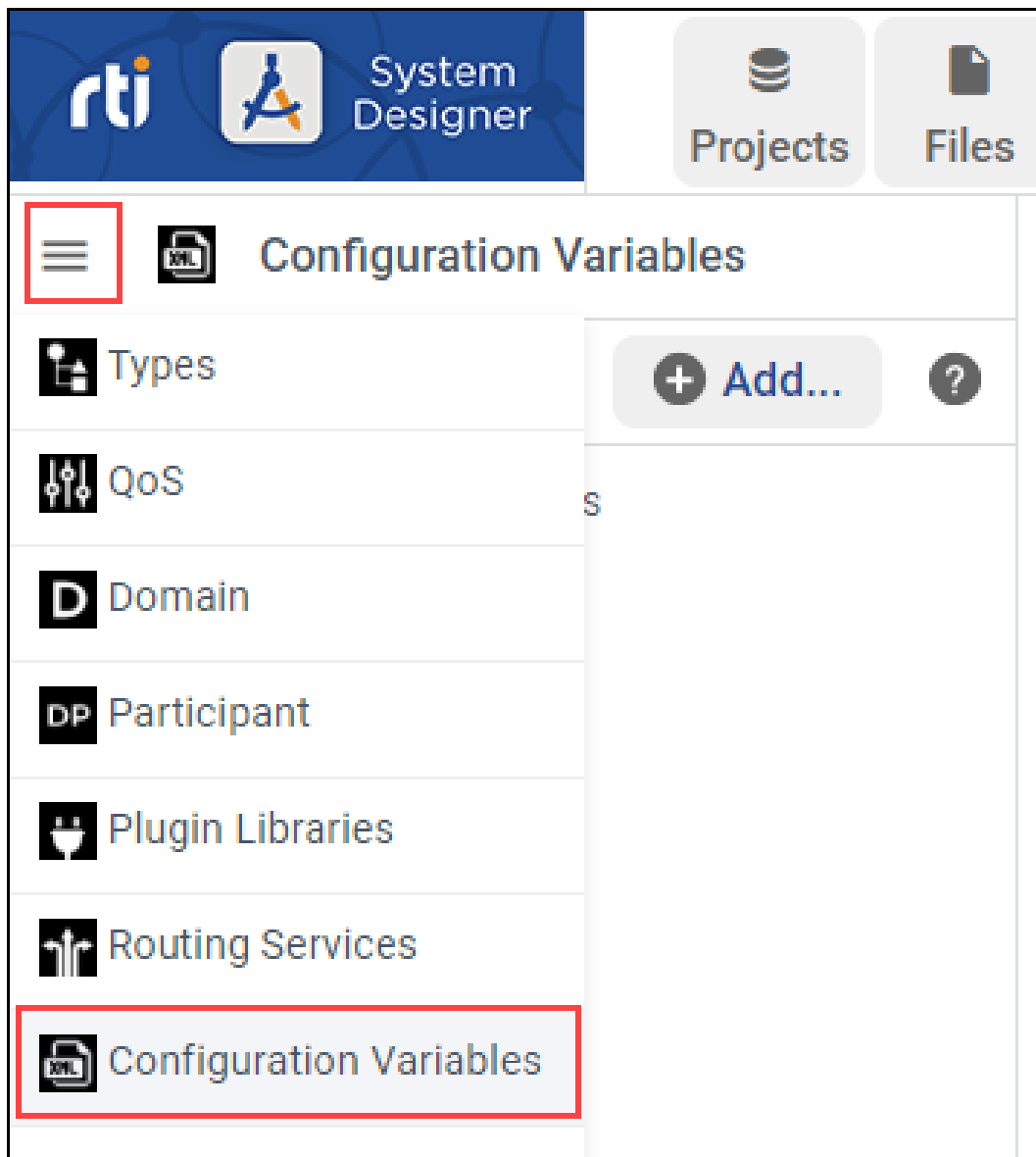
See [Auto Topic Route](#) in the *RTI Routing Service User's Manual* for more information on the configuration available for an Auto Topic Route.

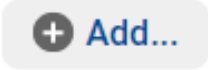
The XML view allows you to check the XML results and manually edit each XML file:

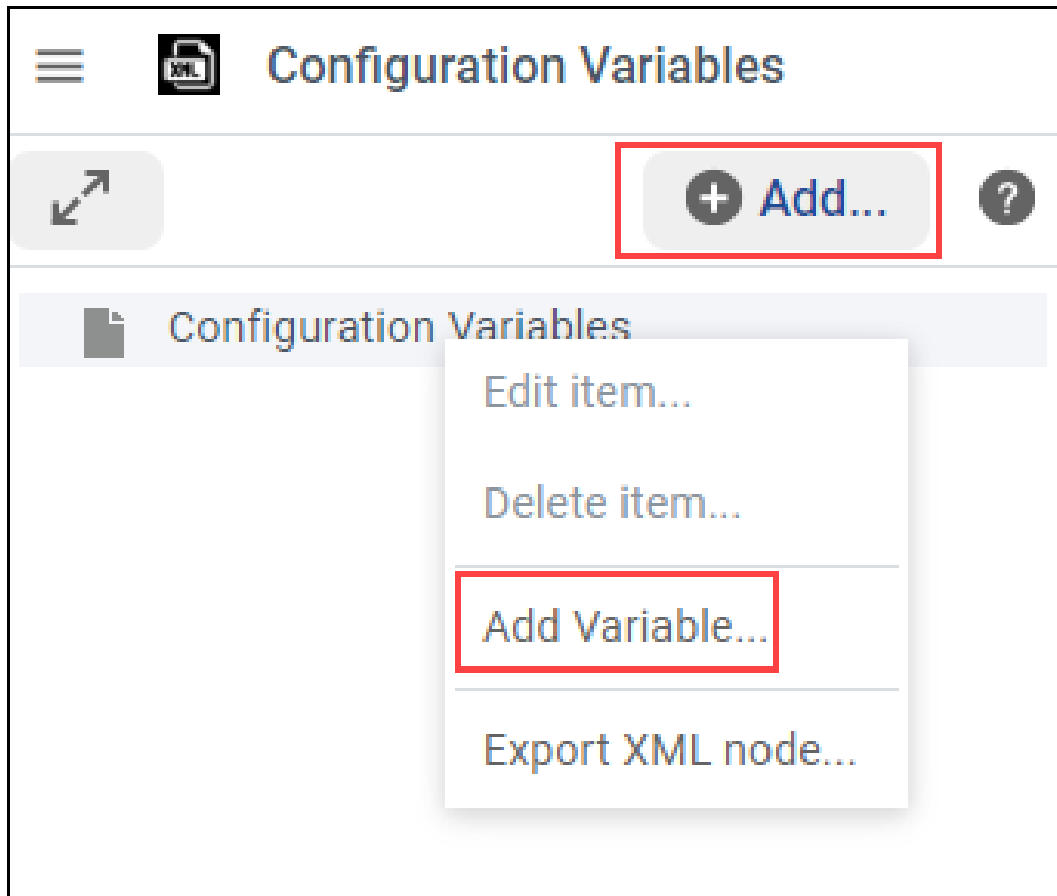


2.4.7 Define User Environment variables

The **Configuration Variables** view allows you to assign default values to the XML variables.



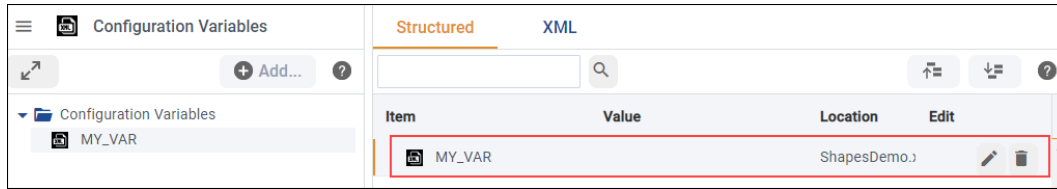
To add a new Configuration Variable to your model, click the  button while Configuration Variables is selected or right-click Configuration Variables in the tree and select **Add Variable...**:



A pop-up window will appear, where you can enter the Configuration Variable name, and the value of such variable:

The image shows a dialog box for adding a configuration variable. It has three main sections: "Name:", "Value:", and "File:". The "Name:" section has a text input field containing "(required)". The "Value:" section has a text input field containing "(optional)". The "File:" section has a dropdown menu showing "ShapesDemo.xml" and a "New..." button. At the bottom of the dialog box, there are two buttons: "Cancel" and "OK".

When a Configuration Variable is added, you can see them in the Structured view:



2.4.8 Using Fully Qualified Names when Referring to Objects

When referring to another type, you can either use the target data type without a namespace (in this case, the search will occur in the same module you are working on), or you must specify the full namespace of the target object. In particular, **partial namespace qualification is not supported because it is naturally ambiguous.**

For example, suppose you have this type (using IDL syntax):

```
module MyModule {
    module AnotherModule {
        struct BaseType {
            short x;
            short y;
        };
    }
}
```

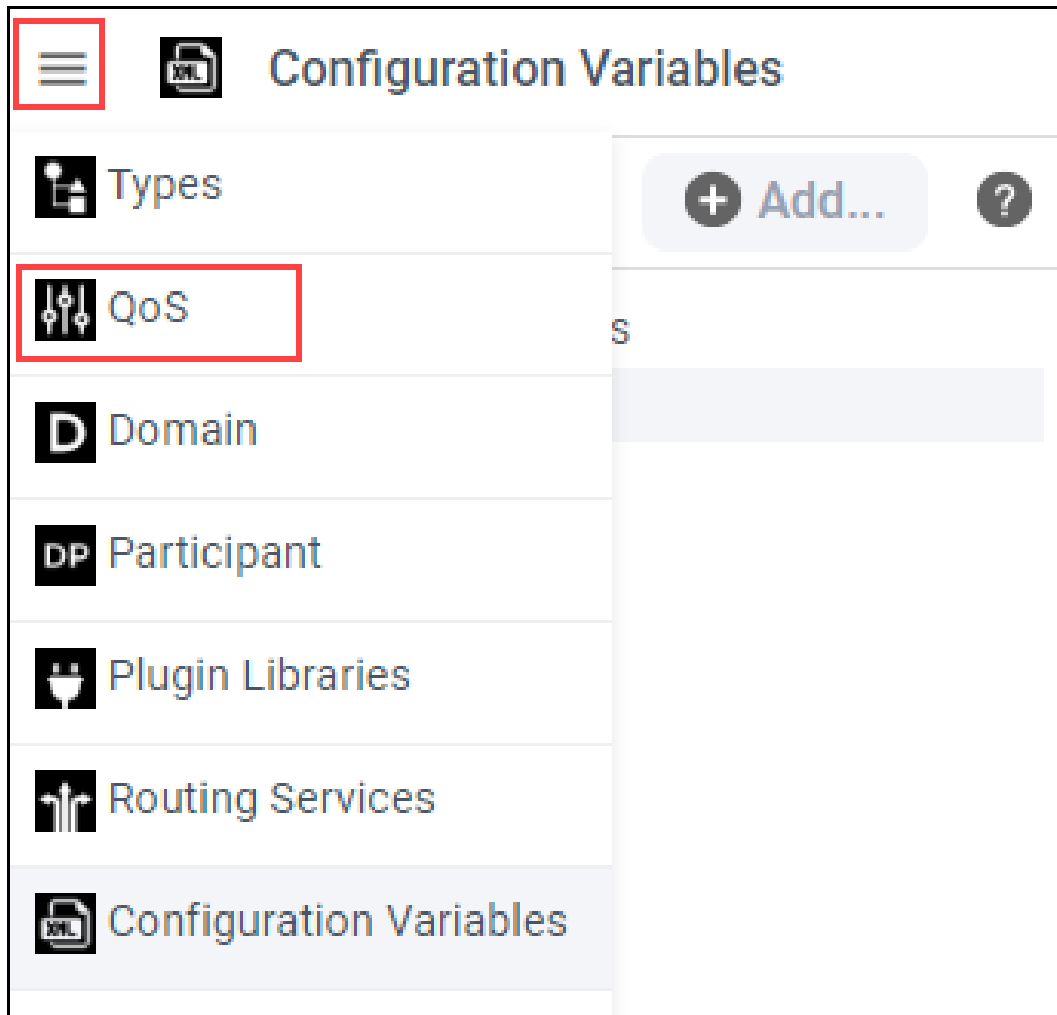
Then suppose you want to create another structure inside `AnotherModule` that extends `BaseType`. You have two choices:

- Use just the name `BaseType` to specify the base class
- Use the fully qualified name `MyModule::AnotherModule::BaseType`

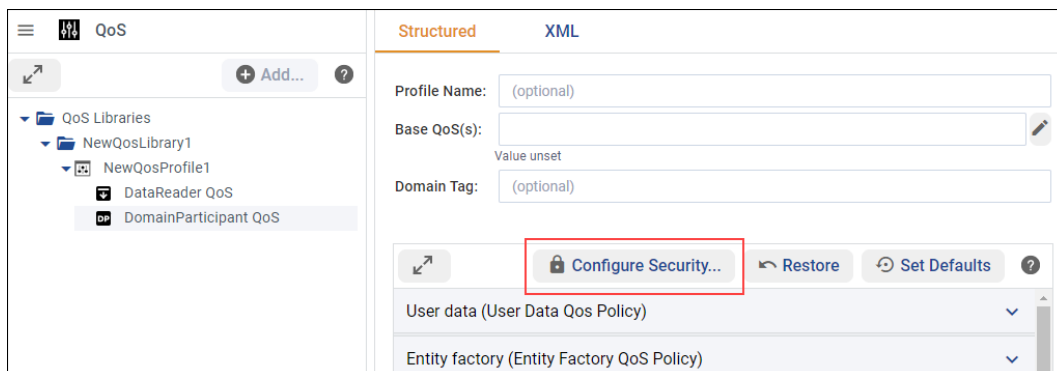
System Designer does not accept partial qualification, such as `AnotherModule::BaseType`, because it is ambiguous.

2.4.9 Configuring Security Plugins

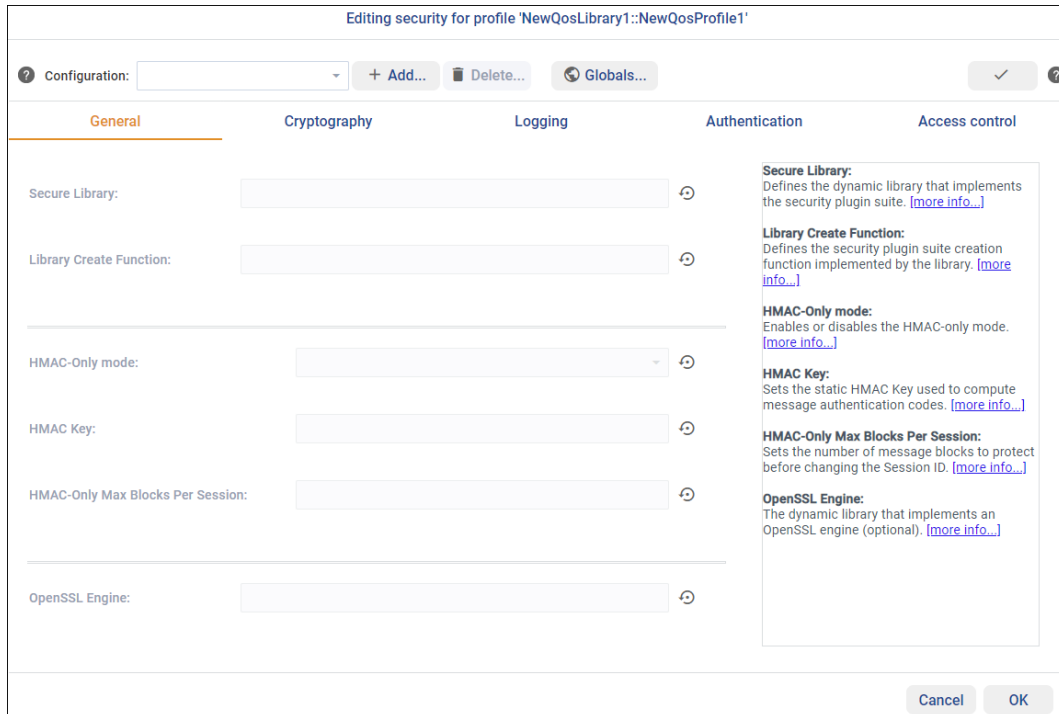
The QoS view also allows you to configure the *Security Plugins* of a *DomainParticipant*'s QoS.

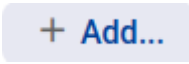


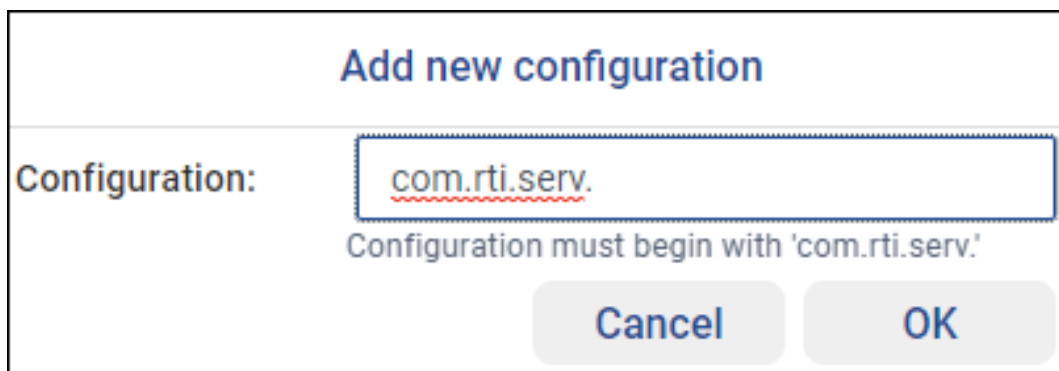
Click  in a *DomainParticipant*'s QoS.



A new dialog appears that will guide you through configuring the *Security Plugins* on the selected *DomainParticipant's* QoSs.




First, you will be asked to add a new configuration. Click  and add the name your security configuration is going to use.



Once the configuration has been added, start configuring the security properties for your newly created configuration.

Apart from allowing you to change specific security properties associated with each configuration, *System Designer* can also be used to configure security properties that are not bound to a specific configuration. These

properties can be configured by clicking on . The following dialog will open, allowing you to configure global properties such as Authentication and Access Control.

Editing security for profile 'NewQosLibrary1::NewQosProfile1'

Configuration: + Add... Delete... Globals... ?

General	Cryptography	Logging	Authentication	Access control
Secure Library:	<input type="text"/>			
Library Create Function:	<input type="text"/>			
HMAC-Only mode:	<input type="text"/>			
HMAC Key:	<input type="text"/>			
HMAC-Only Max Blocks Per Session:	<input type="text"/>			
OpenSSL Engine:	<input type="text"/>			

Secure Library:
Defines the dynamic library that implements the security plugin suite. [\[more info...\]](#)

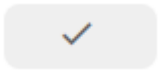
Library Create Function:
Defines the security plugin suite creation function implemented by the library. [\[more info...\]](#)

HMAC-Only mode:
Enables or disables the HMAC-only mode. [\[more info...\]](#)

HMAC Key:
Sets the static HMAC Key used to compute message authentication codes. [\[more info...\]](#)

HMAC-Only Max Blocks Per Session:
Sets the number of message blocks to protect before changing the Session ID. [\[more info...\]](#)

OpenSSL Engine:
The dynamic library that implements an OpenSSL engine (optional). [\[more info...\]](#)

Tip: The dialogs shown above can be used to check whether the security configuration is valid. To validate the configuration, click .

Security Global Settings

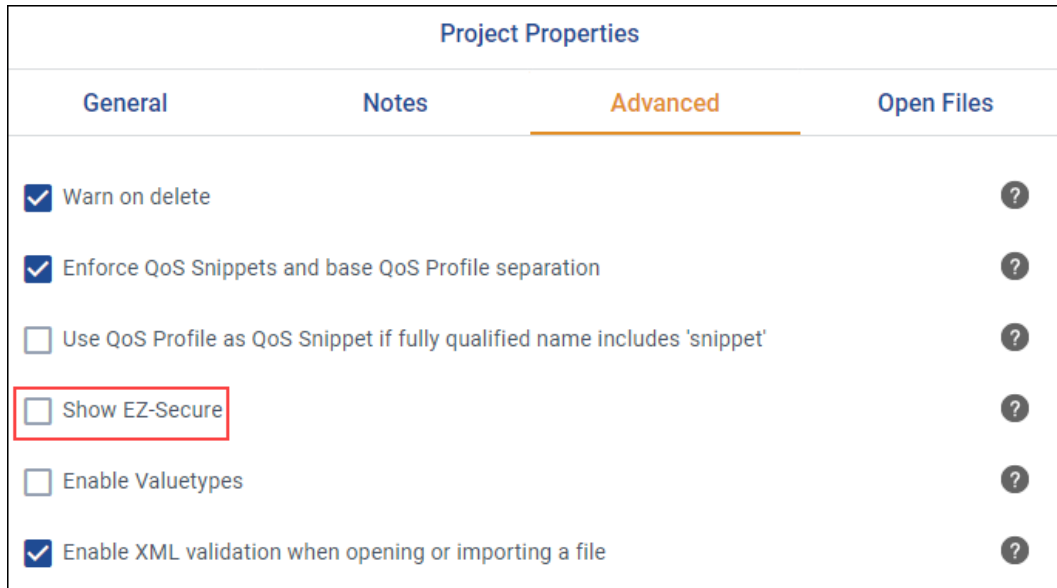
?

Authentication	Access Control	Participant Trust
<p>CA Certificate File: <input type="text"/> <input type="button" value="Browse..."/></p> <p><small>Value is unset and no default value is defined</small></p> <div> <p>ERROR: Missing Identity CA Certificate</p> <p>You have not specified the Identity CA Certificate that is used for verifying the Identity Certificates of local and remote DomainParticipants. This file is required. [Click here for more information...]</p> <p>[Continue checks...]</p> </div> <p>Private Key Password: <input type="text"/> <input type="button" value="Browse..."/></p> <p><small>Value is unset and no default value is defined</small></p>		<p>CA Certificate File: Identity CA certificate. [more info...]</p> <p>Participant Private Key: The participant's private key. [more info...]</p> <p>Participant Certificate: Identity Certificate signed by the Identity CA. [more info...]</p> <p>Private Key Password: The password used to decrypt the private key. [more info...]</p>

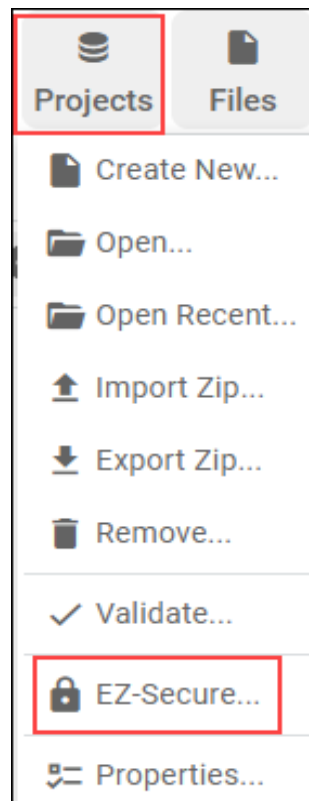
For more information on *Security Plugins*, see the [RTI Security Plugins User's Manual](#).

2.4.10 EZ-Secure

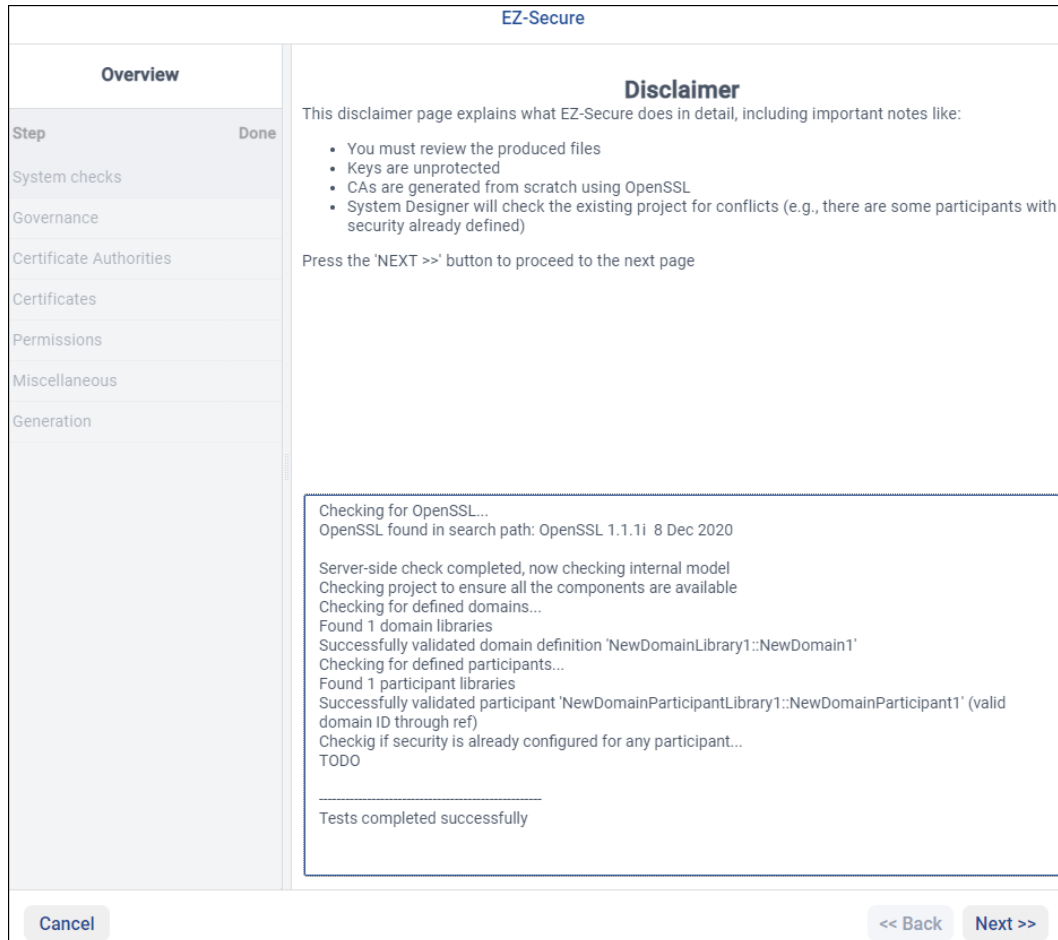
EZ-Secure is an experimental feature that can be used to automatically configure the *Security Plugins* settings. By default, it is disabled but can be enabled by selecting the *Show EZ-Secure* option in the Advanced Properties.



Once enabled, EZ-Secure will be shown in the Projects menu:



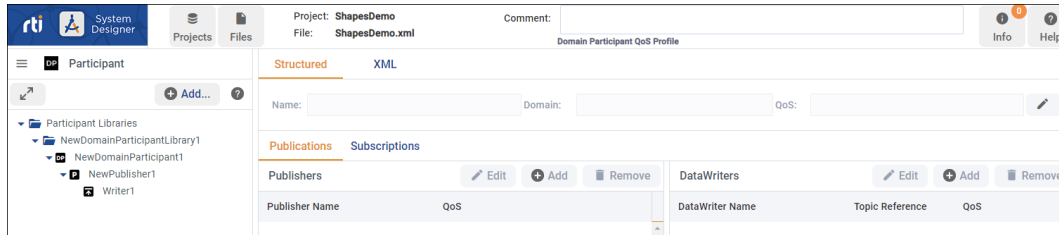
A dialog will guide you through configuring security and generating the certificates:



2.5 Toolbar

System Designer's toolbar:

- Allows you to manage your projects
- Allows you to handle existing XML files in your system
- Shows the name of the current project and last used XML file
- Has a box to add a comment to the selected XML object
- Shows general information and warnings via the Info button
- Has a Help menu

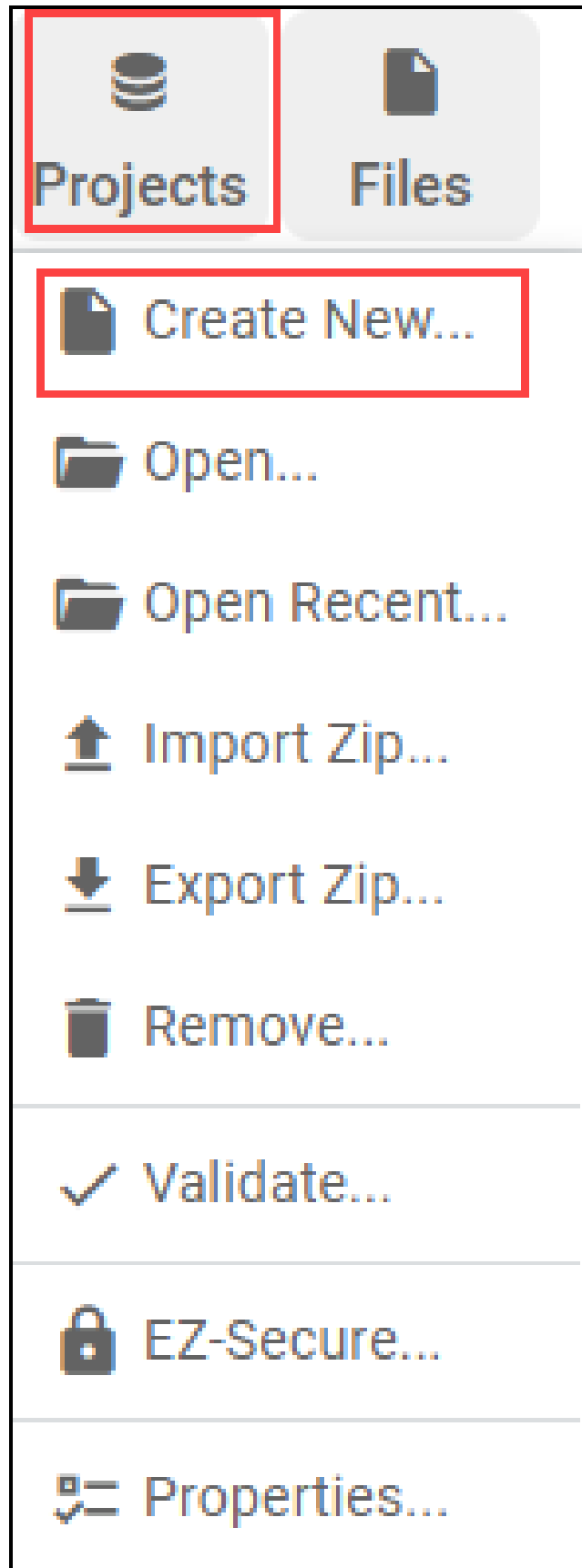


2.6 Project Management

The Projects menu in the toolbar helps you with all operations related to project management. With this menu you can perform the actions described in the following sections.

2.6.1 Create a New Project

To create a brand new project, select “Create New...” from the Projects Menu:



A pop-up will be presented where you can choose the name of the project and select where the project will be saved on your filesystem. You can select if you want to create an XML file now or load/create one later.

You can also choose to autosave the project, see information like the time and date of the last time the project was saved, and view the *System Designer* version and the *Connect DDS* version you are creating a file for.

New Project

General Notes Advanced Open Files

Project Name:

Only alphanumeric characters, underscores '_', dots '.', and hyphens '-' are allowed

Project Path: ...

☒ Automatically create an empty XML file ?

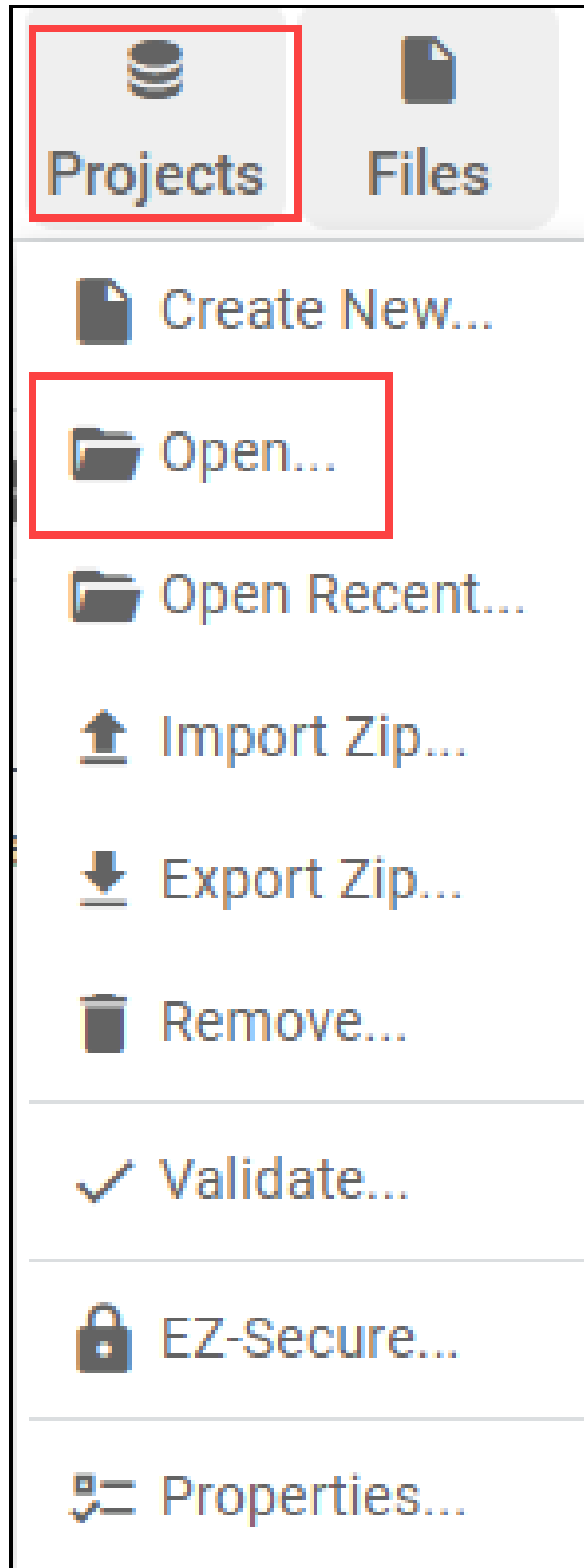
☒ Autosave project ?

Project Created: 14/12/2022 @ 14:36:39

Project Last Saved: 14/1/2023 @ 10:28:27

2.6.2 Open an Existing Project

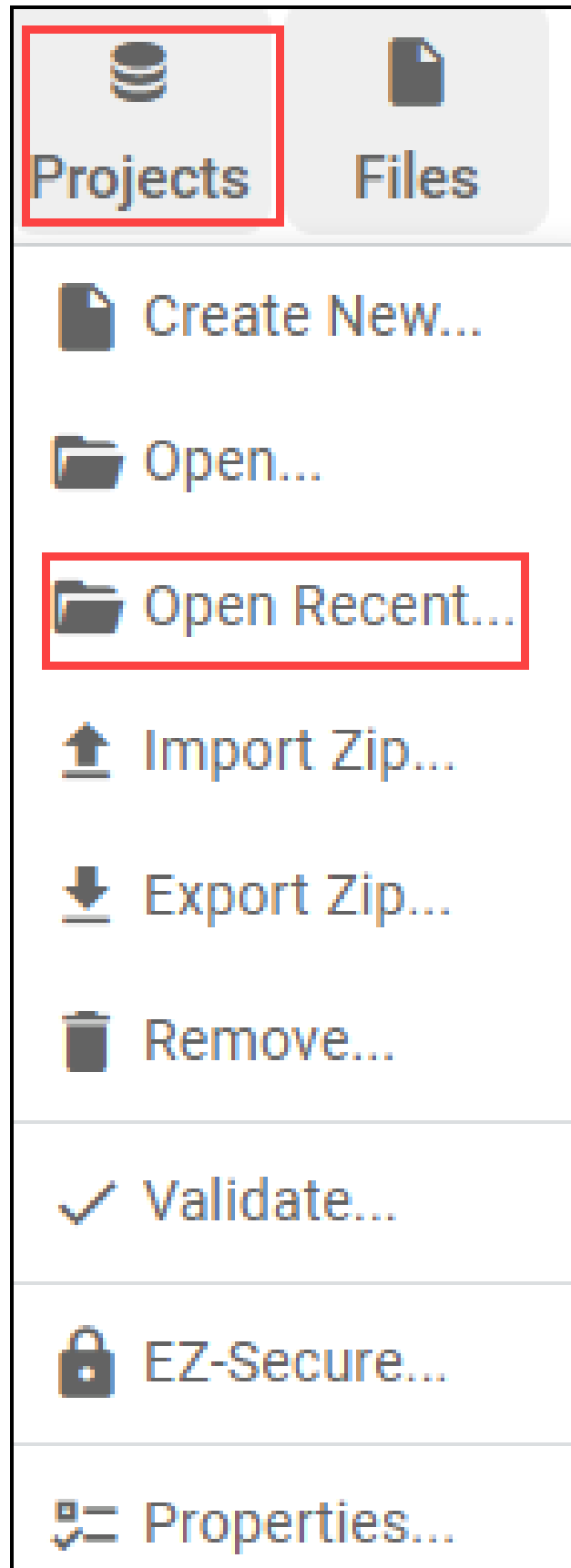
To open an existing project, select the **Open...** option in the Project menu:



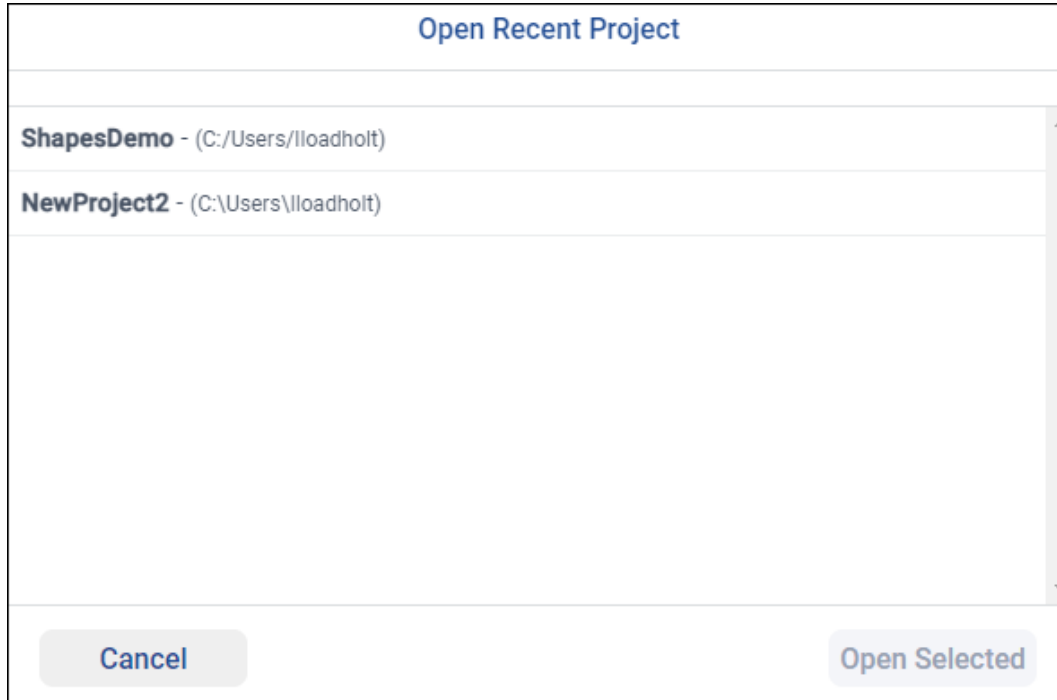
A file browser will be presented. You can use it to find your `.rtisdproj` file.

2.6.3 Open a Recent Project

To open a recent project, select **Open Recent** from the Projects Menu:

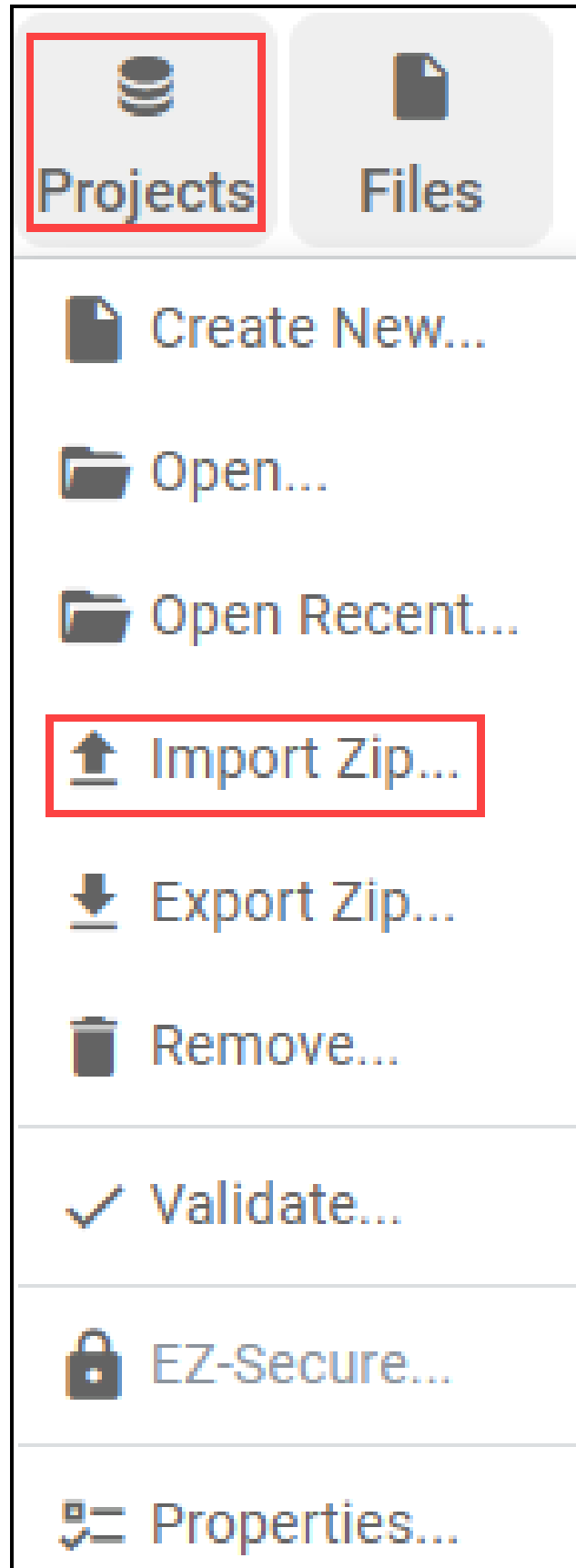


You will see a list of recently opened projects. Select one and click **Open Selected**:

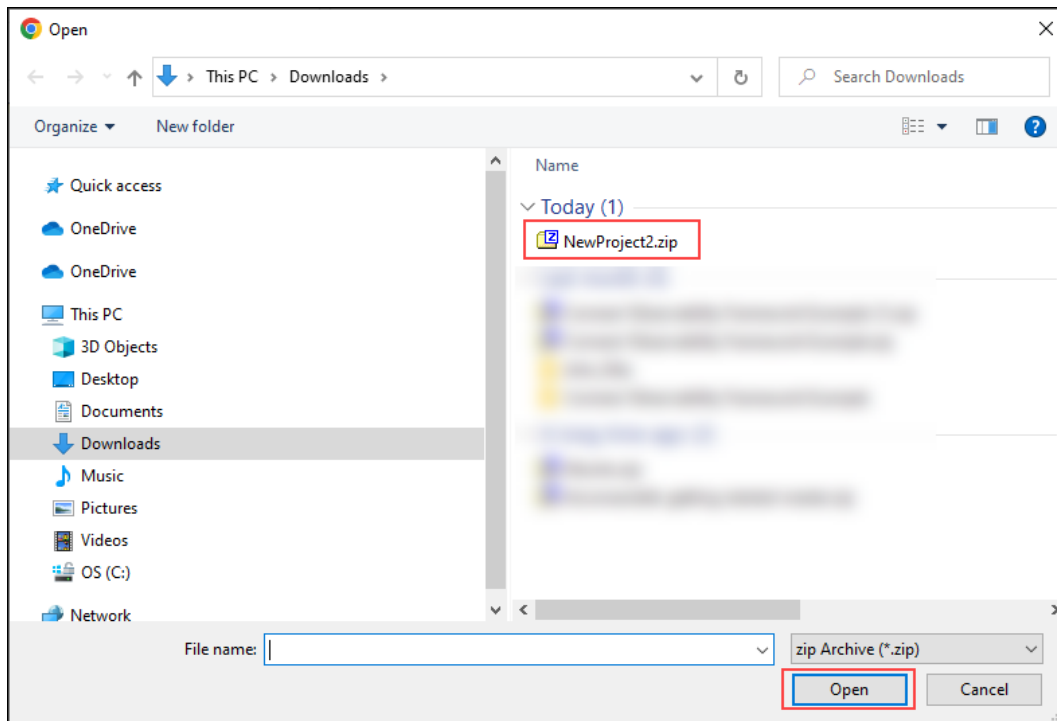


2.6.4 Import a Project from a Zip File

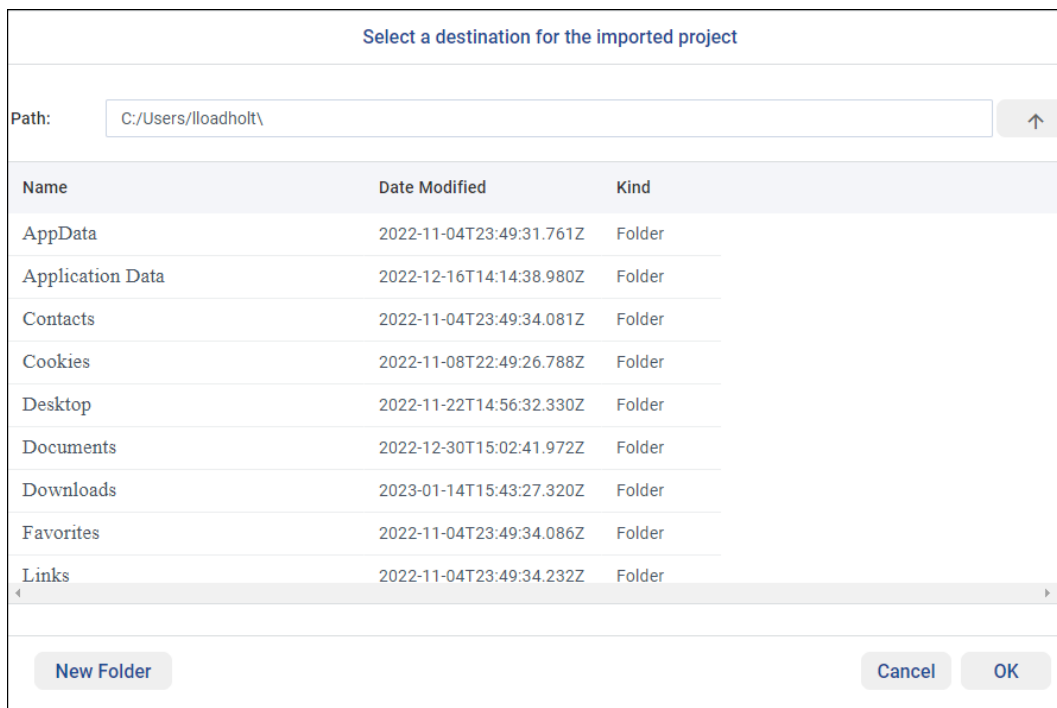
To import a project from a Zip file, select **Import Zip...** from the Projects menu:



Then use the file browser to find the Zip file:



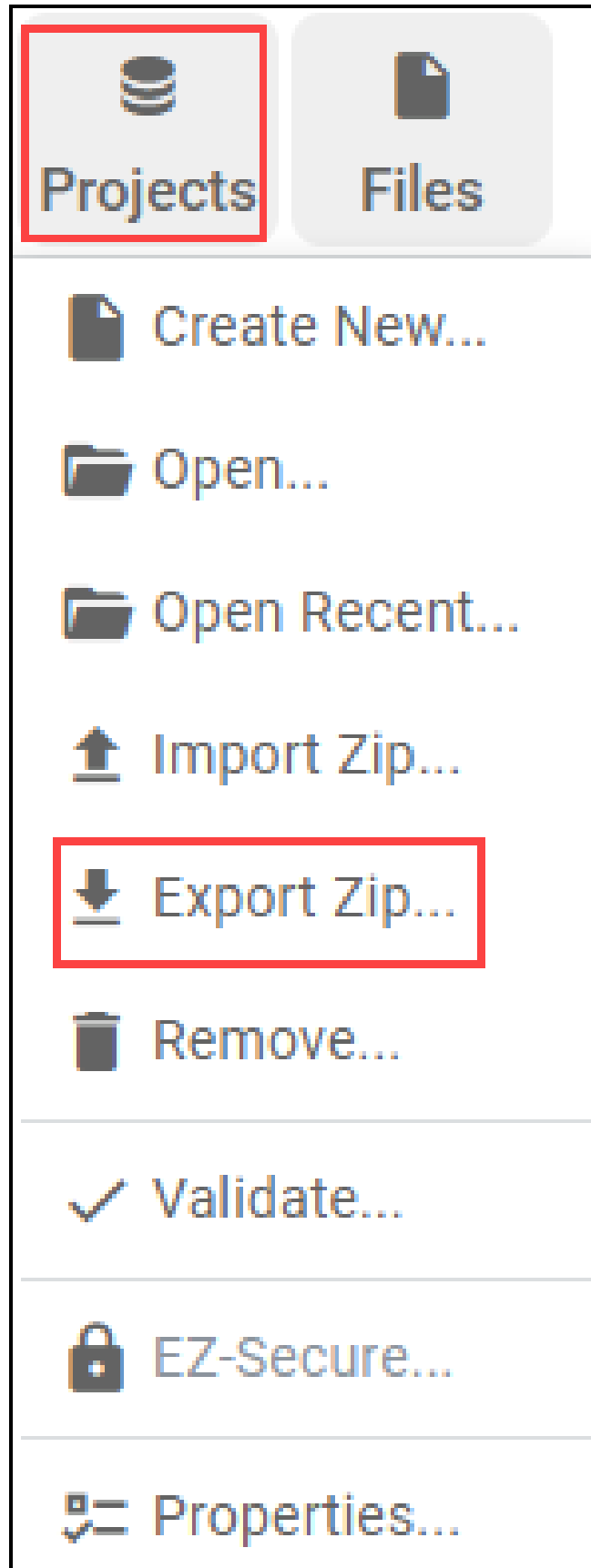
Once you find it, select and open the file. A new dialog will ask you to select a destination directory in which to unzip the file and save the imported project:



Press **OK** to import and open the project in *System Designer*.

2.6.5 Export a Project to a Zip File

To export a project to a Zip file, select **Export Zip...** from the Projects menu:

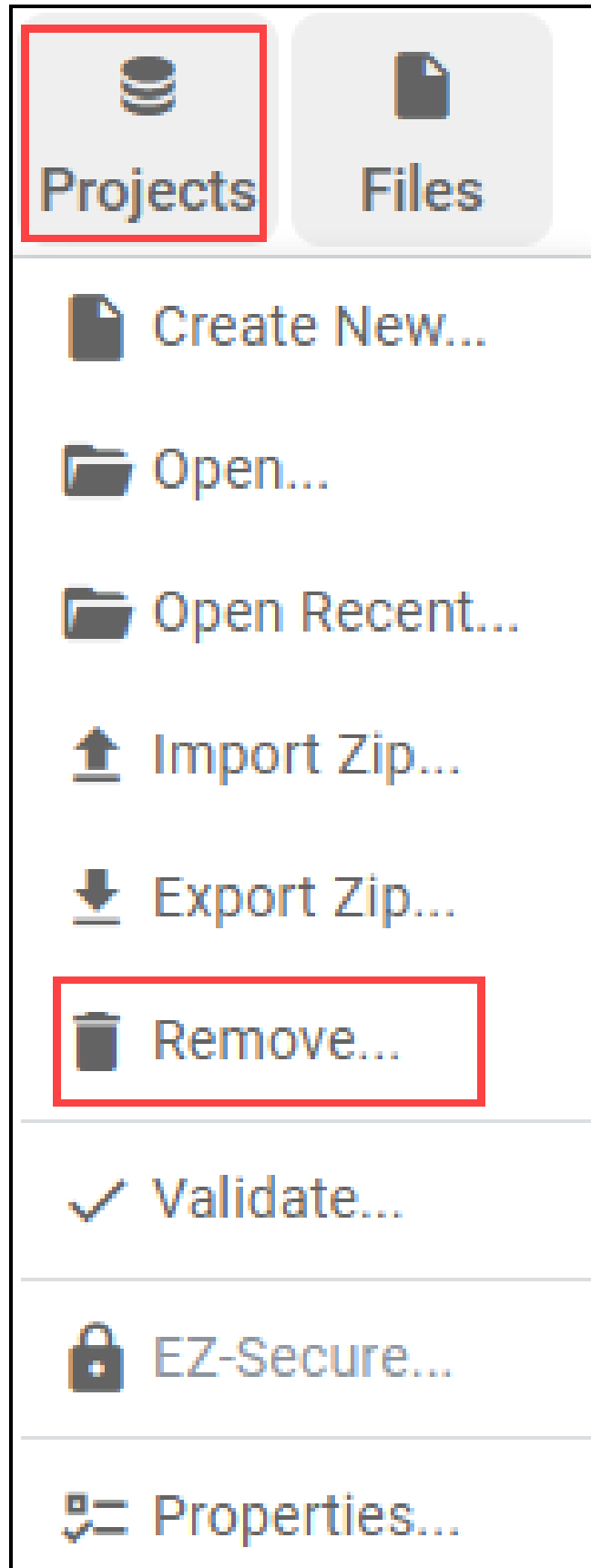


A Zip file will be downloaded in your default download directory. The Zip file will contain the project file and all the XML files used in that project.

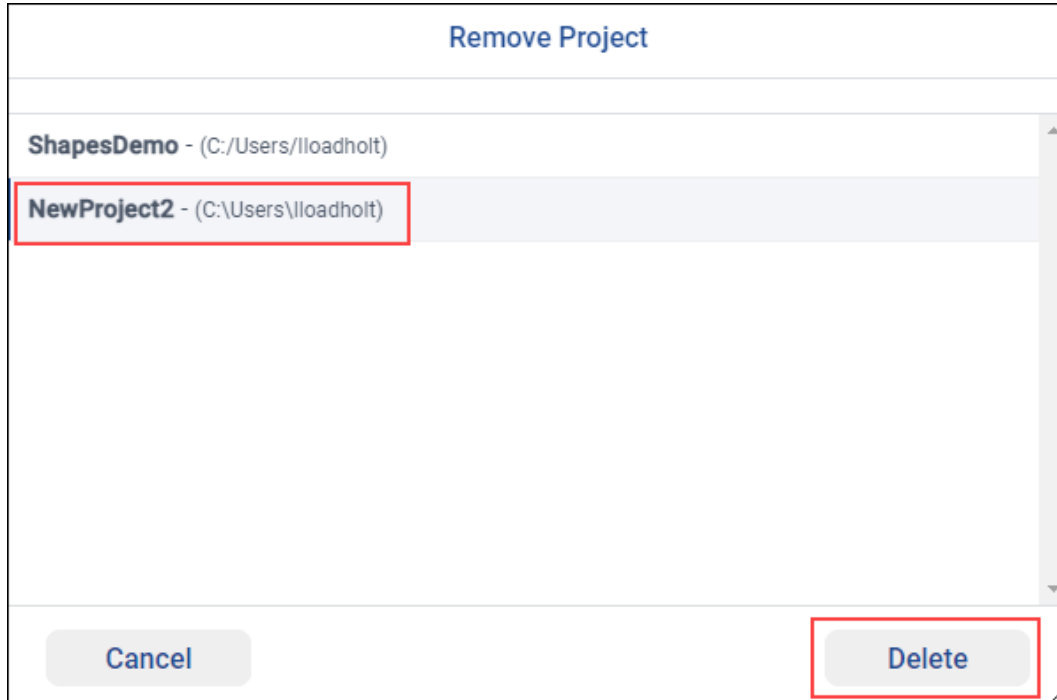
Note: Some browsers (such as Safari) automatically unzip the folder. Disable this behavior or zip the directory again for sharing.

2.6.6 Remove a Project from the List of Recent Projects

To remove a project from the list of recent projects, select **Remove...** from the Projects menu:



Then select the project you want to remove from the list and select **Delete**:

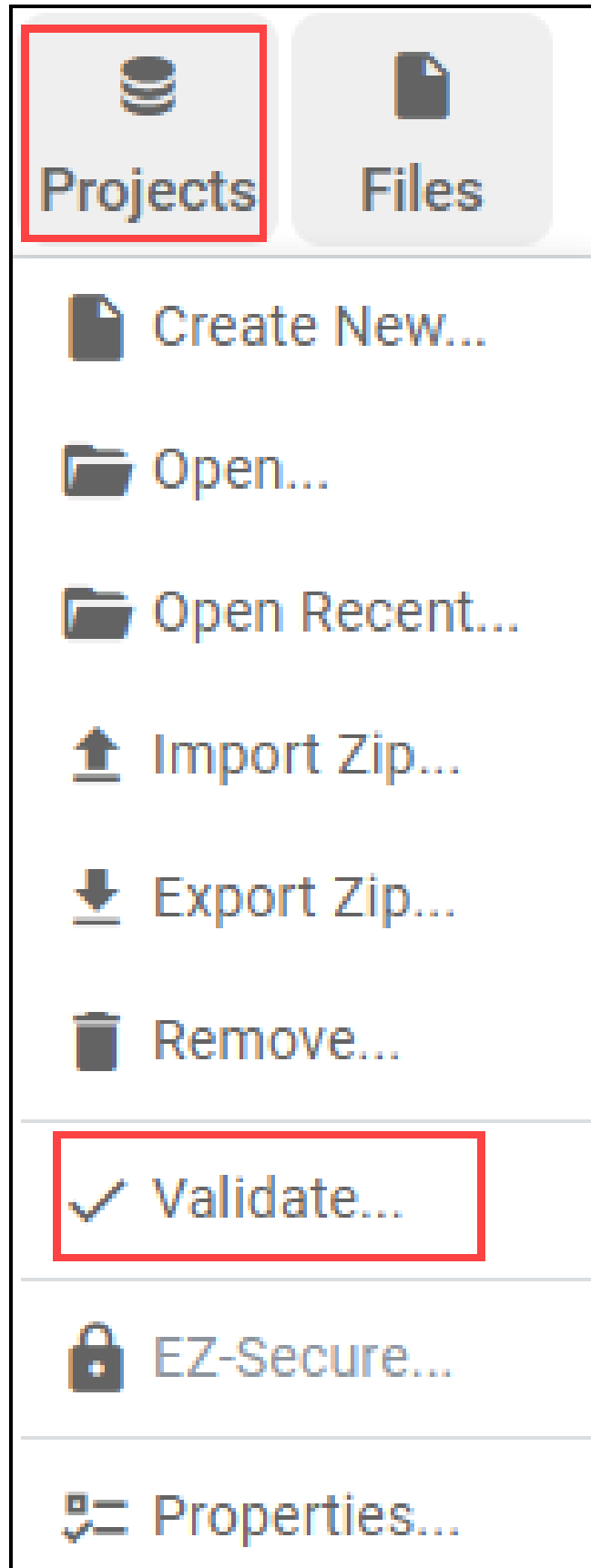


2.6.7 Validate Project

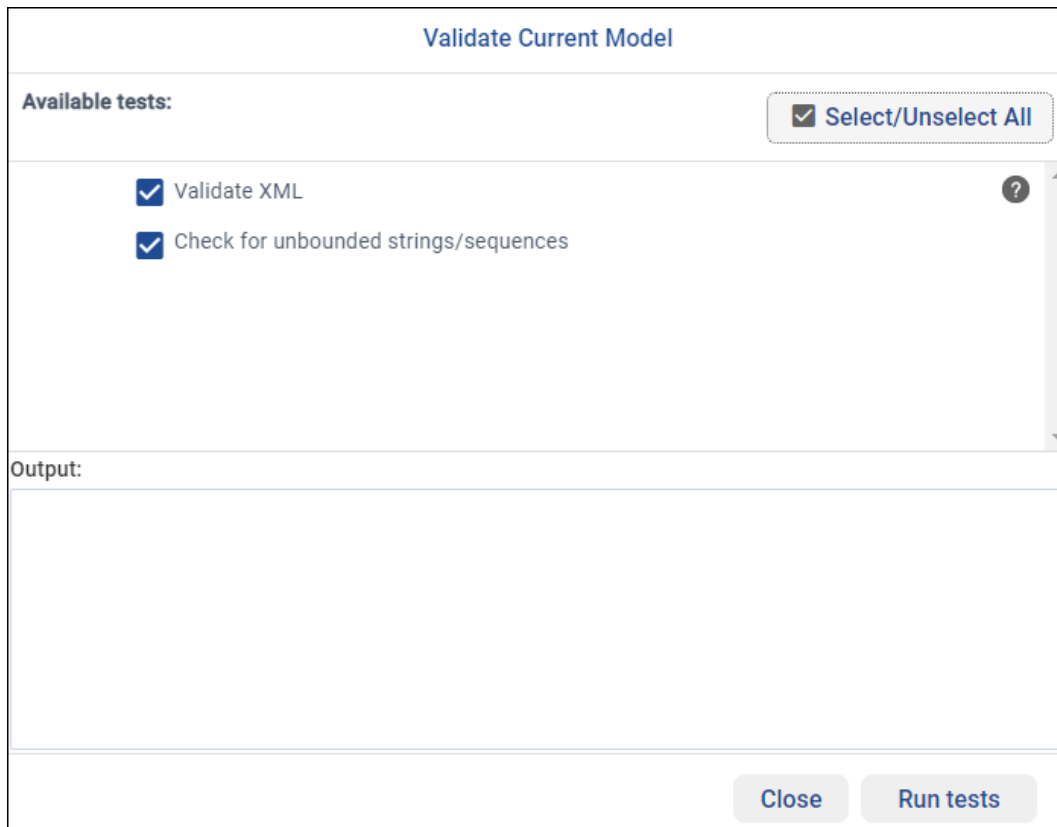
Two validation tests are currently available:

- **Validate XSD** This test uses the tool **xmllint**. See *Prerequisites* for more information.
- **Check for unbounded strings/sequences** This test looks in the current project to see if there are any data types that are being used (in Topics used by DataReaders or DataWriters) that use unbounded sequences or strings. If so, this test prints a warning reminding you of the limitations (and potential problems) of using unbounded sequences or strings. In future releases, *System Designer* will also analyze each Entity's QoS to ensure the required resources are correctly declared.

To validate a project, select **Validate...** from the Projects Menu.

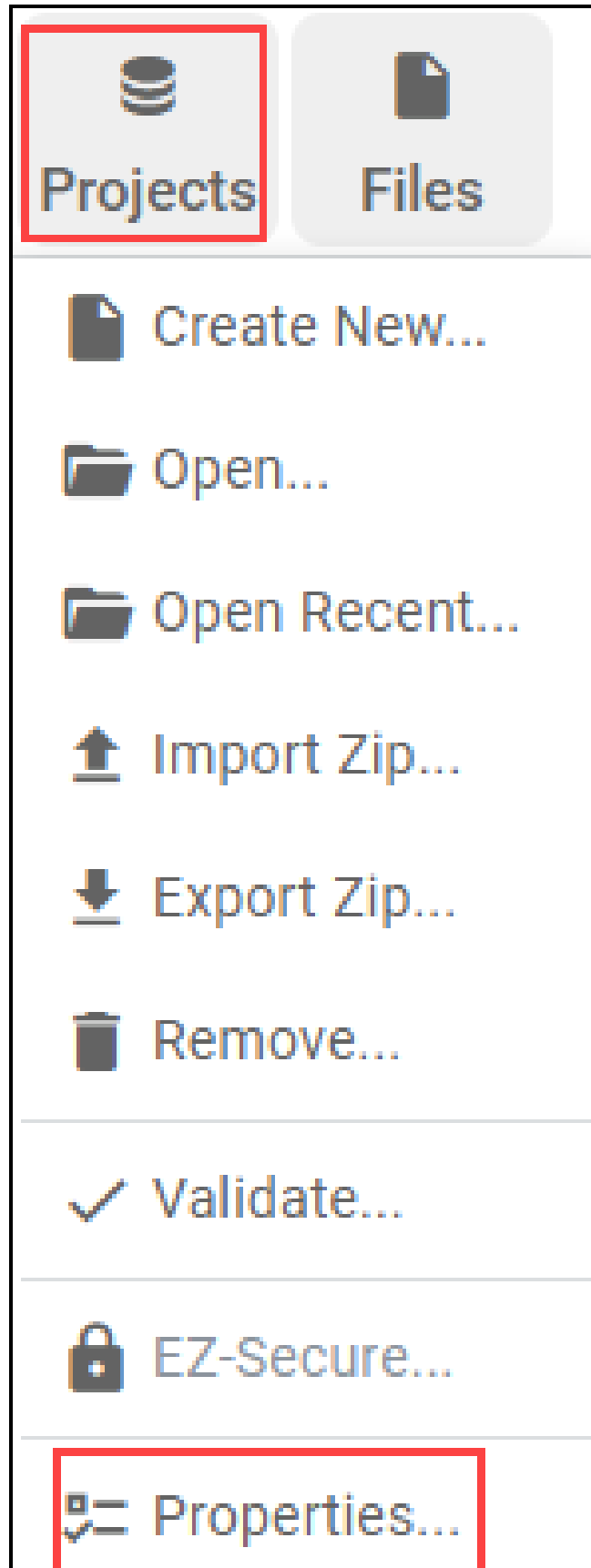


This will open a Validate dialog. You can run some or all of the available tests:



2.6.8 Project Properties

To see the properties of the current project, select Properties from the Projects menu:



This will show a window with four tabs.

- The **General** tab shows the name of the project, where it is saved, if autosave is enabled or not, the last time the project was saved, and the *System Designer* and *Connex DDS* version numbers:

The screenshot shows the 'Project Properties' dialog box with the 'General' tab selected. The dialog has four tabs: 'General', 'Notes', 'Advanced', and 'Open Files'. The 'General' tab contains the following fields and information:

- Project Name:** A text box containing 'NewProject2'. Below it, a note states: 'Only alphanumeric characters, underscores '_', dots '.', and hyphens '-' are allowed'.
- Project Path:** A text box containing 'C:\Users\lloadholt'.
- Autosave project:** A checked checkbox with a help icon (?) to its right.
- Project Created:** '14/1/2023 @ 10:40:11'.
- Project Last Saved:** '14/1/2023 @ 10:40:11'.
- System Designer Version:** '7.1.0'.
- Connex DDS Version:** '7.1.0'.
- Project Created with System Designer:** '7.1.0'.


At the bottom of the dialog are 'Cancel' and 'OK' buttons.

- The **Notes** tab allows you to add a free-form comment for the current project:

The screenshot shows the 'Project Properties' dialog box with the 'Notes' tab selected. The dialog has four tabs: 'General', 'Notes', 'Advanced', and 'Open Files'. The 'Notes' tab contains a large text area with the text 'This project is a demo'.

- The **Advanced** tab allows you to fine-tune some of *System Designer's* behavior.

Project Properties			
General	Notes	Advanced	Open Files
<input checked="" type="checkbox"/>	Warn on delete		?
<input checked="" type="checkbox"/>	Enforce QoS Snippets and base QoS Profile separation		?
<input type="checkbox"/>	Use QoS Profile as QoS Snippet if fully qualified name includes 'snippet'		?
<input type="checkbox"/>	Show EZ-Secure		?
<input type="checkbox"/>	Enable Valuetypes		?
<input checked="" type="checkbox"/>	Enable XML validation when opening or importing a file		?

Click  for more details on each setting.

- The **Open Files** tab shows which files are in use:

Project Properties			
General	Notes	Advanced	Open Files
BaselineRoot.documentationONLY.xml (built-in)			
BuiltinProfiles.documentationONLY.xml (built-in)			
NewProject2.xml			Remove from project

2.7 External XML Files

The **Files** menu in the toolbar helps you with all operations relative to file management in the currently open project:

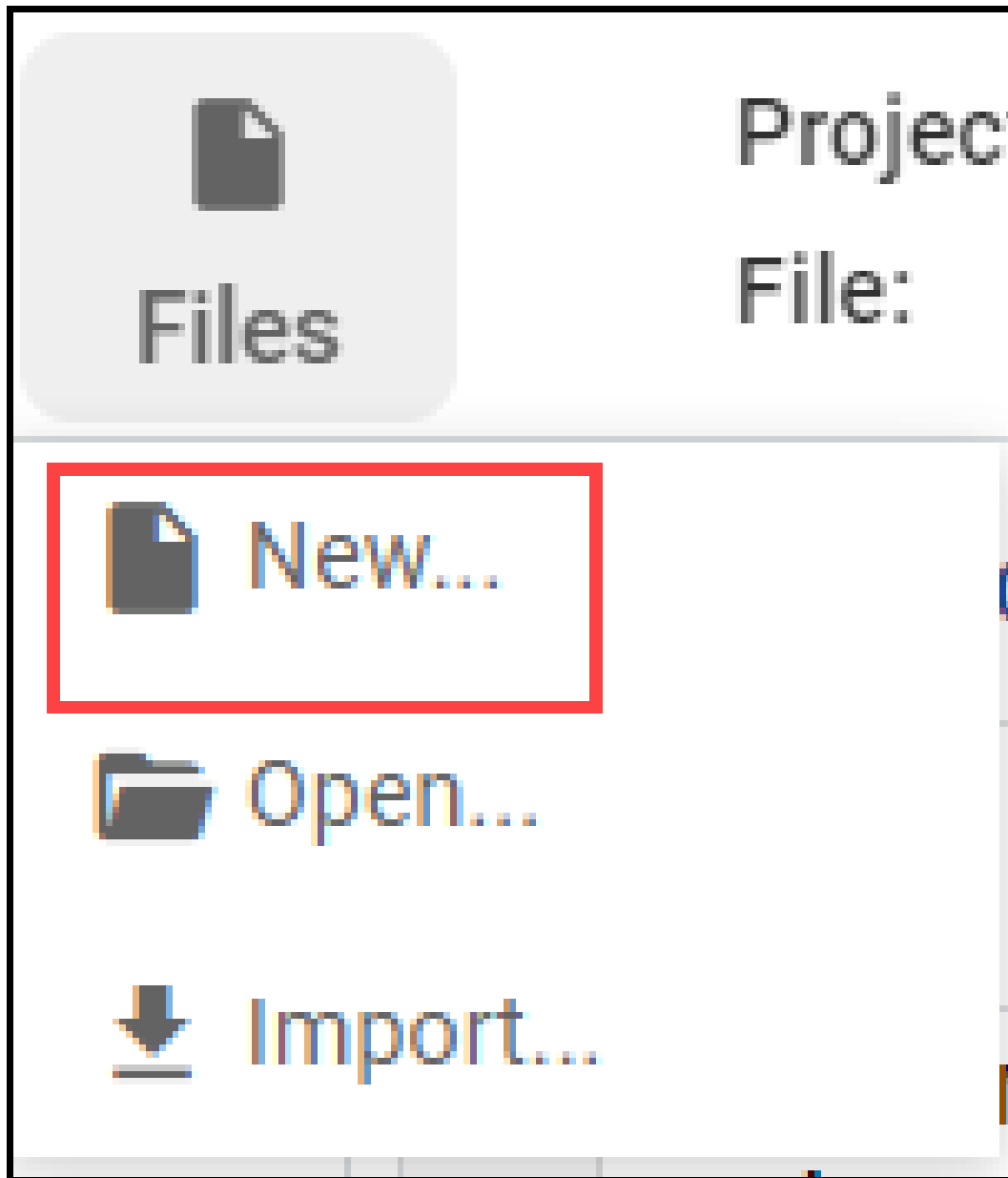
With this menu you can:

- Create a new XML file and add it to the project
- Open an existing XML file and add it to the project
- Import an existing XML file

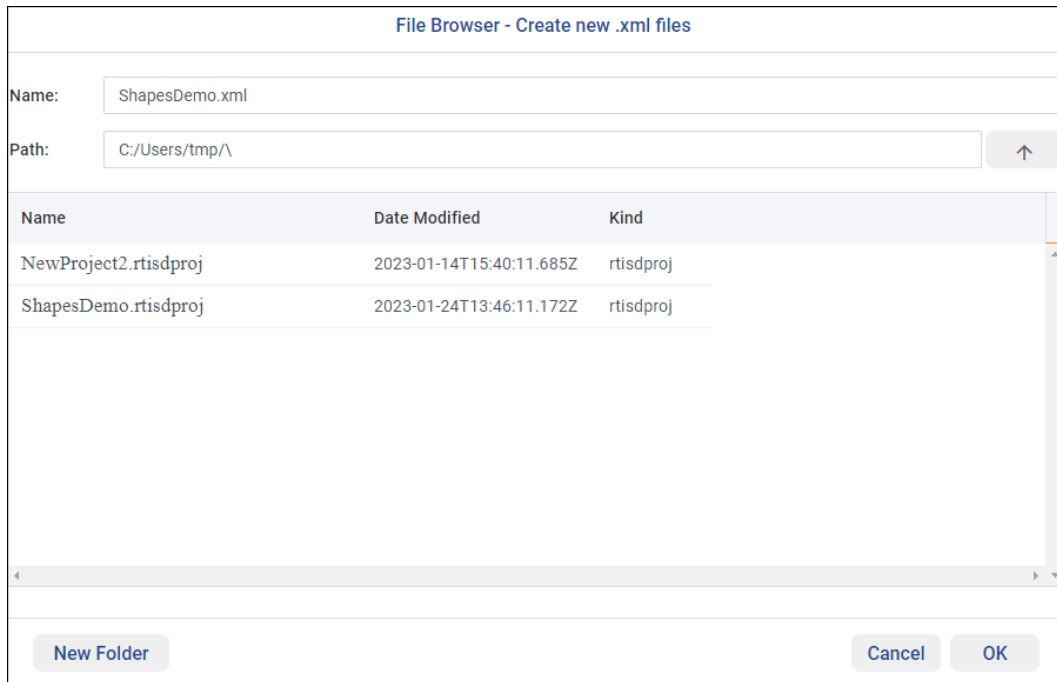
Note: Opening a file means that *System Designer* will be working (and modifying) that file directly. Importing a file helps you make a copy of that file. The original file will not be modified by *System Designer*.

2.7.1 Creating a New File

To create a new file and add it to the current project, select **New...** from the Files menu:

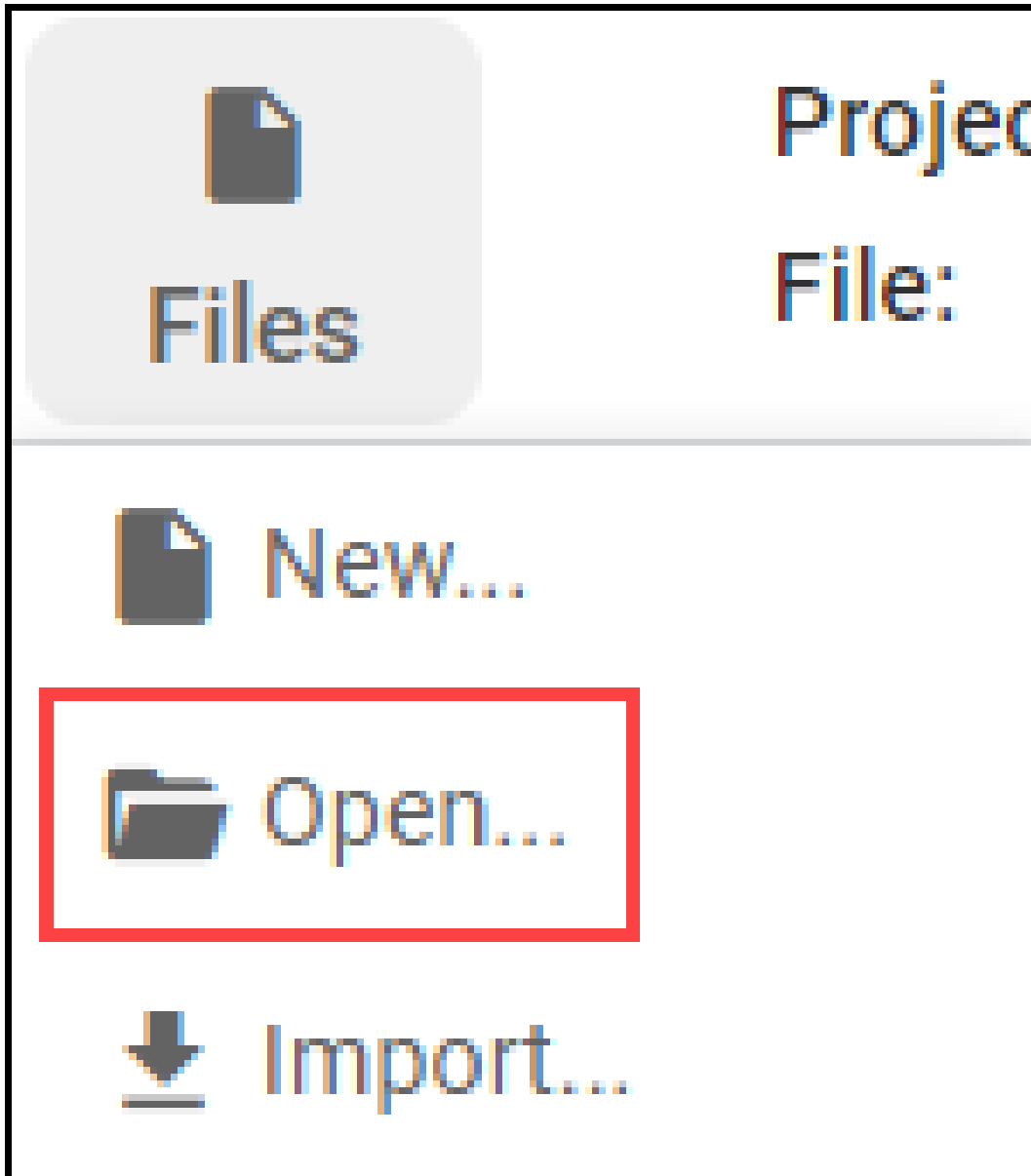


This will open a file browser to select a name and destination for the new file:



2.7.2 Open an Existing File

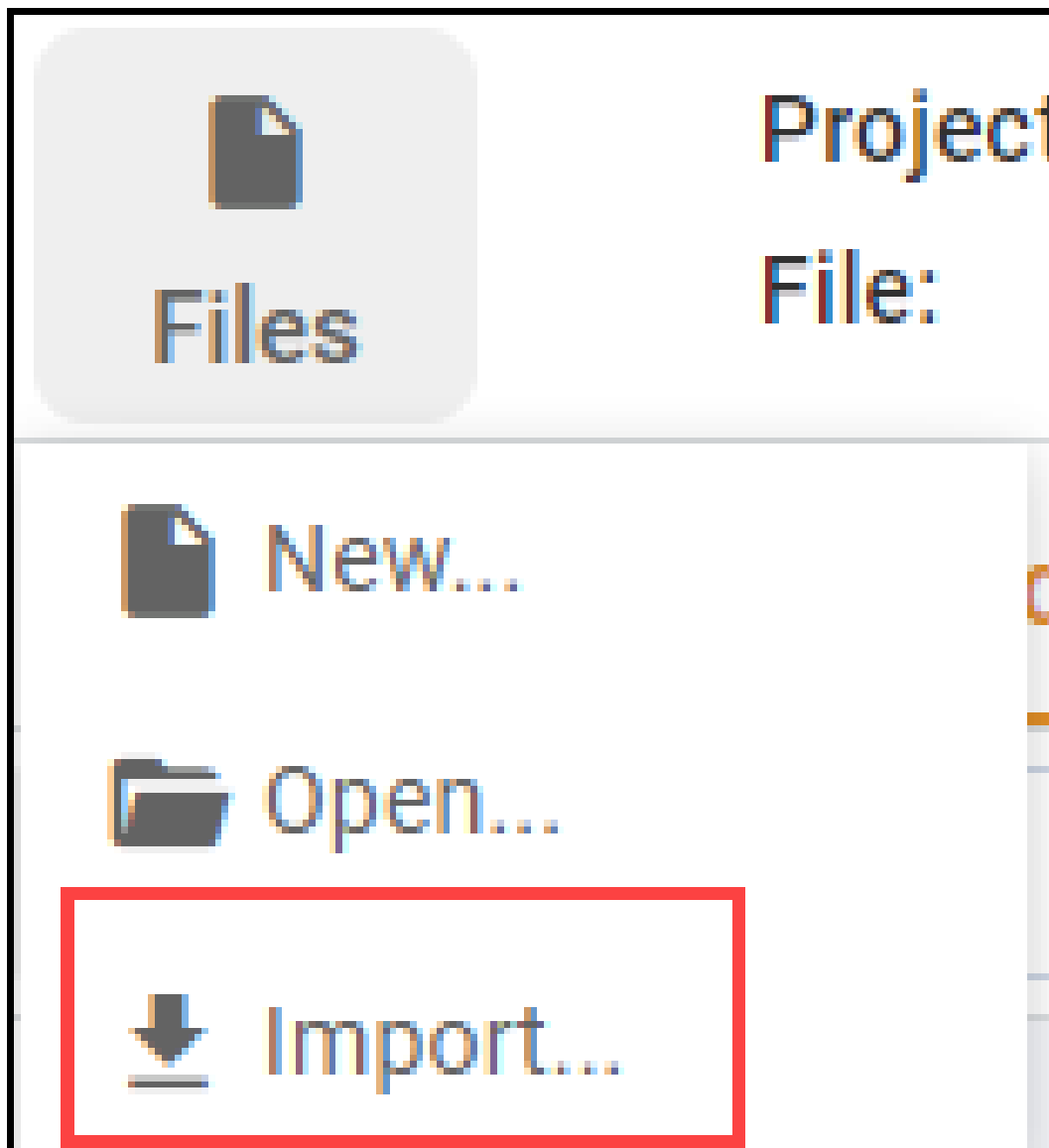
To open an existing file and add it to the current project, select **Open...** from the File menu:



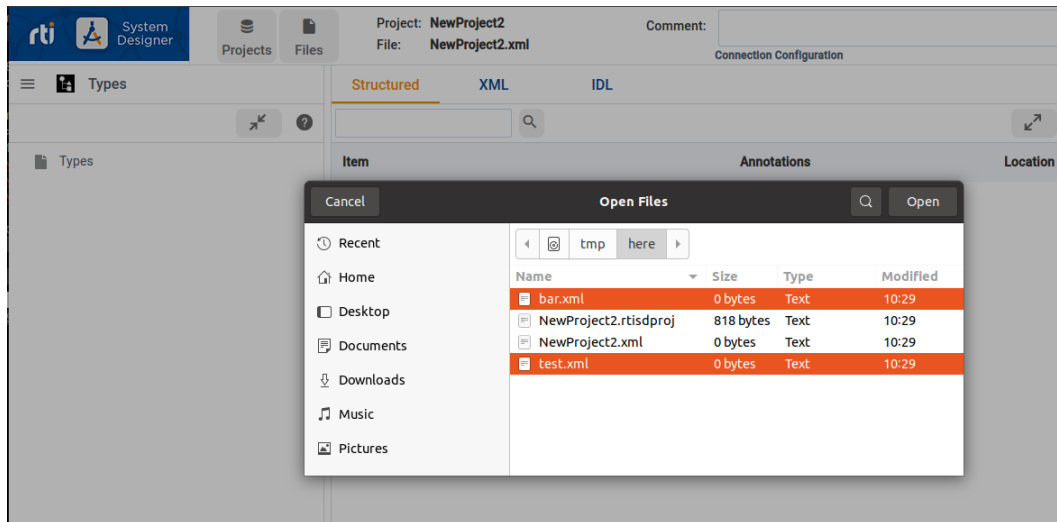
This will open a simple file browser to search for an XML file to add to the current project.

2.7.3 Import a File

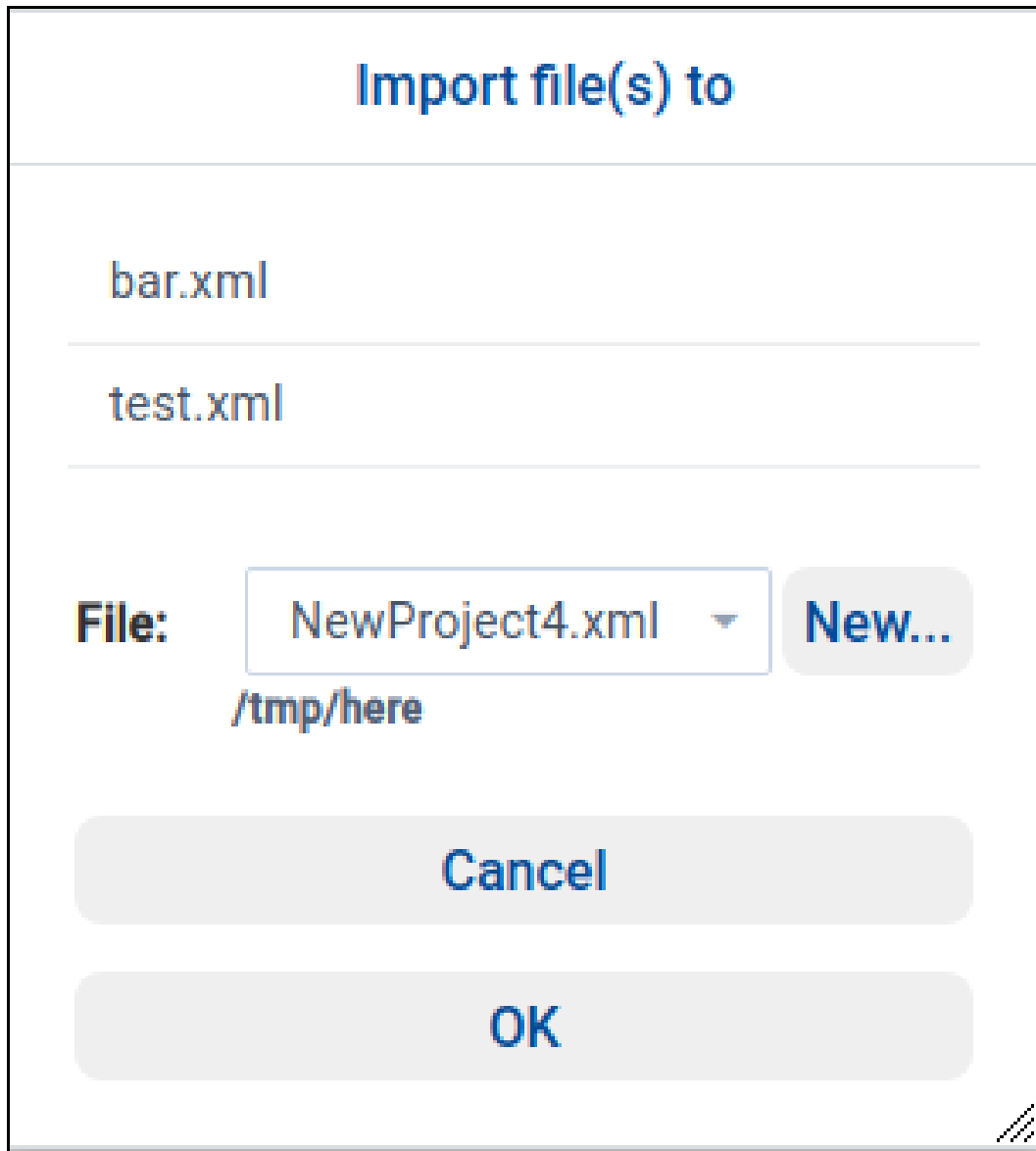
To import one or more files and add them to the current project, select **Import...** from the Files menu:



This will open two file browsers. The first one looks for an XML file or an IDL file to import:

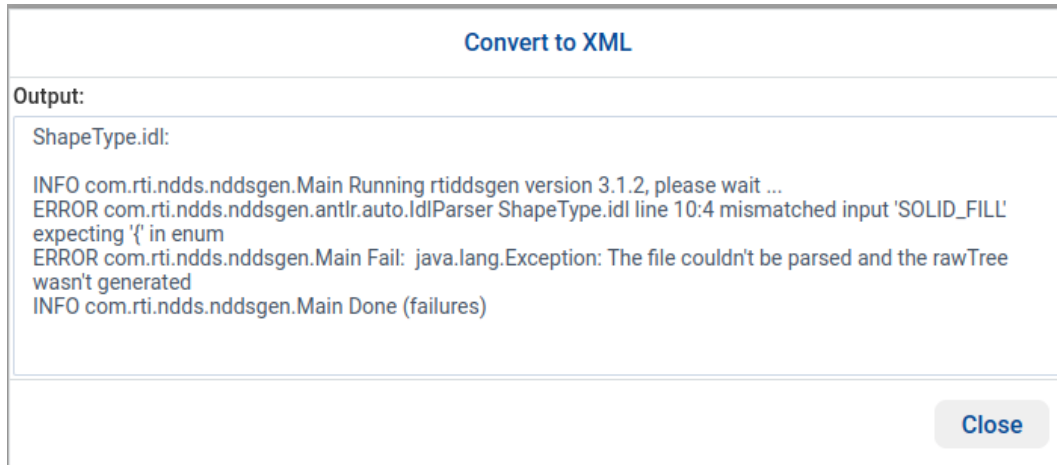


Once a file (or files) is selected, a second window lets you copy the content of the file(s) into an existing XML file or create a new XML file with the selected file content:



Note: If *Code Generator* fails to convert all the IDL files, the second window will not appear.

When importing an IDL file, *System Designer* calls *Code Generator* to convert the IDL to XML. If an error occurs while converting the IDL, a dialog box appears showing the issue:



```

8
9  enum ShapeFillKind |
10      SOLID_FILL,
11      TRANSPARENT_FILL,
12      HORIZONTAL_HATCH_FILL,
13      VERTICAL_HATCH_FILL
14  };
15

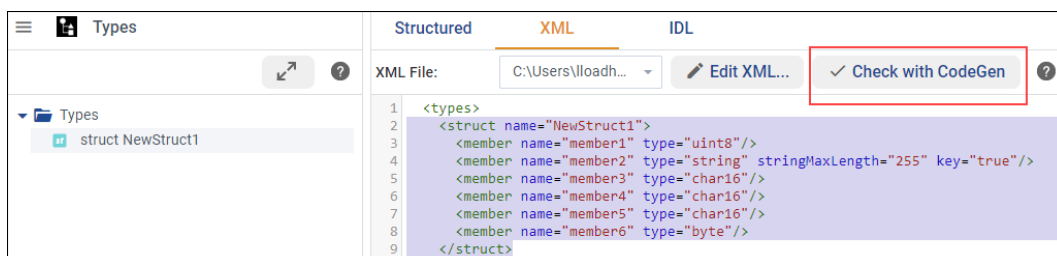
```

2.8 XML Validation

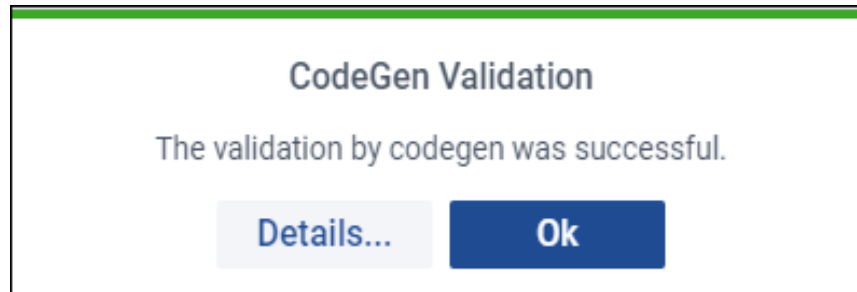
2.8.1 Code Generator

System Designer has an option to validate any XML against *Code Generator*. For example, you can see if the data type that you are defining is valid or not.

To validate the XML, select the XML view, select the XML file from the list of XML files used by the current project, then click the **Check with CodeGen** button.



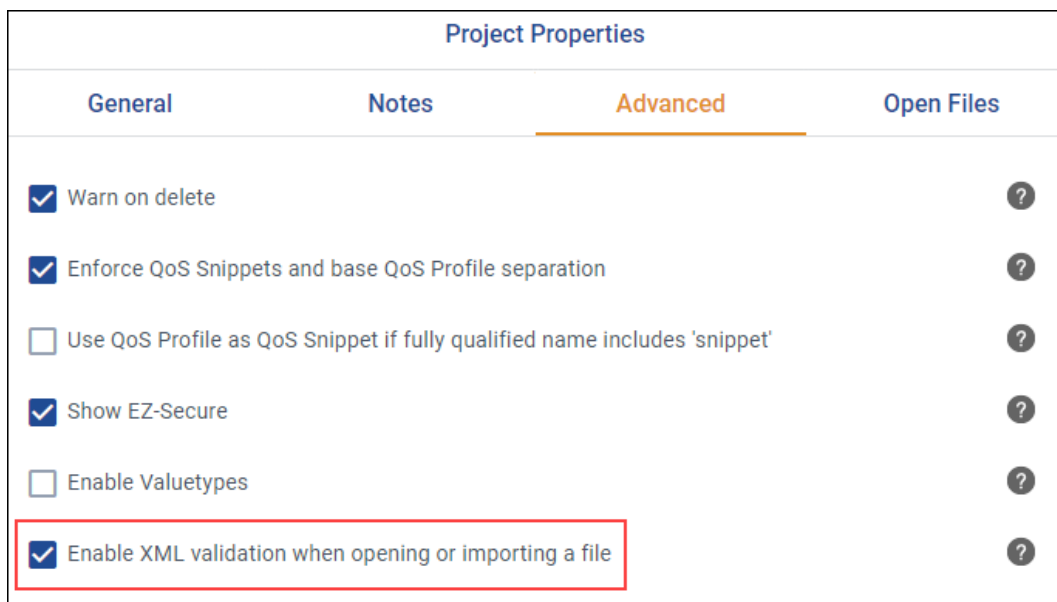
Code Generator will generate code with the file that is selected in the XML File drop-down menu. The result of this operation will be shown once *Code Generator* finishes parsing the input file.



2.8.2 Other Methods

Note: The XML validation described in this section uses the tool **xmllint**. See *Prerequisites* for more information.

System Designer automatically validates all the XML files included in a project against *Code Generator* when opening an existing project, and when adding an XML file to the project. This validation can be disabled in the Advanced Project Properties dialog.



You can also validate XML through the Projects menu. See *Validate Project*.

2.9 Online Help

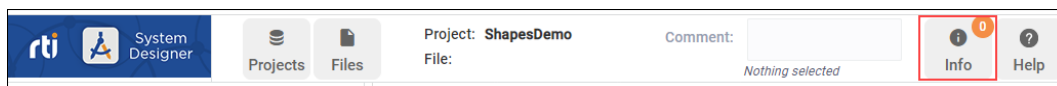


When using the tool, you will often see a help icon: .

Click the icon to see online tips and understand what to do next.

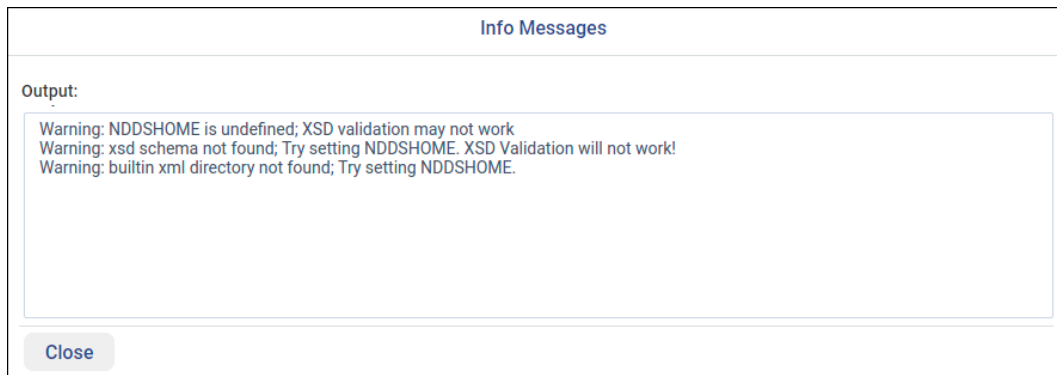
2.10 Info Dialog

This dialog can be accessed using the Info button on the far right of the toolbar:



The Orange badge indicates the number of unread messages; it is reset when the button is clicked.

Once clicked, you will see any Info Messages:



Chapter 3

Tutorials

This chapter describes several examples, all of which require a Project. You can create your own Project by following Section 2.2.

3.1 Defining Your Data Type

This example shows how to define your own data type using *System Designer*. In this example, we are going to create the ShapeTypeExtended data type used by the Shapes Demo:

```
enum ShapeFillKind {
    SOLID_FILL,
    TRANSPARENT_FILL,
    HORIZONTAL_HATCH_FILL,
    VERTICAL_HATCH_FILL
};

struct ShapeType
{
    string<128> color; //@key
    long x;
    long y;
    long shapesize;
}; //@Extensibility EXTENSIBLE_EXTENSIBILITY

struct ShapeTypeExtended : ShapeType {
    ShapeFillKind fillKind;
    float angle;
}; //@Extensibility EXTENSIBLE_EXTENSIBILITY
```

1. Create a new project (see Section 2.2), name it **ShapesDemo**, and click OK.

New Project

General
Notes
Advanced
Open Files

Project Name:

ShapesDemo

Only alphanumeric characters, underscores '_', dots '.', and hyphens '-' are allowed

Project Path:

C:/Users/tmp

...

☒
Automatically create an empty XML file

?

☒
Autosave project

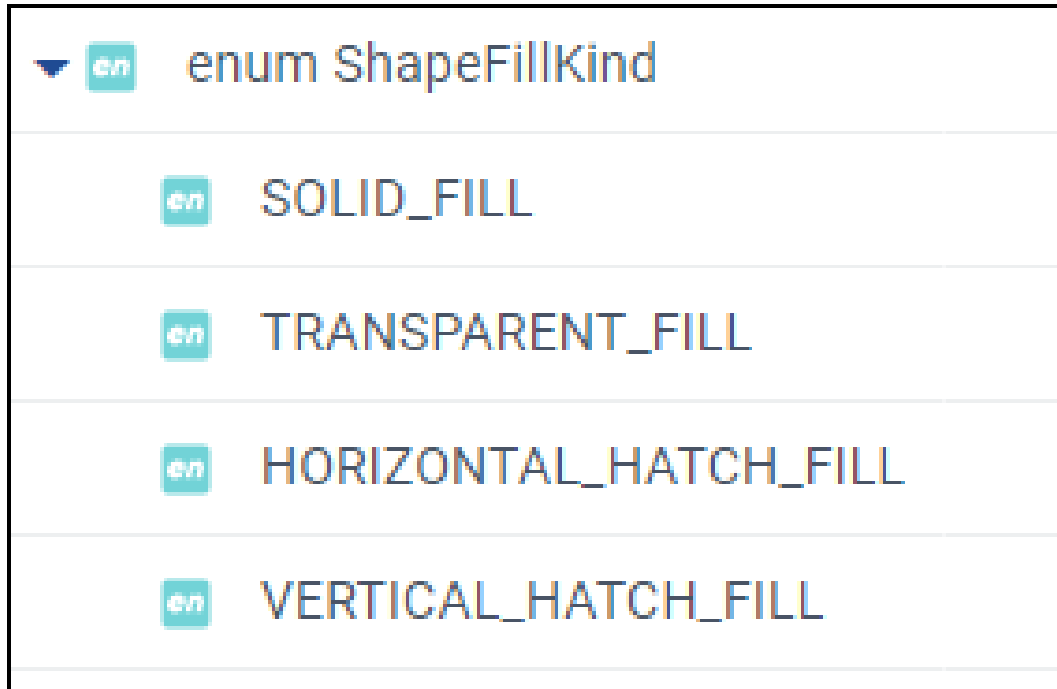
?

System Designer Version: **7.1.0**

Connex DDS Version: **7.1.0**

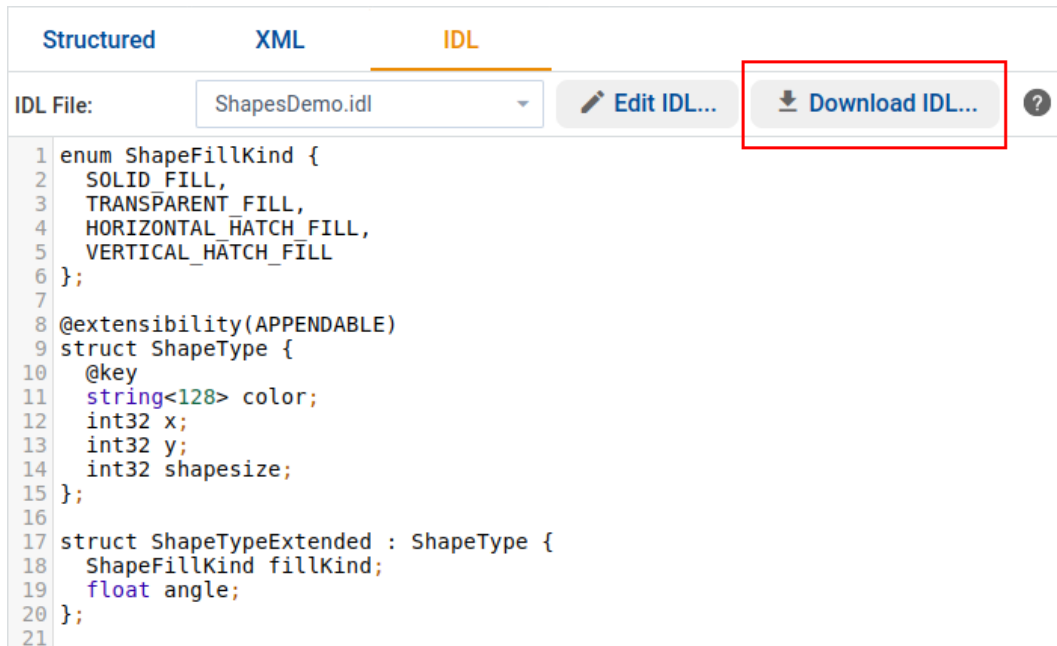
Project Created with System Designer: **7.1.0**

2. Go to the Types View. By default it should be the first view shown by *System Designer*.
3. Add an enumeration by right-clicking on Types and selecting **Add Enumeration...** Name your enumeration **ShapeFillKind** and click OK.
4. Right-click on the recently created enum and select **Add Member...** to start adding each one of the following four enumerator value names: SOLID_FILL, TRANSPARENT_FILL, HORIZONTAL_HATCH_FILL, and VERTICAL_HATCH_FILL. You can leave the Value blank and @default_literal unchecked.



5. Create a struct and name it **ShapeType** by right-clicking on Types and selecting **Add Struct...** Select APPENDABLE as its @extensibility value and click OK. Fill the struct, adding each member by right-clicking on the struct and selecting **Add Member...**:
 - Add a bounded (128) string named **color**. This element will be the key of our type, so select the @key checkbox. (You can accept the other defaults in this window.)
 - Add three longs named **x**, **y** and **shapsize**. Although 'long' is not in the list of suggestions for the Type field, you can type it in. Or use the newer int32 instead of long. See *Working with Types*.
6. Create a second struct and name it **ShapeTypeExtended**. Choose ShapeType as its base type. Add its elements by right-clicking on the struct and selecting **Add Member...**:
 - Add a ShapeFillKind enumerator named **fillKind**.
 - Add a float32 named **angle**.

When the type definition is complete, you can download the IDL by clicking on the Download IDL button in the IDL view.



Now you have an IDL file to use to create your own Shapes Demo application.

3.2 Defining an XML Application Creation Configuration File

This example shows how to define an XML configuration file using *System Designer*. In this example, we are going to create the same configuration file used by the **hello_world_xml_dynamic** example.

For more information about XML-Based Application Creation, see the [RTI Connex DDS Core Libraries XML-Based Application Creation Getting Started Guide](#).

1. Create a new XML file inside the open Project by clicking the Files icon at the top and selecting **New...** You can use the same project as you used in the previous tutorial, or a new project. Name the file **USER_QOS_PROFILES.xml**. Click OK.

Note: If you see a default XML file already associated with the project, that's because you kept the default selection, "Automatically create an empty XML file" when you created the project. You can still create **USER_QOS_PROFILES.xml**, it will not replace the default one.

2. Go to the Types View. By default it should be the first view shown by *System Designer*.
3. Add two constants (right-click Types > **Add Constant...**) as follows:
 - Name 'MAX_NAME_LEN', Type 'long', value 64, File **USER_QOS_PROFILES.xml**
 - Name 'MAX_MSG_LEN', Type 'long', value 128, File **USER_QOS_PROFILES.xml**

Add Constant

Name: MAX_NAME_LEN

Type: long
Press [Key Down] or start typing for suggestions

Value: 64

☒ @resolve-name

File: USER_QOS_PROFILES.xml **New...**
C:/Users/loadholt/

Constant Declaration

Constants have the following restrictions:

- Name cannot contain spaces or special characters like %/#...
- The type of a constant can be a primitive type (which includes all integers types, floating point types, characters and string types) or aliases to those types defined via a typedef.
- String (and wstring) constants **must be enclosed in double quotes** (otherwise the value is misinterpreted as another constant)
- Limits on strings and wstrings are not specified.
- **@resolve-name** tells rtiddsgen to attempt to resolve the type of this const (on by default). When checked, if a particular type cannot be resolved during code generation rtiddsgen will report an error. If unchecked, generation will proceed (but you might get a compilation error). This annotation is normally used

Cancel OK

4. Create a struct (right-click **Types > Add Struct...**) and name it **HelloWorld**. Fill the struct by adding each of the following members (right-click on the struct and select **Add Member...**):

- Add a bounded string named **sender** with a maximum length of MAX_NAME_LEN. (For this and the following members, you can accept the defaults in the window.)
- Add a second bounded string named **message** with a maximum length of MAX_MSG_LEN.

Add Struct Member

Name: sender

Type: string
Press [Key Down] or start typing for suggestions

String Maximum Length: MAX_NAME_LEN

☐ Sequence size: Unbound

☐ Array dimensions: i.e.: 10,20

☐ @key ☐ @optional

☐ @external ☒ @resolve-name

☐ @hashid

☐ @id

☐ @default

☐ @min ☐ @max

Member Declaration

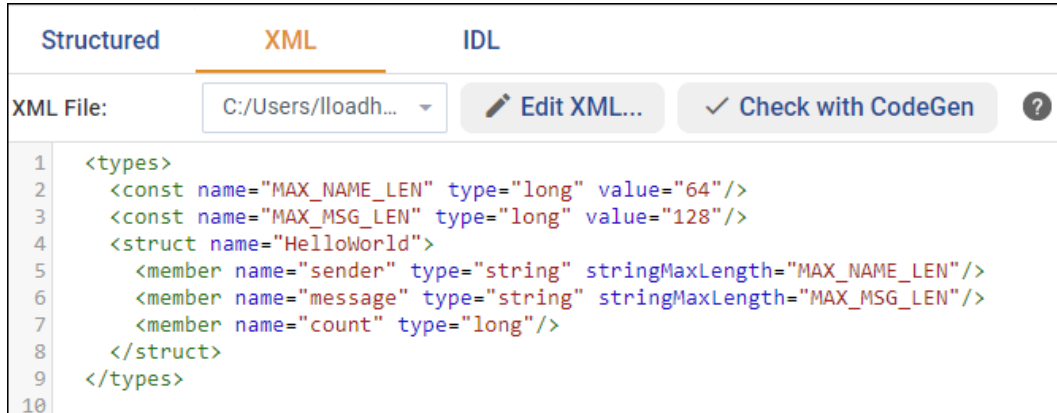
Here you define an individual member of your struct.

- **@key** lets you specify that this member is part of a key used in your type.
- **@external** tells the middleware that the target data should be stored, not inline within its containing type, but external to it (i.e. using a pointer).
- **@resolve-name** tells rtiddsgen to attempt to resolve the type of this member (on by default). When checked, if a particular type cannot be resolved during code generation rtiddsgen will report an error. If unchecked, generation will proceed (but you might get a compilation error). It is normally used on architectures where there is no preprocessor that can resolve #includes. Applies only if non-primitive types are used for this member.
- **@optional** lets you to set the value NULL when writing to avoid sending any value for this field.
- **@hashid** lets you manually specify the hash ID that will be used to identify this member.
- **@id** is optionally used to specify the ordering of this member relative to the other members of the struct.
- **@visibility** lets you specify whether this member is private or public (valuetypes only).

Add more... Cancel OK

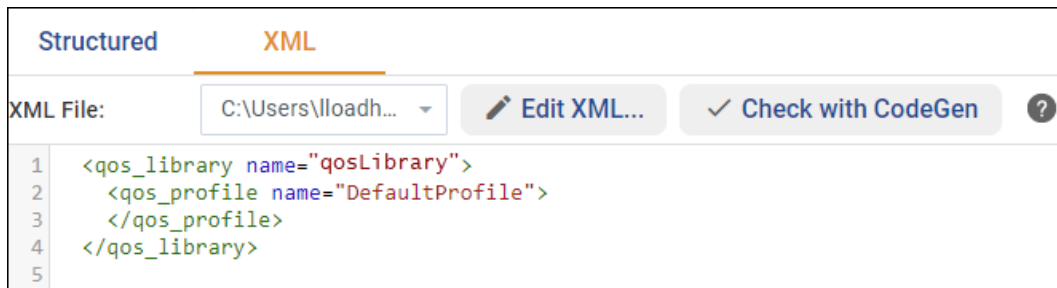
- Add a long named **count**.

5. Click the XML view to view how your data type definition looks in XML:



6. Define the QoS that the application is going to use by clicking on the QoS view. Since we are creating the same XML file as the **hello_world_xml_dynamic** example has, to match the content of that file you only need to add a new QoS Library named **qosLibrary** and a QoS Profile named **DefaultProfile** inside the created library. (You do not have to select any of the checkboxes.)

In the XML view, your QoS Profile should look like this:



7. Now that the QoS has been defined, define the system's Topic and its corresponding data types, first by defining the Domain:

- In the Domain view, right-click on Domain Libraries and select **Add Library**. Name your domain **MyDomainLibrary** and click OK.

Note: If your project contains more than one XML file, select the USER_QOS_PROFILES.xml file.

- Add a domain to your library by right-clicking MyDomainLibrary and selecting **Add Domain**. Name it **HelloWorldDomain**, entering 0 as its Domain Id.
- Select HelloWorldDomain and click the **Add** button to add a registered type. Name the type **HelloWorldType**, selecting the 'HelloWorld' struct that we created previously as its type reference.

Registered Type Properties

Name:

Type Ref.:
Press [Key Down] or start typing for suggestions

☐ Use compiled types ?

Type Kind: ?

- In the Topics panel, click the **Add** button to add a topic named **HelloWorldTopic**, selecting the 'HelloWorldType' registered type as its registered type reference.

Topic Properties

Name:

Reg. Type Ref.:
Press [Key Down] or start typing for suggestions

Topic QoS:

In the XML view, your domain configuration should look like this:

XML File:

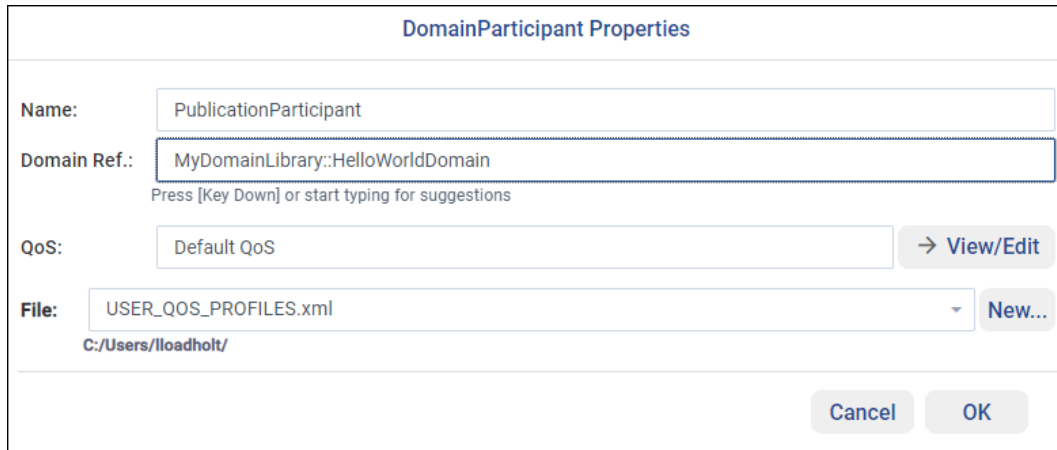
```

1  <domain_library name="MyDomainLibrary">
2    <domain name="HelloWorldDomain" domain_id="0">
3      <register_type name="HelloWorldType" type_ref="HelloWorld"/>
4      <topic name="HelloWorldTopic" register_type_ref="HelloWorldType"/>
5    </domain>
6  </domain_library>
7
8

```

- Define the elements that are going to be part of our system by clicking on the Participant view. In this example, we are going to create a simple application containing one DomainParticipant, one Publisher, one Subscriber, one DataReader, and one DataWriter.
 - Add a new Participant Library, naming it **MyParticipantLibrary**. Add a Participant inside that library named **PublicationParticipant**, selecting the previously created 'MyDomainLibrary::HelloWorldDomain' as its domain reference.

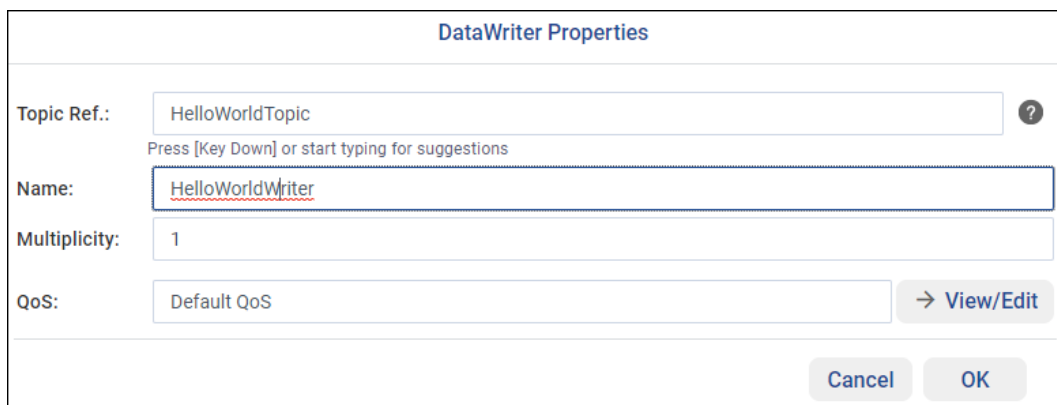
Note: If your project contains more than one XML file, select the USER_QOS_PROFILES.xml file.



The **DomainParticipant Properties** dialog box contains the following fields and controls:

- Name:** PublicationParticipant
- Domain Ref.:** MyDomainLibrary::HelloWorldDomain
Press [Key Down] or start typing for suggestions
- QoS:** Default QoS → View/Edit
- File:** USER_QOS_PROFILES.xml New...
C:/Users/Iloadholt/
- Buttons:** Cancel, OK

- Click the **Add** button to add a publisher named **MyPublisher**.
- Click the **Add** button to add a DataWriter named **HelloWorldWriter** selecting 'HelloWorldTopic' as its topic reference.




The **DataWriter Properties** dialog box contains the following fields and controls:

- Topic Ref.:** HelloWorldTopic ?
Press [Key Down] or start typing for suggestions
- Name:** HelloWorldWriter
- Multiplicity:** 1
- QoS:** Default QoS → View/Edit
- Buttons:** Cancel, OK





- Add a second Participant inside that library named **SubscriptionParticipant**, selecting the previously created 'MyDomainLibrary::HelloWorldDomain' as its domain reference.
- Click Subscriptions and add a subscriber named **MySubscriber**.
- Add a DataReader named **HelloWorldReader**, selecting 'HelloWorldTopic' as its topic reference. Unlike for the DataWriter, for the DataReader we are going to select the QoS profile that we created earlier, named 'DefaultProfile' (in the Default QoS): click on View/Edit inside the DataReader dialog and name the QoS Profile **HelloWorld_reader_qos**. Select 'qosLibrary::DefaultProfile' as its Base QoS.

Editing DataReader QoS Profile for DataReader Definition


Profile Name:

Base QoS(s): 
Defined locally

Topic Filter:




 Restore
  Set Defaults
 

Durability (Durability QoS Policy)	▼
Deadline (Deadline QoS Policy)	▼
Latency budget (Latency Budget QoS Policy)	▼
Liveliness (Liveliness QoS Policy)	▼
Reliability (Reliability QoS Policy)	▼
Destination order (Destination Order QoS Policy)	▼
History (History QoS Policy)	▼
Resource limits (Resource Limits QoS Policy)	▼

 Close

9. Your XML view should look like this:

Structured XML

XML File:  Edit XML...  Check with CodeGen 

```

1  <domain_participant_library name="MyParticipantLibrary">
2    <domain_participant name="PublicationParticipant" domain_ref="MyDomainLibrary::HelloWorldDomain">
3      <publisher name="MyPublisher">
4        <data_writer name="HelloWorldWriter" topic_ref="HelloWorldTopic"/>
5      </publisher>
6    </domain_participant>
7    <domain_participant name="SubscriptionParticipant" domain_ref="MyDomainLibrary::HelloWorldDomain">
8      <subscriber name="MySubscriber">
9        <data_reader name="HelloWorldReader" topic_ref="HelloWorldTopic">
10         <datareader_qos name="HelloWorld_reader_qos" base_name="qosLibrary::DefaultProfile">
11         </datareader_qos>
12       </data_reader>
13     </subscriber>
14   </domain_participant>
15 </domain_participant_library>
16

```

You can use this XML file to define your *Connex DDS* system using XML-Based Application Creation.



3.3 Defining a Routing Service Configuration File

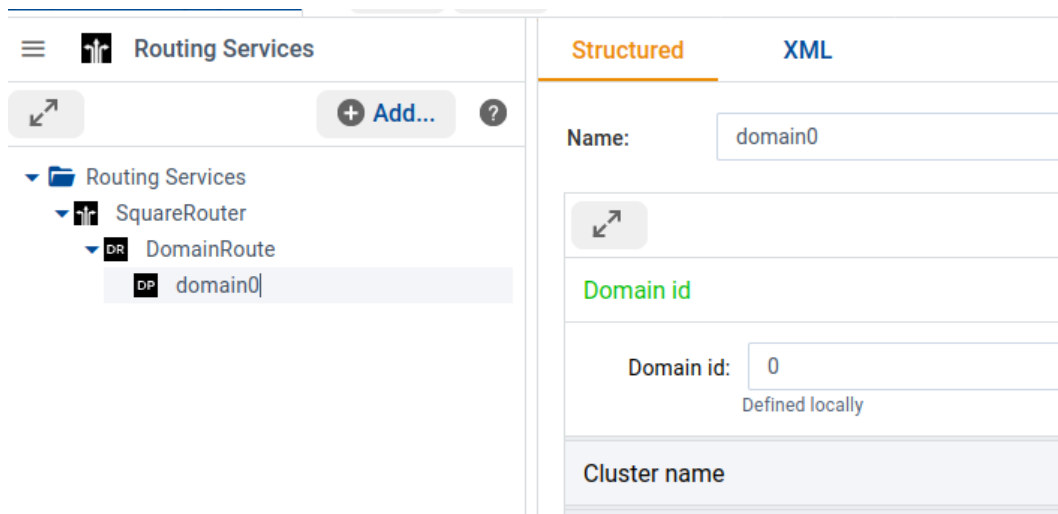
This example shows how to define a *Routing Service* configuration file using *System Designer*. In this example, we are going to create the same configuration file used by the first example in the *Routing Service* Tutorial, see [Routing a single specific Topic in the RTI Routing Service User's Manual](#).

For more information about the *Routing Service* configuration file, see the [Configuration Section of the RTI Routing Service User's Manual](#).

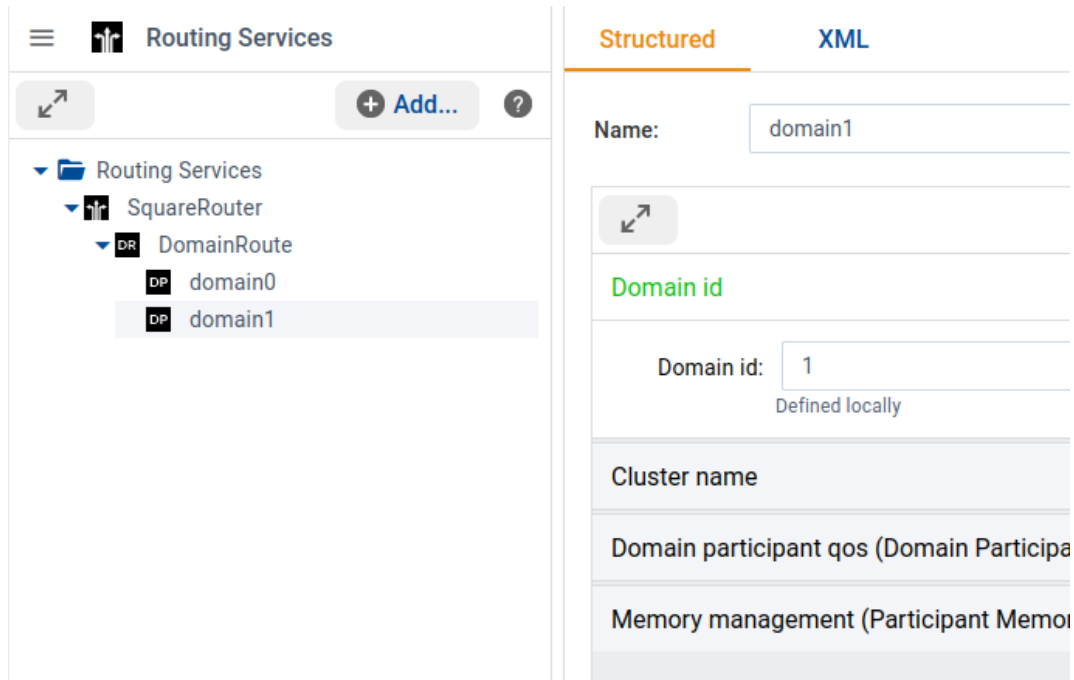
1. Create a new XML file inside the open project by clicking the Files icon at the top and selecting **New...** You can use the same project as you used in the previous *Tutorials*, or a new project. Name the file **rti_rs_example_square_topic.xml**. Click OK.




Note: If you see a default XML file already associated with the project, that's because you kept the default selection, "Automatically create an empty XML file" when you created the project. You can still create **rti_rs_example_square_topic.xml**, it will not replace the default one.

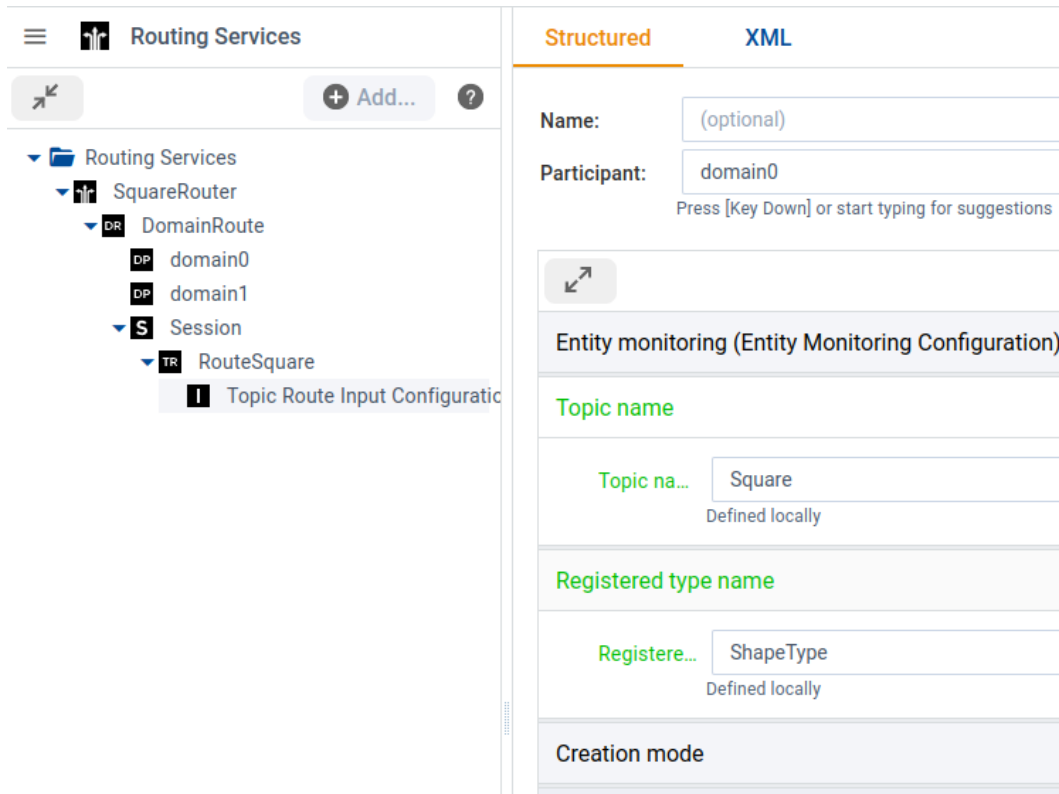
2. Go to the Routing Services View.
3. Right-click on Routing Services and select **Add Routing Service**. Name your service **SquareRouter** and click OK. *System Designer* will automatically select and open the newly created service.
4. Add a Domain Route (right-click on the 'SquareRouter' service or click on the  in the Domain Routes table) and name it **DomainRoute**.
5. Add two Participants to the newly created Domain Route (right-click on the 'DomainRoute' Domain Route and select **Add Participant** or click on the  in the Connections, Participants & Session table and select **Participant**).
 - Name the first Participant **domain0** and set the Domain id to 0.




- Name the second Participant **domain1** and set the Domain id to 1.

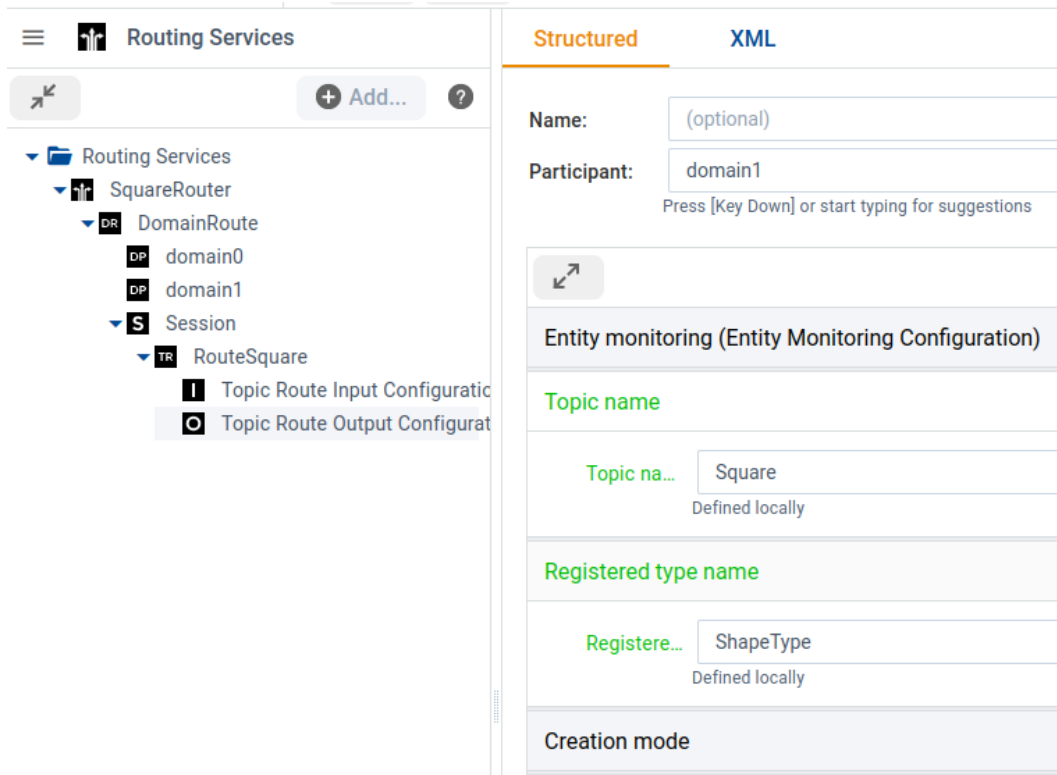


6. Add a Session to the 'DomainRoute' Domain Route (right-click on the 'DomainRoute' Domain Route and select **Add Session** or click on the  in the Connections, Participants & Session table and select **Session**) and name it **Session**.
7. Add a Topic Route to the 'Session' Session (right-click on the 'Session' Session and select **Add Topic Route** or click on the  in the Routes & AutoRoutes table and select **Topic Route**) and name it **RouteSquare**.
8. Add an Input to the 'RouteSquare' Topic Route (right-click on the 'RouteSquare' Topic Route and select **Add Input** or click on the  in the Input & Output table and select **Input**).
 - Set the Participant to **domain0**.
 - Set the Topic name to **Square**.
 - Set the Registered type name to **ShapeType**.

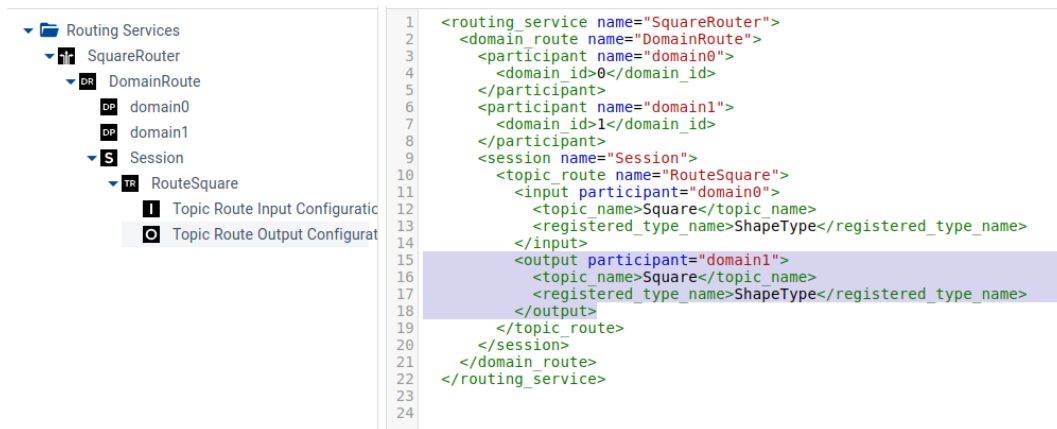


9. Add an Output to the 'RouteSquare' Topic Route (right-click on the 'RouteSquare' Topic Route and select **Add Output** or click on the  in the Input & Output table and select **Output**).

- Set the Participant to **domain1**.
- Set the Topic name to **Square**.
- Set the Registered type name to **ShapeType**.



10. Your XML view should look like this:



You can use this XML file to route the samples for the Topic **Square** from domain 0 to 1 following the [Routing a single specific Topic in the RTI Routing Service User's Manual](#) example in the *Routing Service Tutorial*.

Chapter 4

Release Notes

4.1 Supported Platforms and Browsers

System Designer 7.0.0 is compatible with *Connex* 7.0.0.

See the *System Designer* column in the [Table of Supported Platforms for Connex Applications, in the Core Libraries Release Notes](#).

System Designer works with these browsers:

- Chrome® version 77
- Firefox® version 69
- Safari® version 12 (on macOS systems only)

4.2 Compatibility

For backward compatibility issues between the current and previous releases of *System Designer*, see the Migration Guide on the RTI Community Portal (<https://community.rti.com/documentation>).

4.3 What's New in 7.2.0

4.3.1 Specify the time period to check for Identity Certificate and CRL file changes

Two new fields, **CRL File Poll Period** and **Identity Certificate File Poll Period**, are available in the **Authentication** tab in the Security Settings dialog box. These fields enable you to configure the period at which the *Security Plugins* check for changes in the Identity Certificate and CRL files.

4.3.2 System Designer now accepts fully qualified names when searching types

System Designer now accepts fully qualified names (module::module::name) when searching types. In addition, the tooltip now shows the fully qualified name.

4.3.3 New option disables type checking to allow using undefined types

The **Use compiled types** checkbox is a new optional setting in the **Registered Type Properties** dialog accessed from the Domain or Routing Services view. When this checkbox is enabled, *System Designer* will not check if the referenced type is defined in the Types view.

Registered Type Properties

Name: (required)

Type Ref.: (required)
Press [Key Down] or start typing for suggestions

☐ Use compiled types ?

Registered ... (optional)

Cancel OK

4.3.4 “Add..” button is now available when a QoS Profile is selected

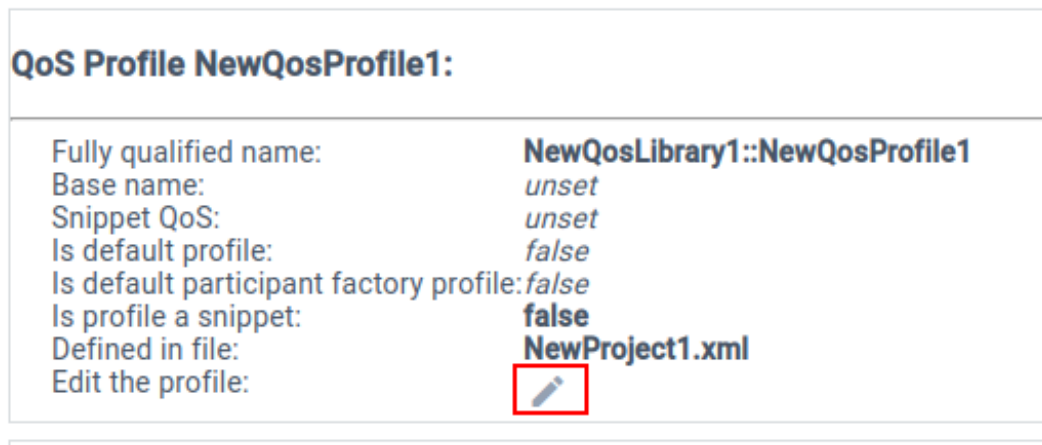
System Designer now supports creating any entity QoS, such as a DataReader QoS, using the **Add** button when a QoS Profile is selected in the QoS view.

4.3.5 Last Modified Time project property no longer updates on open

The Last Modified Time project property is updated only when the project is modified. The modified time is no longer updated when the project is opened for the first time.

4.3.6 QoS Profiles can now be edited directly from their panel

The QoS Profile panel now includes an **Edit the profile** command, enabling you to edit a profile directly from the panel.



4.3.7 QoS Profiles defined in the open project display first when working with inheritance

The QoS Profiles inheritance panel now displays the QoS Profiles defined in the open project before the Built-in Profiles.

4.3.8 IDL files can now be imported

System Designer now supports importing IDL files to the current project. On importing an IDL file, *Code Generator* is called to convert the IDL to XML. If an error occurs while converting the IDL, a detailed error message displays in the **Convert to XML** dialog box.

4.3.9 Disable @key option when a member type contains an optional member

When creating a new struct member, *System Designer* now checks if its type contains an optional member. If so, the **@key** checkbox is disabled because a member cannot be declared as a key when its type contains an optional member.

4.3.10 Edit IDL files directly in System Designer

System Designer now supports editing IDL files, with changes reflected in the Structured view.

4.3.11 Display common profiles only when creating a QoS inheritance

A new **Show Common Profiles** checkbox in the QoS Profiles Inheritance Chain panel filters the list of available QoS Profiles to just those that are commonly used. These common profiles include:

- Builtin profiles starting with BuiltinQosLib::Pattern
- QoS Profiles defined in the open project

4.3.12 Specify maximum number of removed participants to track

A new field, **Maximum Number of Remote Participants per Key**, is available in the **Authentication** tab in the Security Settings dialog box. This field enables you to configure the number of remote *DomainParticipants* to keep track of that were banished by the Security Plugins due to a revoked or expired certificate.

For more information, see [Automatically Banishing Ignored Participants](#) in the *RTI Security Plugins User's Manual*.

4.4 What's Fixed in 7.2.0

4.4.1 Negative values were not allowed when setting the minimum, maximum, and default member values

System Designer did not allow negative values when setting the minimum, maximum and default value of a member in the Types View. This problem has been resolved.

[RTI Issue ID SYSD-1123]

4.4.2 Invalid use of the Unix path separator in Windows

System Designer sometimes used the Unix path separator (“/”) in place of the Windows path separator (“\”). This usage caused issues when trying to navigate file paths on Windows machines. This problem has been resolved.

[RTI Issue ID SYSD-1108]

4.4.3 System Designer incorrectly displayed the <base_name> tag after removing a QoS Snippet

System Designer displayed the <base_name> tag after removing the QoS Snippet from a QoS Profile or Entity QoS in the XML View. This problem has been resolved so that <base_name> is not displayed when the QoS Profile or Entity QoS is not inheriting from anything.

[RTI Issue ID SYSD-1105]

4.4.4 System Designer allowed inheritance from types defined below the current type

System Designer allowed inheritance from types defined below the current type, as demonstrated in the code snippet below. This problem has been resolved. Now, *System Designer* throws an error when trying to inherit from a type defined below the current one.

```
struct A : B {};  
struct B {};
```

[RTI Issue ID SYSD-623]

4.5 Previous Releases

4.5.1 What's New in 7.1.0

New Types View option enables adding, editing, and removing #include directives

The Types View now includes an option to add, edit, and remove #include directives. In addition, *System Designer* displays the Data Types defined inside any #include directive when:

- Expanding the element in the Structured panel.
- Defining a Struct or Typedef. The elements defined inside the #include directive will appear in the suggestion list.

New Routing Service XML configuration views

The following new Routing Service configuration views have been added in *System Designer*.

- **Routing Services:** Use this view to specify a library of Routing Service plugins.
- **Plugin Libraries:** Use this view to assign default values to XML variables.

New field to configure RTPS PSK Protection

A new field, **RTPS Protection Preshared Key**, is available to configure the new RTPS PSK Protection. The field is located on the Cryptography tab when configuring security on the DomainParticipant QoS.

New field to configure notification when a security certificate is about to expire

A new field, **Certificate Expiration Advance Notice**, is available in Security Global Settings. This field enables you to configure how far in advance, in seconds, you want to be notified about a security certificate that is about to expire.

Copy directive (@copy) now available for C# and Python

The following C# and Python directives are now available in *System Designer*:

- copyCs
- copyCsDeclaration
- copyPy
- copyPyDeclaration

The following C++ directives were removed from *System Designer*. XML files containing these deprecated directives will not load, and *System Designer* will throw the error message `Error: Invalid directiveKind: copyCppcli`.

- copyCppcli
- copyCppcliDeclaration

Support for the OBSERVABILITY_COLLECTOR_SERVICE_QOS kind category

System Designer now supports the OBSERVABILITY_COLLECTOR_SERVICE_QOS kind category.

New fields to configure the Participant Factory QoS Monitoring Tag

A new set of fields enables you to configure the Monitoring Tag when defining a Participant Factory QoS.

Third-party software upgrades

The following third-party software used by *System Designer* has been upgraded:

Table 4.1: Third-Party Software Changes

Third-Party Software	Previous Version	Current Version
Express	4.17.2	4.18.2
body-parser	1.19.1	1.20.1

These upgrades may fix potential vulnerabilities. See *Fixes Related to Vulnerabilities*. For information on third-party software used by Connex products, see the “3rdPartySoftware” documents in your installation: <NDDSHOME>/doc/manuals/connex_dds_professional/release_notes_3rdparty.

4.5.2 What’s Fixed in 7.1.0

System Designer did not select newly created files

After creating an element in a new file while working in any available View (Types, QoS, Domain, etc.), *System Designer* did not select the new file. As a result, the XML tab displayed all elements instead of only those elements defined in the new file. This problem has been resolved.

[RTI Issue ID SYSD-1096]

The Add more... command did not generate a new member name when adding an Enumerator or Struct member

When adding multiple members to an Enumeration or Struct data type, the **Add more...** button did not update the **Name** field with a new, unique member name. This problem has been resolved.

[RTI Issue ID SYSD-1039]

System Designer failed to create string and wstring constants

When trying to define a string or wstring constant in *System Designer*, the following error was thrown:

Value does not match constant type

This problem has been resolved.

[RTI Issue ID SYSD-1042]

System Designer did not open new projects in a new window

When the **Open in a new window** option was selected in the New Project dialog, *System Designer* did not open the project in a new window and generated the following error:

`Unable to load the project: missing path from req.query.`

This problem has been resolved.

[RTI Issue ID SYSD-1045]

System Designer did not create a new XML file when creating a new element

After creating a new element from any tab (QoS, Types, Domain, or Participant), *System Designer* failed to create a new XML file. This issue has been resolved.

[RTI Issue ID SYSD-1053]

System Designer failed to create a Typedef when using @default, @min or @max annotations

System Designer did not create a new Typedef if @default, @min, or @max annotations were selected in the Add Typedef dialog. This problem has been resolved.

[RTI Issue ID SYSD-1056]

System Designer failed to change the Directive Kind in an existing directive

When editing an existing directive, *System Designer* did not allow changes to the Directive Kind field. This problem has been resolved.

[RTI Issue ID SYSD-1067]

Fixes Related to Vulnerabilities

Potential performance issues in System Designer

System Designer's body-parser, express, and express-session had a third-party dependency on Debug version 2.6.9, which is known to be affected by a number of publicly disclosed vulnerabilities. The impact on *System Designer* was:

- This vulnerability led to inefficient regular expression complexity that may have impacted performance.
- CVSS v3.1 Score: 3.5 LOW
- CVSS v3.1 Vector: [AV:A/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:L](#)

These vulnerabilities have been fixed by upgrading Debug to version 4.3.4, the latest stable version.

User Impact without Security

The impact on *System Designer* was:

- This vulnerability led to inefficient regular expression complexity that might impact the performance of *System Designer*.
- CVSS v3.1 Score: 3.5 LOW
- CVSS v3.1 Vector: [AV:A/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:L](#)

User Impact with Security

Same as *User Impact without Security*.

[RTI Issue ID SYSD-1081]

Potential hang when receiving an attack payload passed in the URL query string

System Designer had a third-party dependency on **express** version 4.17.2. This **express** version had a dependency on **qs** version 6.9.6, which is known to be affected by a number of publicly disclosed vulnerabilities. These vulnerabilities have been fixed by upgrading **express** to version 4.18.2, the latest stable version.

User Impact without Security

- Exploitable through an attack payload in the query string of the URL used to visit *System Designer*.
- *System Designer* could hang.
- CVSS v3.1 Score: 7.5 HIGH
- CVSS v3.1 Vector: [AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H](#)

User Impact with Security

Same impact as *User Impact without Security*.

[RTI Issue ID SYSD-1073]

Fixes Related to Generated Code

System Designer allowed invalid type definitions, causing Code Generator to fail

When defining a datatype, *System Designer* allowed invalid annotation combinations that could not be used by Code Generator. This problem has been resolved.

The following *System Designer* changes have been made to prevent invalid type definitions:

- @external checkbox will be disabled when @default is selected, and vice versa
- If @language_binding(FLAT_DATA) is selected:
 - An error appears when the value of the @allowed_data_representation annotation is not XCDR2
 - An error appears when the value of the @extensibility annotation is APPENDABLE
 - An error appears when trying to define non-fixed-size members (sequences and @external)

[RTI Issue ID SYSD-1055]

4.5.3 What's New in 7.0.0

Support for the DISCOVERY logging category

System Designer now supports the DISCOVERY logging category.

New field to configure Domain tag when defining Participant QoS

A new field has been added to easily configure the Domain Tag when defining a Participant QoS:

Profile Name:	(optional)
Base QoS(s):	BuiltinQosLib::Generic.Security
	Inherited from parent profile
Domain Tag:	(optional)

The Domain Tag can also be configured as a regular property, `dds.domain_participant.domain_tag`. In this case, *System Designer* will automatically show the Domain Tag value in the new field.

Profile Name:

Base QoS(s):

Domain Tag:

↕

Transport builtin (Transport Builtin QoS Policy)
Default unicast (Transport Unicast QoS Policy)
Discovery (Discovery QoS Policy)
Resource limits (Domain Participant Resource Limits QoS Policy)
Event (Event QoS Policy)
Receiver pool (Receiver Pool QoS Policy)
Database (Database QoS Policy)
Discovery config (Discovery Config QoS Policy)
Property (Property QoS Policy)

Stores name/value (string) pairs that can be used to configure certain parameters of Connex DDS that are not e

Value:

double-click to add a new item.

The new field will also indicate if the Domain Tag has been inherited from a base QoS.

Profile Name:	(optional)
Base QoS(s):	NewQosLibrary1::NewQosProfile1::Participant1 Defined locally
Domain Tag:	Domain A Inherited from parent profile

Improvements to .rtisdproj file to be human-readable and contain more metadata

The *System Designer* .rtisdproj file has been changed to a human-readable JSON format, instead of a single-line string.

The .rtisdproj file now contains these additional properties:

- The original creation date of the project.
- The version of System Designer used to create the project.

The Project Properties pane has been updated to display these properties.

New field to configure Partition QoS when defining a DomainParticipant

A new field allows you to configure the Partition QoS policy when defining a DomainParticipant's QoS.

StructuredXML

Profile Name: (optional)

Base QoS(s):
Value unset

Domain Tag: (optional)

↕ ↗

User data (User Data QoS Policy)

Entity factory (Entity Factory QoS Policy)

Partition (Partition QoS Policy)

Provides another way to control which DataWriters will match which DataReaders... [\[More...\]](#)

double-click to add a new item.

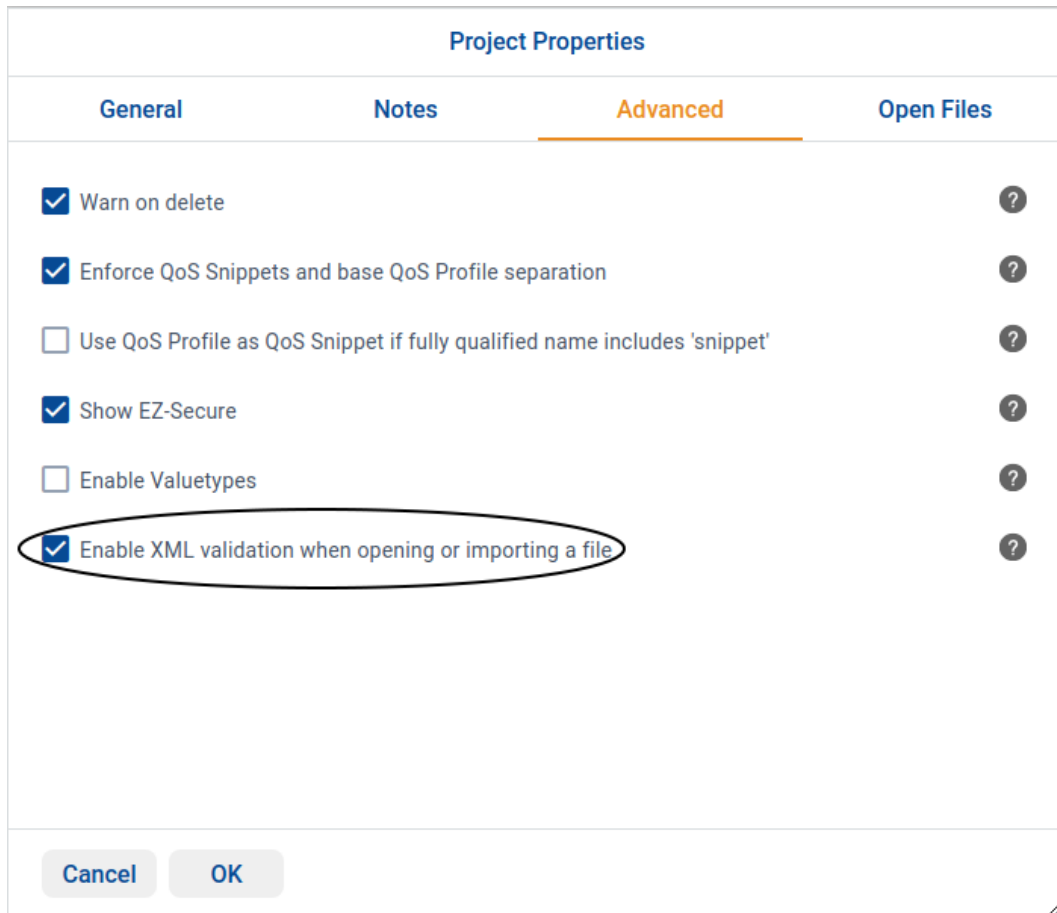
Name:

QoS snippets can now be composed of other QoS snippets

When creating QoS Snippets, now they can be composed of other QoS Snippets.

Automatic validation of XML files when opening a project or adding XML files to a project

System Designer now automatically validates all the XML files when opening an existing project, and when adding an XML file to a project. You can disable this validation in the Advanced Project Properties dialog.



Improved usability of Representation QoS Policy editor

The Representation QoS Policy editor has been improved by adding a suggestion list with the following values:

- AUTO_DATA_REPRESENTATION
- XML_DATA_REPRESENTATION
- XCDR_DATA_REPRESENTATION
- XCDR2_DATA_REPRESENTATION

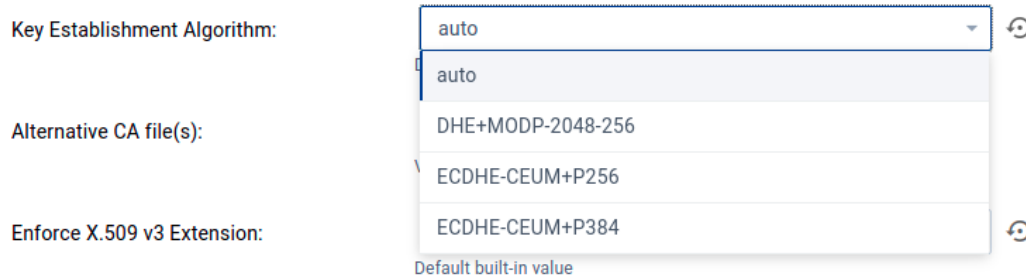
Updated Shared Secret Algorithm field in Configure Security dialog

The “Shared Secret Algorithm” field in the Configure Security dialog for a Participant QoS has been renamed to “Key Establishment Algorithm”.

New algorithms have been added and existing algorithms have been renamed:

- auto
- ECDHE-CEUM+P384 (Elliptic Curve Diffie-Hellman p384)
- Diffie-Hellman (DH) has been renamed to “DHE+MODP-2048-256”

- Elliptic Curve Diffie-Hellman (ECDH) has been renamed to “ECDHE-CEUM+P256”



New security properties in Global Security settings

The “Configure Security” dialog for a Participant QoS has been updated. The following fields have been added to the Participant Trust tab in the “Global Security Settings” dialog to configure new Security properties:

- Maximum Key Redistribution Delay in seconds
- Maximum Key Revision History Depth

Third-party software upgrades

The following third-party software used by *System Designer* has been upgraded:

Table 4.2: Third-Party Software Changes

Third-Party Software	Previous Version	Current Version
nodejs®	10.24.1	16.15.1
express-fileupload	1.2.1	1.4.0

4.5.4 What’s Fixed in 7.0.0

Unsupported QoS elements and policies appeared in Configuration

The following elements have been removed from *System Designer*, since they are not supported:

- table_allocation_block_size from DDSDatabaseQosPolicy
- writer_depth from DDSDurabilityServiceQosPolicy
- multi_channel from DDSDatareaderQosProfile
- assertions_per_lease_duration from DDSLivelinessQosPolicy only for DataReader and Topic QoS
- cpu_list from thread only for Event and Database QoS
- cpu_rotation from thread only for Event and Database QoS
- address_bit_count

[RTI Issue ID SYSD-936]

Unconfigurable QoS XML elements and policies appeared in Configuration

The following elements have been removed from *System Designer*, since they are not configurable through XML:

Participant QoS:

- `discovery_config.asynchronous_publisher.thread.cpu_list`
- `discovery_config.asynchronous_publisher.thread.cpu_rotation`
- `discovery_config.asynchronous_publisher.disable_asynchronous_batch`
- `discovery_config.asynchronous_publisher.asynchronous_batch_thread`
- `discovery_config.asynchronous_publisher.disable_topic_query_publication`
- `discovery_config.asynchronous_publisher.topic_query_publication_thread`
- `discovery_config.publication_reader.min_app_ack_response_keep_duration`
- `discovery_config.subscription_reader.min_app_ack_response_keep_duration`

Publisher QoS:

- `asynchronous_publisher.thread.cpu_list`
- `asynchronous_publisher.thread.cpu_rotation`
- `asynchronous_publisher.asynchronous_batch_thread.cpu_list`
- `asynchronous_publisher.asynchronous_batch_thread.cpu_rotation`
- `asynchronous_publisher.topic_query_publication_thread.cpu_list`
- `asynchronous_publisher.topic_query_publication_thread.cpu_rotation`
- `presentation.drop_incomplete_coherent_set`

DataReader QoS:

- `representation.compression_settings.writer_compression_level`
- `representation.compression_settings.writer_compression_threshold`

Topic QoS:

- `durability.writer_depth`

[RTI Issue ID SYSD-950]

Renamed items were not updated in System Designer's UI

References to items were not updated in *System Designer* after the item was renamed. This caused *System Designer* to generate an invalid configuration file. This problem has been resolved. Now *System Designer* will automatically update all references to the renamed item.

[RTI Issue ID SYSD-975]

Incomplete error messages when importing an XML file in some web browsers

When importing an XML file, any error messages reported by *System Designer* were incomplete in some web browsers. This problem has been resolved. Now *System Designer* will report the cause of the error if there is a problem importing an XML file.

[RTI Issue ID SYSD-978]

Deleting project files from disk may have left some empty directories

Deleting project files from disk may have left some empty directories. This occurred if any of the XML files in the project were in a different directory than the project file. This problem has been resolved.

[RTI Issue ID SYSD-984]

System Designer incorrectly allowed XML and IDL to be edited in XML and IDL views

System Designer incorrectly allowed the XML and IDL to be edited in the XML and IDL view. These changes were not reflected in the structured view. This problem has been resolved. Editing the XML directly can only be done by clicking on the Edit XML button.

[RTI Issue ID SYSD-989]

Discrepancy between range defined by System Designer and by API for certain elements

For certain elements, the range defined by *System Designer* did not match the range defined by the API. This problem has been resolved. Now, values that are supported by the API but not *System Designer*, will not appear as an error in *System Designer*, and vice versa.

[RTI Issue ID SYSD-998]

System Designer unable to edit Domain after detecting invalid Domain ID

If you entered an invalid Domain ID when creating a Domain, *System Designer* reported an error, such as “Invalid domain id”. Then *System Designer* would not allow you to further edit that domain and logged another error, such as “Domain name already used.” This problem has been resolved.

[RTI Issue ID SYSD-1000]

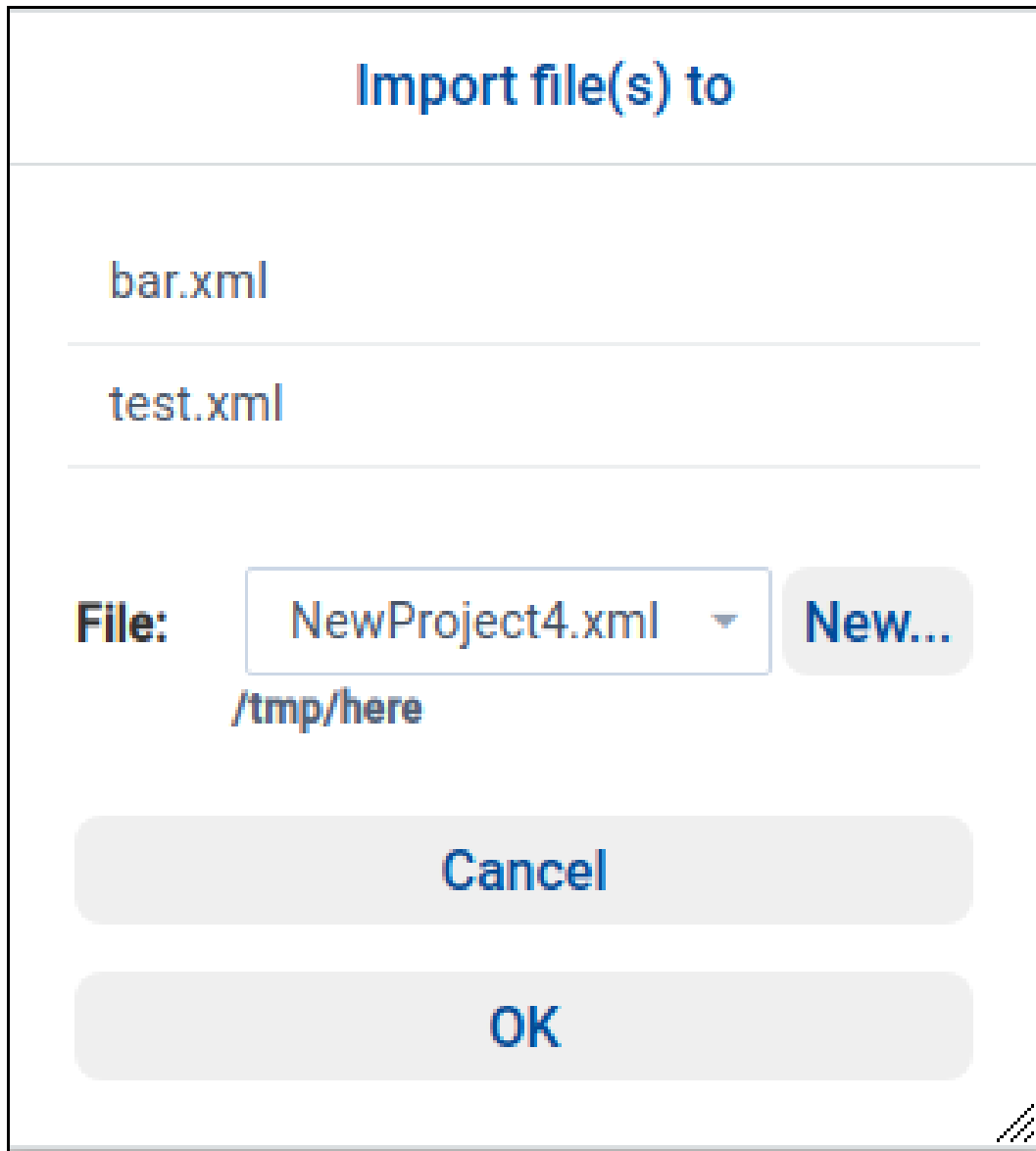
IDL and XML text editors appeared empty on network-isolated system

On a network-isolated system, *System Designer* failed to populate the IDL and XML text editors with their content; the editors appeared empty. This problem has been resolved.

[RTI Issue ID SYSD-1003]

System Designer failed to import multiple XML files

System Designer was unable to import multiple XML files. This problem has been resolved. The second window that *System Designer* displays after a file is selected has been updated to list the XML files that you are trying to import.



[RTI Issue ID SYSD-1009]

Unable to add default case to Union member when adding more than one case value

The default case could not be added to a Union member when adding more than one case value. This problem has been resolved. Now you can enter the default case when adding more than one case value.

Add Union Member

Case Value:

Name:

Type:

Press [Key Down] or start typing for suggestions

Union Member Declaration

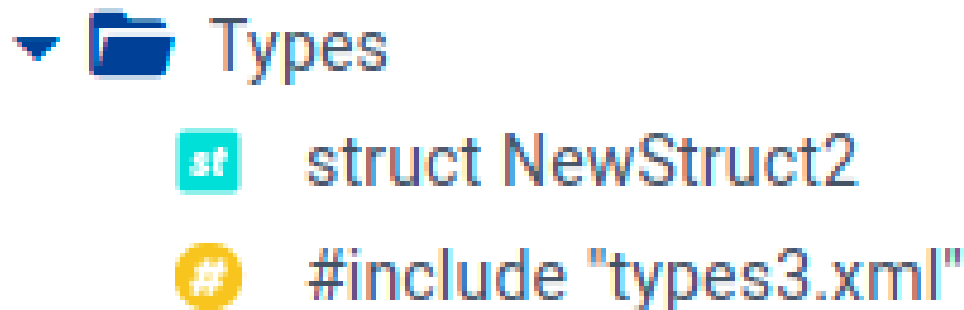
Here you define an individual cases of the union.

- To specify the **default** case of the union, just leave the 'Case value' field empty or add 'default'.
- To enter more than one case value for this member, separate the **Case values** with a comma. I.e. 1,2,3

[RTI Issue ID SYSD-1011]

Types tab did not show #include directives from XML file

System Designer's Types tab did not display any #include directives found in the XML file. This problem has been resolved. Now *System Designer* will display any #include directives from the XML file in the Types Tab.



[RTI Issue ID SYSD-1012]

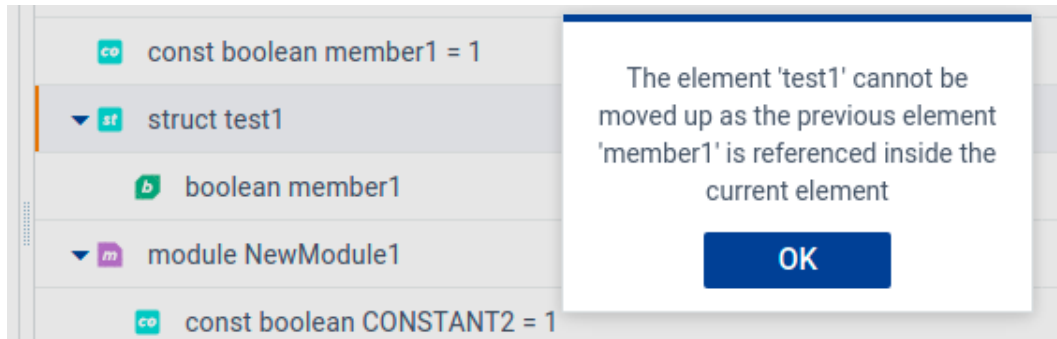
Unable to move an element up or down in Types tab, in some cases

Moving an element up or down in the Types tab did not work in in some cases.

For instance, suppose you wanted to move an element we'll call "E". It could not be moved if:

- Another element, above or below E, was not available to be moved due to its DomainParticipant-level Partition.
- Another element, above or below E, had a member with the same name as E.

- The name of another element, above or below E, was the same as one of E's members.

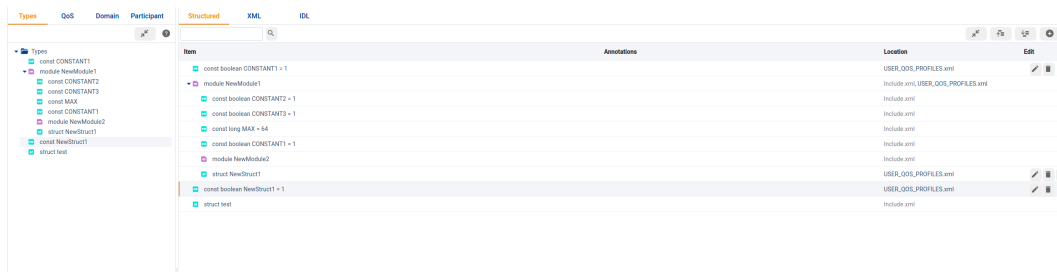


This problem has been resolved. Now *System Designer* will move the elements up or down as requested, even in the above scenarios.

[RTI Issue IDs SYSD-1016, SYSD-1018]

Failure in Types tab when trying to add module with same name as another

When defining a module in the Types tab, *System Designer* failed when trying to add a module with the same name as another module that was defined in a read-only file. This problem has been resolved (as shown in the screen capture below).



[RTI Issue ID SYSD-1017]

Could not create new file in Types tab if project contained read-only file

System Designer could not create a new file in the Types tab if the project contained a read-only file. This problem has been resolved.

[RTI Issue ID SYSD-1022]

Fixes related to vulnerabilities

This release fixes some potential vulnerabilities, described below.

Potential crash when receiving a modified form

System Designer had a third-party dependency on express-fileupload version 1.2.1. This version had a dependency on dicer, which is known to be affected by a number of publicly disclosed vulnerabilities. These vulnerabilities have been fixed by upgrading express-fileupload to the latest stable version, 1.4.0. See *Third-party software upgrades*.

For *System Designer*, the impact of using express-fileupload 1.2.1 was:

- Exploitable through a modified form
- *System Designer* could crash upon receiving the modified form. A complete denial of service can be achieved by sending the malicious form in a loop.
- CVSS v3.1 Score: 7.5 HIGH
- CVSS v3.1 Vector: AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H

[RTI Issue ID SYSD-1029]

4.6 Known Issues

Note: For an updated list of critical known issues, see the Critical Issues List on the RTI Customer Portal at <https://support.rti.com/>.

4.6.1 Modules defined in multiple files, with different annotations

Modules can be defined in multiple files, and each file might define the module with different annotations:

- @autoid
- @transfer_mode
- @language_binding
- @allowed_data_representation

Semantically, those annotations (when defined in a module) apply only to the contained entities. Since internally modules are identified by a single object, when their definition comes from multiple files, their annotations must be the same.

Chapter 5

Copyrights and Notices

© 2015-2023 Real-Time Innovations, Inc. All rights reserved. October 2023

Trademarks

RTI, Real-Time Innovations, Connex, Connex Drive, NDDS, the RTI logo, 1RTI and the phrase, “Your Systems. Working as one.” are registered trademarks, trademarks or service marks of Real-Time Innovations, Inc. All other trademarks belong to their respective owners.

Copy and Use Restrictions

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form (including electronic, mechanical, photocopy, and facsimile) without the prior written permission of Real-Time Innovations, Inc. The software described in this document is furnished solely under and subject to RTI’s standard terms and conditions available at <https://www.rti.com/terms> and in accordance with your License Acknowledgement Certificate (LAC) and Maintenance and Support Certificate (MSC), except to the extent otherwise accepted in writing by a corporate officer of RTI.

Third-Party Software

RTI software may contain independent, third-party software or code that are subject to third-party license terms and conditions, including open source license terms and conditions. Copies of applicable third-party licenses and notices are located at community.rti.com/documentation. IT IS YOUR RESPONSIBILITY TO ENSURE THAT YOUR USE OF THIRD-PARTY SOFTWARE COMPLIES WITH THE CORRESPONDING THIRD-PARTY LICENSE TERMS AND CONDITIONS.

Notices

Deprecations and Removals

Any deprecations or removals noted in this document serve as notice under the Real-Time Innovations, Inc. Maintenance Policy #4220 and/or any other agreements by and between RTI and customer regarding maintenance and support of RTI's software.

Deprecated means that the item is still supported in the release, but will be removed in a future release. *Removed* means that the item is discontinued or no longer supported. By specifying that an item is deprecated in a release, RTI hereby provides customer notice that RTI reserves the right after one year from the date of such release and, with or without further notice, to immediately terminate maintenance (including without limitation, providing updates and upgrades) for the item, and no longer support the item, in a future release.

Technical Support Real-Time Innovations, Inc. 232 E. Java Drive Sunnyvale, CA 94089 Phone: (408) 990-7444 Email: support@rti.com Website: <https://support.rti.com/>