

# ***RTI System Designer***

**(Prototype Feature)**

## **Getting Started Guide**

Version 0.1



Your systems. Working as one.



© 2013-2017 Real-Time Innovations, Inc.

All rights reserved.

Printed in U.S.A. First Printing.

June 2015.

### **Trademarks**

Real-Time Innovations, RTI, DataBus, and Connexx are trademarks or registered trademarks of Real-Time Innovations, Inc. All other trademarks used in this document are the property of 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 under and subject to the RTI software license agreement. The software may be used or copied only under the terms of the license agreement.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.



# Contents

[Introduction](#)

[Installing the RTI System Designer](#)

[Trying RTI System Designer](#)

[Online Demo](#)

[Download it](#)

[Start it](#)

[Using the RTI System Designer](#)

[First Execution](#)

[Use existing XML File](#)

[Upload an XML File](#)

[Importing an XML file](#)

[Online Help](#)

[Workflow](#)

[Exploring the Tabs](#)

[The Types TAB](#)

[Exploring Referenced Types](#)

[Add Local types](#)

[The QoS TAB](#)

[The Domain Tab](#)

[Exploring Referenced Domain Libraries](#)

[Adding Local Domains](#)

[Register a new Type](#)

[Register a new Topic](#)

[Participant Library TAB](#)

[Exploring Referenced Participant Library](#)

[Adding Local Participants](#)

[Export an XML Application Creation File](#)

[Additional Notes about RTI System Designer](#)

[Release Notes](#)

[What's New/Fixed in this Version](#)

[Known Issues and Limitations](#)

[System Designer](#)

[Open Source Content](#)

# Introduction

This document assumes you have a basic understanding of RTI® Connex<sup>TM</sup> DDS application development and concepts, such as a DDS Domain, DomainParticipant, Topic, DataWriter and DataReader. For an overview of these concepts, please see the ***RTI Connex<sup>TM</sup> DDS Core Libraries Getting Started Guide***.

This Tool is based on RTI® XML-Based Application Creation - a mechanism to simplify the development of Connex<sup>TM</sup> DDS applications. For an overview of its functionality please see the ***RTI Connex<sup>TM</sup> DDS Core Libraries XML-Based Application Creation Getting Started Guide***.

*RTI System Designer* is a tool that allows you to graphically design and configure Connex<sup>TM</sup> DDS systems. You can see it as a User Interface to XML-Based Application Creation, a technology that allow 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* present within each *DomainParticipant*, each associated with its corresponding *Topic*.

The application code simply indicates the participant configuration name of the *DomainParticipant* that the application 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.

Writing this XML can be quite challenging. Using a good editor with autocompletion capabilities along with the XML Schema provided can be of great help. But it is still quite complicated, especially when as the project gets bigger and involves multiple files.

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

## Installing the RTI System Designer

The *RTI System Designer* is distributed as a zip file and is available for the following architectures:

- Darwin 64 bit (Mac OsX)
- Windows 32 and 64 bit
- Linux 32 and 64 bit

Once you unzip the bundle, you can cd into the **bin** directory and run the `rtiddssystemdesigner` script.

*RTI System Designer* has a server component and a UI that will run inside a browser. The script will both start the server and it will try to open your default browser pointing to the right address.

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

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

The *RTI System Designer* allows to create and to import existing XML files containing:

- QoS
- Types
- Domain Libraries
- Participant Library

## Trying RTI System Designer

### Online Demo

You can access a live version of the *RTI System Designer* without installing it by visiting

<https://community.rti.com/downloads/experimental/system-designer>

This will allow you to test a demo version of the *RTI System Designer* and learn more about it before you even download it and install it.

### Download it

Once you are on the online demo you can will see a button called “Get it now!” that will allow you, after a short registration, to get a copy of the tool:



## Start it

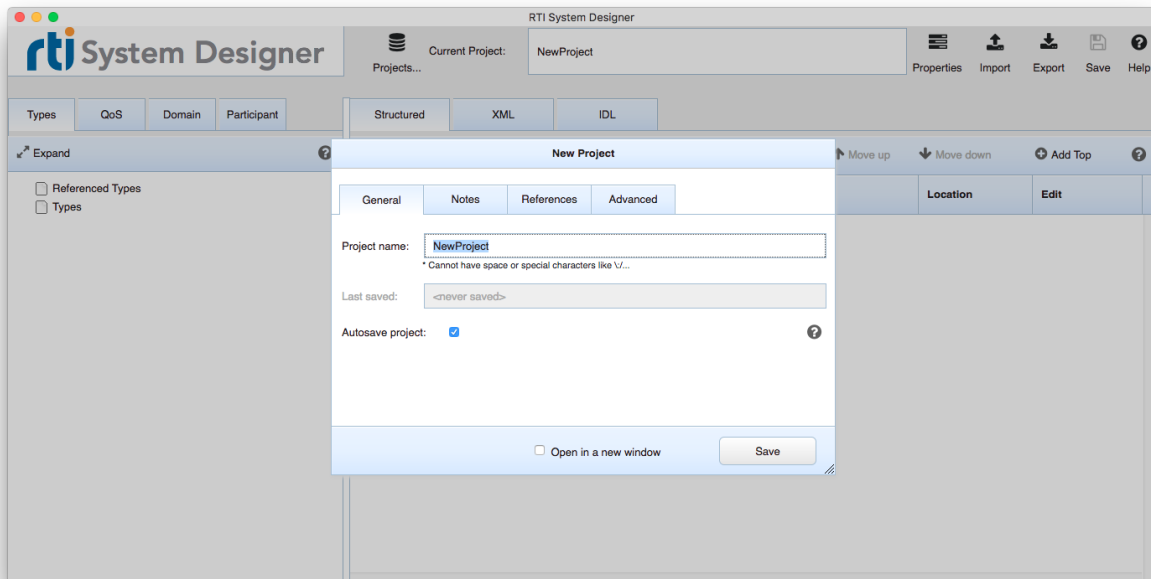
Once you downloaded the tool, all you have to do is to execute the script in the bin directory:

```
./bin/rtiddssystemdesigner[.bat]
```

## Using the RTI System Designer

### First Execution

The first time you start the *RTI System Designer*, the application will prompt you to create a new project as in the following screenshot:



Just insert a Project Name and Click Save.

### Use existing XML File

There are two supported ways to use existing XML files:

- **Upload** an XML file: the content of XML files **uploaded** into the UI tool will be



part of the **Referenced** participants, domains, types and QoSs and they **cannot** be modified.

- **Import** an XML file: the content of XML files **imported** into the UI tool will be part of the **Local** participants, domains, types and QoSs and **can** be modified.

## Upload an XML File

In this example we will upload an existing XML file that you can find in the example directory:

```
<install directory>/example/ShapesExample.xml
```

This file contains:

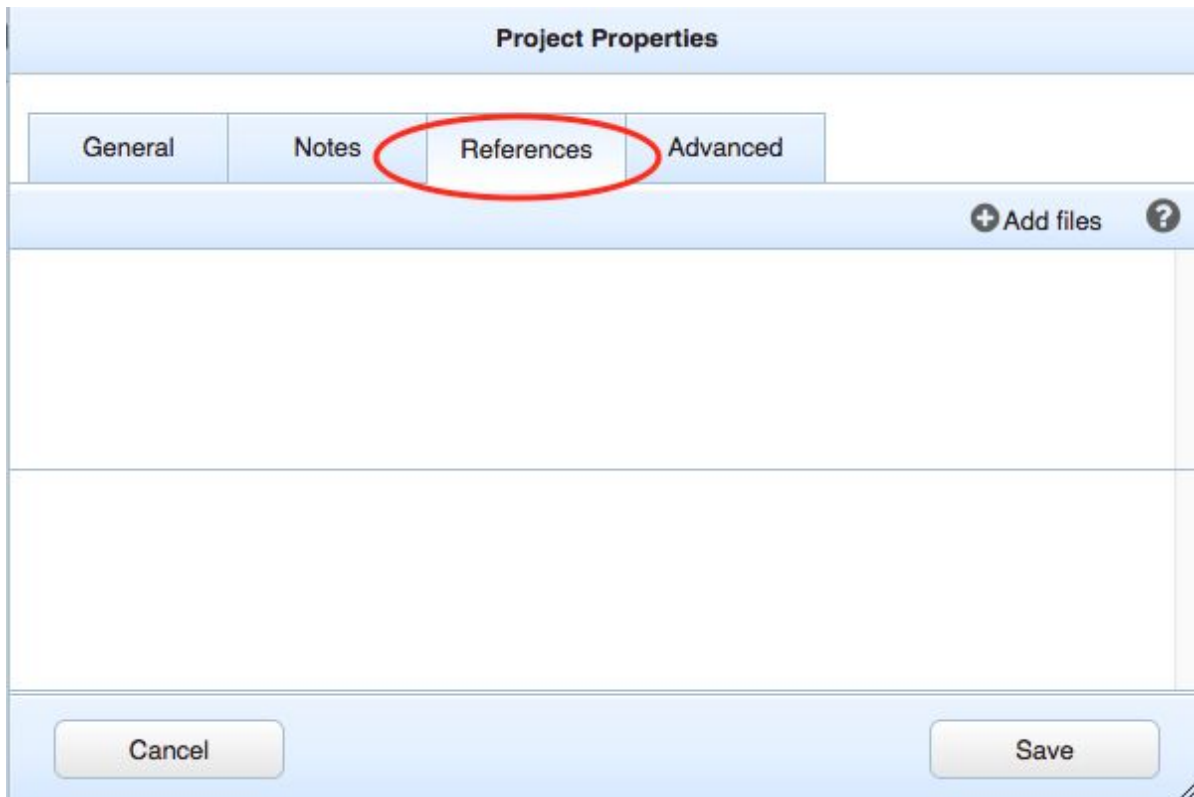
- Type Definition
- QoS definition
- Domain Library
- Participant Library

All these elements will be added to the Imported sections and the user will be able to browse and use them in his project, but not modify them.

To upload the content of this file inside the UI, select the **Properties** button in the top bar:

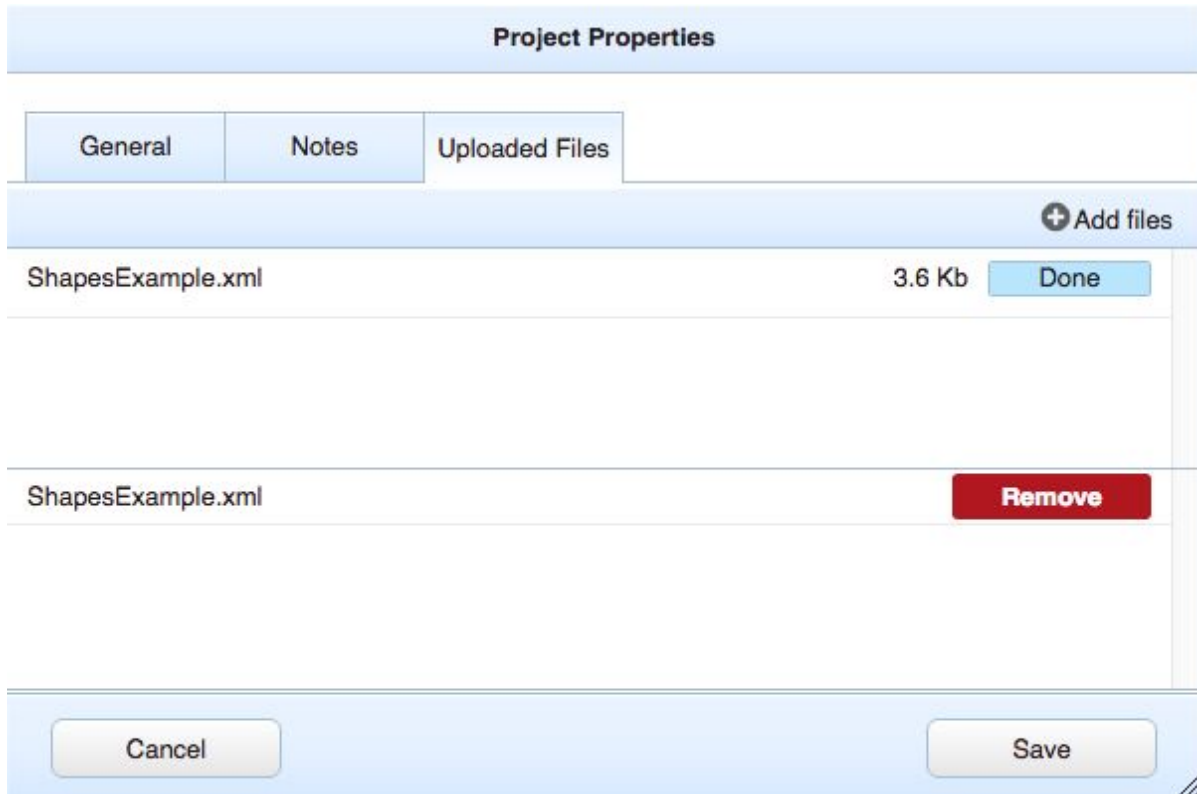


And then select the **References** Tab:



You can now select the **+ Add files** button to access a file system browser and select the file to upload.

During the upload you can see a status bar next to the file being uploaded. Once the file is uploaded it will be appear in the bottom part of the popup:



You can always come back to this windows to remove file by clicking on the **Remove** button next to the File Name.

After the you are done, you can select “Save” to apply the changes to the model.

### Importing an XML file

In this example we will upload an existing XML file that you can find in the example directory:

```
<install directory>/example/ShapesExtExample.xml
```

This file contains:

- A new type definition
- A New Domain Library

All these elements will be added to the Local sections and the user will be able to browse, use and modify them.




To Import a file, just select the **Import** button from the Top Bar:



A file system browser will be presented; select the file and click ok. The elements should have been added to the relative tabs and be ready to be modified.

## Online Help

Using the tool you will often see an help icon: . You can click that icon to see online tips and to understand what to do next.

## Workflow

The *RTI System Designer* has four tabs. They each represent one of the sections in the XML-Based Application Creation. We suggest you start defining a Type and proceed with the definition of the QoS. Once you are done you can move to the Domain Tab, register your types and declare your topic. At the end you can, in the Participant Tab, define your participant along with readers and writers.

Of course you can always, and probably you will, go back and forth to tweak the system.

## Exploring the Tabs

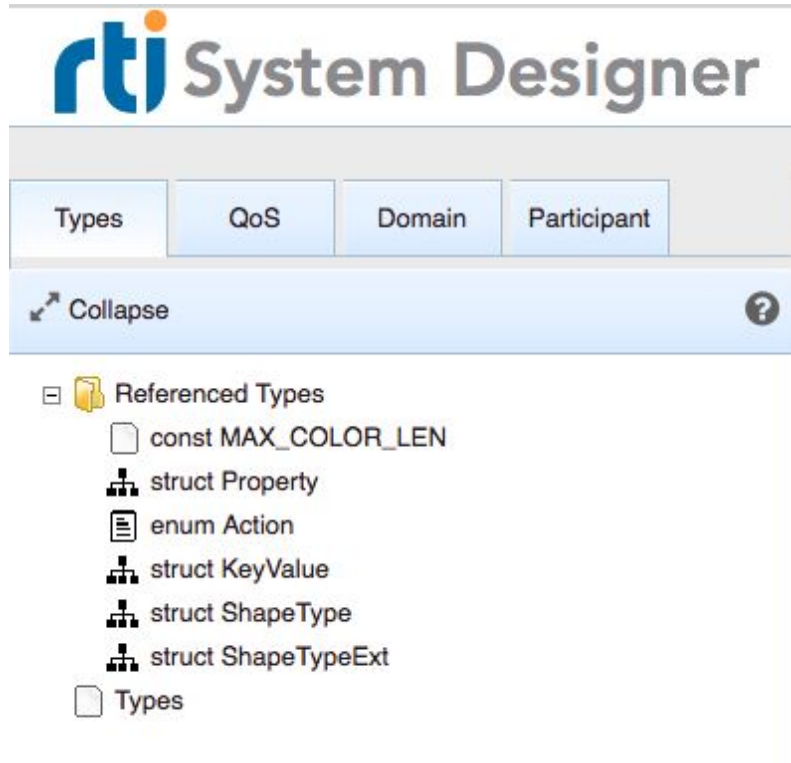
### The Types TAB

In this tab you can visualize and edit all the types that are going to be part of your DDS system.

#### Exploring Referenced Types

Once you uploaded a file as shown above, you will notice, in the left panel, a new tree called "Referenced Types" under the "Type Tab":


You can use the Expand/Collapse button to explore your imported types:



For a more detailed view of the types you can look at the main panel: it has 3 tabs:

- Structured View
- XML View
- IDL View

Expand			
		Move up	Move down
		Add Top	?
Item	Annotations	Location	Edit
<input type="checkbox"/> const long MAX_COLOR_LEN = 128		ShapesExample.xml:49	
<input checked="" type="checkbox"/> struct Property		ShapesExample.xml:51	
<input checked="" type="checkbox"/> enum Action		ShapesExample.xml:55	
<input checked="" type="checkbox"/> struct KeyValue		ShapesExample.xml:69	
<input checked="" type="checkbox"/> struct ShapeType		ShapesExample.xml:73	
<input checked="" type="checkbox"/> struct ShapeTypeExt: ShapeType	extensibility=extensible	ShapesExample.xml:81	

The Expand/Collapse  button will allow you to go as deep as needed when exploring types:

<div> <span>⌵ Collapse</span> <span>⬆ Move up</span> <span>⬇ Move down</span> <span>➕ Add Top</span> <span>?</span> </div>			
Item	Annotations	Location	Edit
<input type="checkbox"/> const long MAX_COLOR_LEN = 128		ShapesExample.xml:49	
<div> <div> <span>⌵</span> <span>⌵</span> </div> <div>struct Property</div> </div>		ShapesExample.xml:51	
<div> <div> <span>⬤</span> </div> <div>string&lt;MAX_</div> </div> <div>struct Property</div>		ShapesExample.xml:52	
<div> <div> <span>⬤</span> </div> <div>string&lt;MAX_COLOR_LEN&gt; value</div> </div>		ShapesExample.xml:53	
<div> <div> <span>⌵</span> <span>⌵</span> </div> <div>enum Action</div> </div>		ShapesExample.xml:55	
<input type="checkbox"/> NORMAL = 1		ShapesExample.xml:56	
<input type="checkbox"/> QUARANTINE = 2		ShapesExample.xml:57	
<input type="checkbox"/> ISOLATE = 3		ShapesExample.xml:58	
<input type="checkbox"/> THWART = 4		ShapesExample.xml:59	
<div> <div> <span>⌵</span> <span>⌵</span> </div> <div>struct KeyValue</div> </div>		ShapesExample.xml:69	
<div> <div> <span>⬤</span> </div> <div>string&lt;10&gt; k</div> </div>	ID=0	ShapesExample.xml:70	
<div> <div> <span>⬤</span> </div> <div>string v</div> </div>	ID=1	ShapesExample.xml:71	
<div> <div> <span>⌵</span> <span>⌵</span> </div> <div>struct ShapeType</div> </div>		ShapesExample.xml:73	
<div> <div> <span>⬤</span> </div> <div>short a</div> </div>	ID=0	ShapesExample.xml:74	
<div> <div> <span>⬤</span> </div> <div>string b</div> </div>	ID=1	ShapesExample.xml:75	
<div> <div> <span>⬤</span> </div> <div>sequence&lt;short, 10&gt; c</div> </div>	ID=2	ShapesExample.xml:76	
<div> <div> <span>⬤</span> </div> <div>double d[2]</div> </div>	ID=3	ShapesExample.xml:77	
<div> <div> <span>⬤</span> </div> <div>KeyValue e[5]</div> </div>	ID=4	ShapesExample.xml:78	
<div> <div> <span>⌵</span> <span>⌵</span> </div> <div>struct ShapeTypeExt: ShapeType</div> </div>	extensibility=extensible	ShapesExample.xml:81	
<div> <div> <span>⬤</span> </div> <div>sequence&lt;Property, 3&gt; properties</div> </div>		ShapesExample.xml:84	

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, if it is a key or if it is an optional or extensible member.

The **Location** column will indicate in which file and at which line member is defined.

The **Edit** column is only available for Local types.

The XML View will show the types in the XML format. You can highlight a specific member by selecting it in the left tree panel:

The XML tab displays the following code:

```
<const name="MAX_COLOR_LEN" type="long" value="128"/>
<struct name="Property">
  <member name="name" type="string" stringMaxLength="MAX_COLOR_LEN"/>
  <member name="value" type="string" stringMaxLength="MAX_COLOR_LEN"/>
</struct>
<enum name="Action">
  <enumerator name="NORMAL" value="1"/>
  <enumerator name="QUARANTINE" value="2"/>
  <enumerator name="ISOLATE" value="3"/>
  <enumerator name="THWART" value="4"/>
</enum>
<struct name="KeyValue">
  <member name="k" type="string" stringMaxLength="10" id="0"/>
  <member name="v" type="string" id="1"/>
</struct>
<struct name="ShapeType">
  <member name="a" type="short" id="0"/>
  <member name="b" type="string" id="1"/>
  <member name="c" type="short" sequenceMaxLength="10" id="2"/>
  <member name="d" type="double" arrayDimensions="2" id="3"/>
  <member name="e" type="nonBasic" nonBasicTypeName="KeyValue" arrayDimensions="5" id="4"/>
</struct>
<struct name="ShapeTypeExt" baseType="ShapeType" extensibility="extensible">
  <member name="properties" type="nonBasic" nonBasicTypeName="Property" sequenceMaxLength="3"/>
</struct>
```

Finally, the IDL Tab will show you the IDL definition of the type:

The IDL tab displays the following code:

```
const long MAX_COLOR_LEN = 128;

struct Property {
  string<MAX_COLOR_LEN> name;
  string<MAX_COLOR_LEN> value;
};

enum Action {
  NORMAL = 1,
  QUARANTINE = 2,
  ISOLATE = 3,
  THWART = 4
};

struct KeyValue {
  string<10> k; // @Id=0
  string v; // @Id=1
};

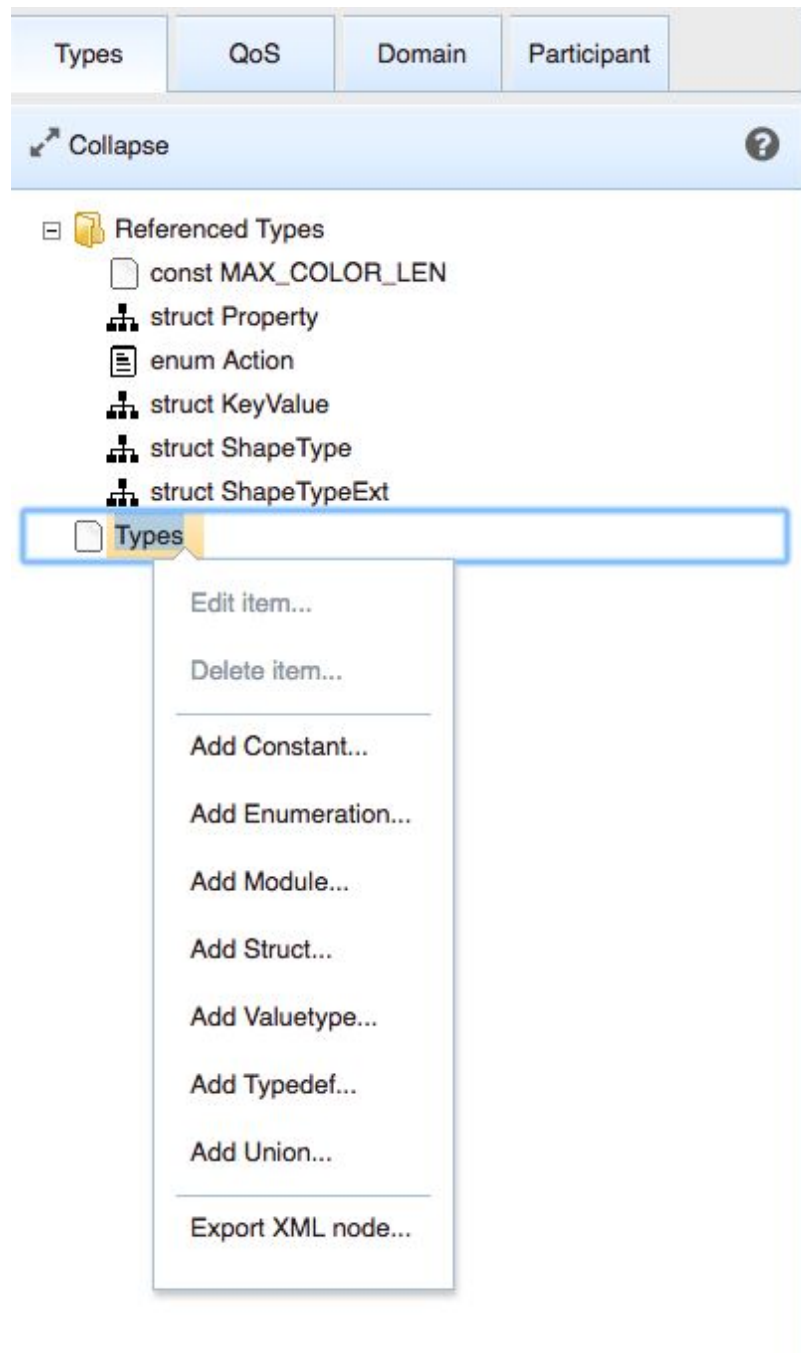
struct ShapeType {
  short a; // @Id=0
  string b; // @Id=1
  sequence<short, 10> c; // @Id=2
  double d[2]; // @Id=3
  KeyValue e[5]; // @Id=4
};

struct ShapeTypeExt : ShapeType {
  sequence<Property, 3> properties;
}; // @Extensibility extensible
```

## Add Local types

To introduce new types in your project, just right click on the “Types” node in the Tree

View and add one of the top level types:



A pop up will guide you on how to create a new top level type:





Name:

NewMember1

Type:

|

long

longLong

longDouble

☐ Sequ
 ☐ Array
 ☐ Pointer

☐ @key
 ☒ @resolve-name
 ☐ @id
 ☐ @optional

Add more...








OK

Member Declaration

Here you define an individual member of your struct.

- The **@key** attribute let you specify that this member is part of a key used in your type.
- Bitfields are not supported.
- Optional properties are not supported.
- IDs are optionally used to specify the ordering of this member relative to the other members of the struct.
- The **@resolve-name** directive tells rtdsgen to attempt to resolve the type of this member (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). 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.

This is how the new type would look after adding a couple of members:

Expand			
Item	Annotations	Location	Edit
<input type="checkbox"/> const long MAX_COLOR_LEN = 128		ShapesExample.xml:49	
<input checked="" type="checkbox"/> struct Property		ShapesExample.xml:51	
<input checked="" type="checkbox"/> enum Action		ShapesExample.xml:55	
<input checked="" type="checkbox"/> struct KeyValue		ShapesExample.xml:69	
<input checked="" type="checkbox"/> struct ShapeType		ShapesExample.xml:73	
<input checked="" type="checkbox"/> struct ShapeTypeExt: ShapeType	extensibility=extensible	ShapesExample.xml:81	
<input checked="" type="checkbox"/> struct NewStruct1			  
<input checked="" type="checkbox"/> long NewMember1			 
<input checked="" type="checkbox"/> string<255> NewMember2			 

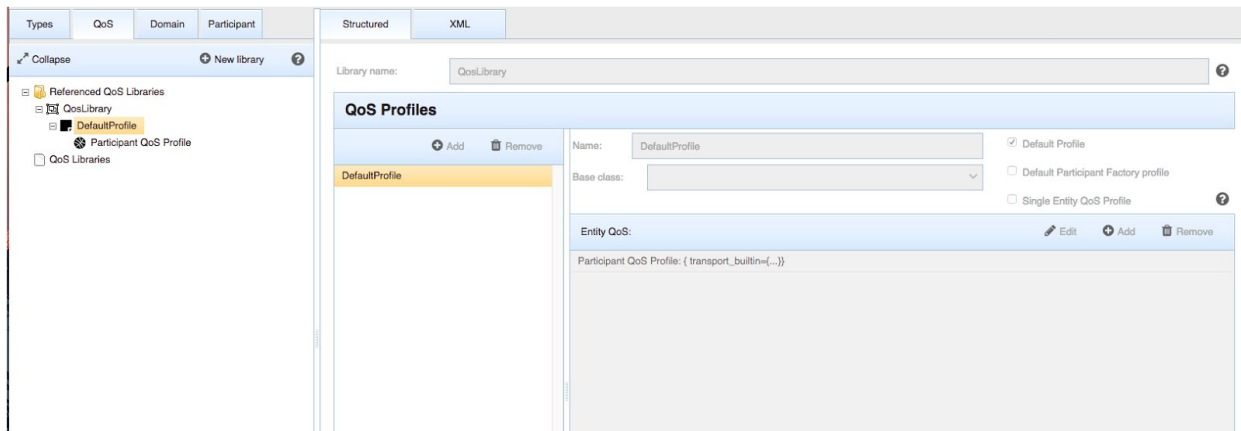
You can change the position of the members using the



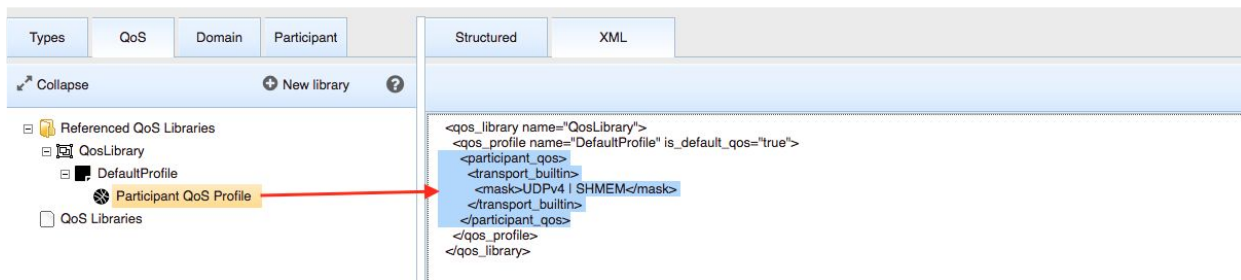
## The QoS TAB

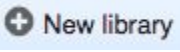
As all the tabs, the QoS Tab, has a tree on the left panel that will show you QoS coming from the references files and the qos Created in the Local project.

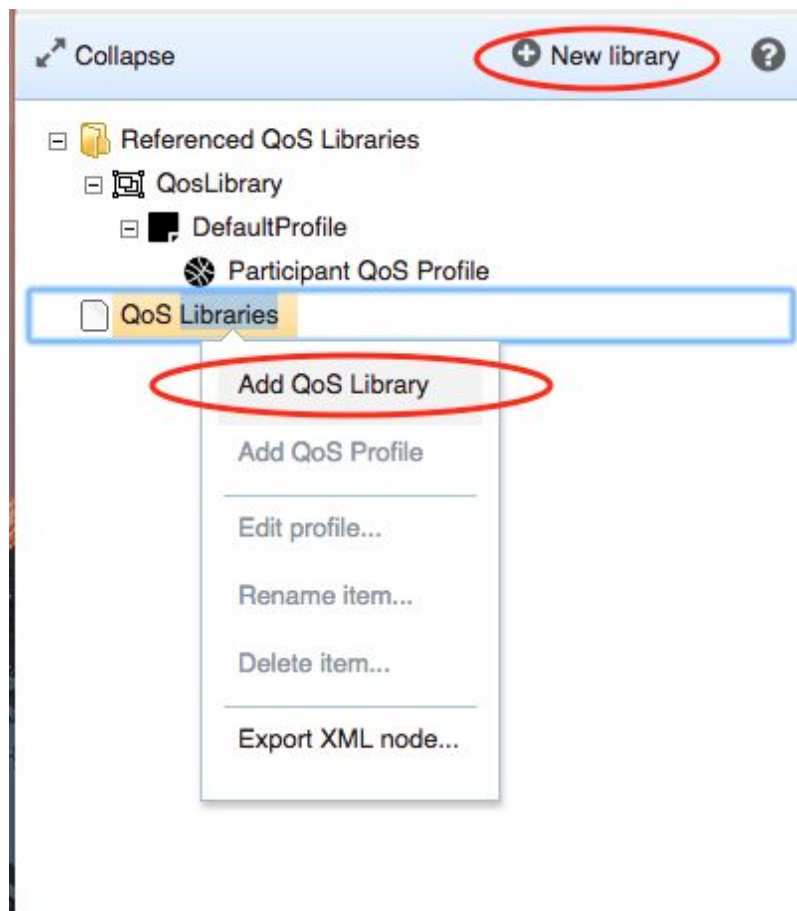
You can select a Profile on the right side and see the details on the main panel. Both in a structured view:




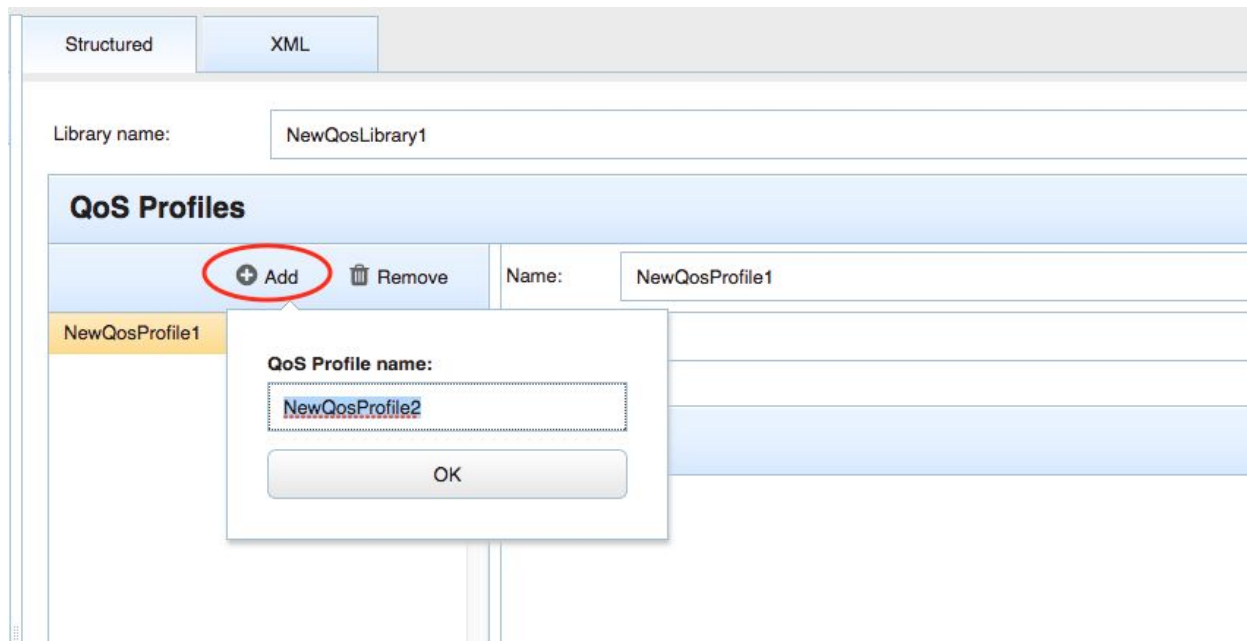
Or in the XML format:



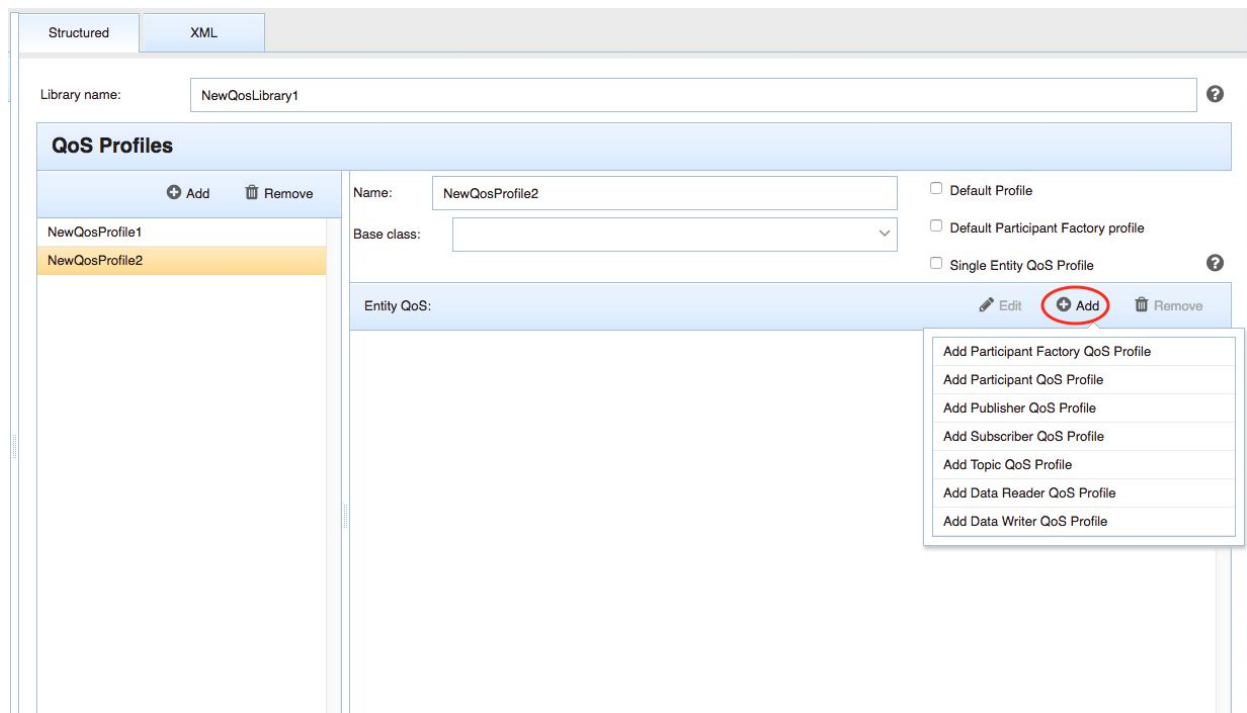
To add new QoS to your local configuration, you can click the  button or right click on the QoS Libraries tree:




You can then add QoS Profiles to that library using the  button in the main panel:



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



When an Entity is selected, a popup will be presented that will allow you to set all the QoS related to that Entity. You can browse all of them and select  to set and get more informations on a specific setting:

Edit Data Reader QoS Profile from profile 'NewQosProfile2'

Profile name: (optional)

Base QoS:

Topic filter: (optional)

Expand Modified QoS
Reset

Liveliness (Liveliness QoS Policy)

Reliability (Reliability QoS Policy)

Destination order (Destination Order QoS Policy)

History (History QoS Policy)

Configures the number of DDS samples that Connex DDS will store locally for DataWriters and DataReaders. ... [\[More...\]](#)

Kind:

Depth:

Refilter:

Resource limits (Resource Limits QoS Policy)

User data (User Data QoS Policy)

Ownership (Ownership QoS Policy)

Cancel OK

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:

History (History QoS Policy) ^

Configures the number of DDS samples that Connex DDS will store locally for DataWriters and DataReaders. ... [\[More...\]](#)

Kind:

Depth:

Refilter: 

KEEP\_LAST\_HISTORY\_QOS

KEEP\_ALL\_HISTORY\_QOS

Resource limits (Res

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

Reliability (Reliability QoS Policy) v

Destination order (Destination Order QoS Policy) v

History (History QoS Policy) ^

Configures the number of DDS samples that Connex DDS will store locally for DataWriters and DataReaders. ... [\[More...\]](#)

Kind: 

KEEP\_LAST\_HISTORY\_QOS

Depth:

Refilter:

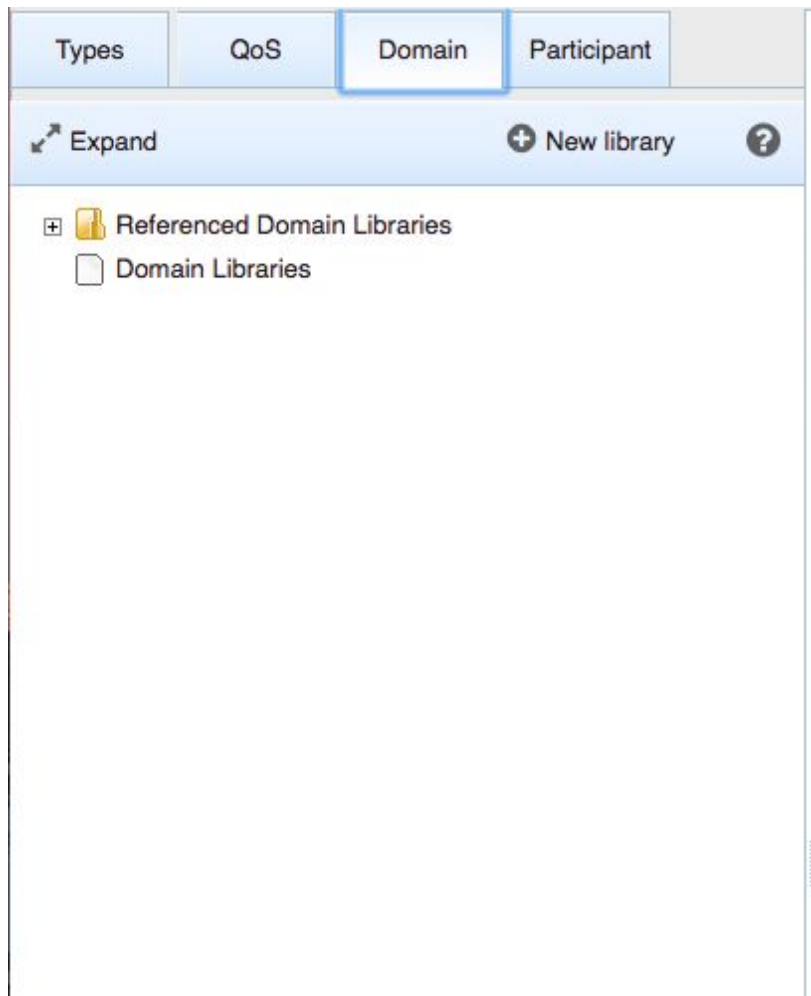
Resource limits (Resource Limits QoS Policy) v

User data (User Data Qos Policy) v

## The Domain Tab

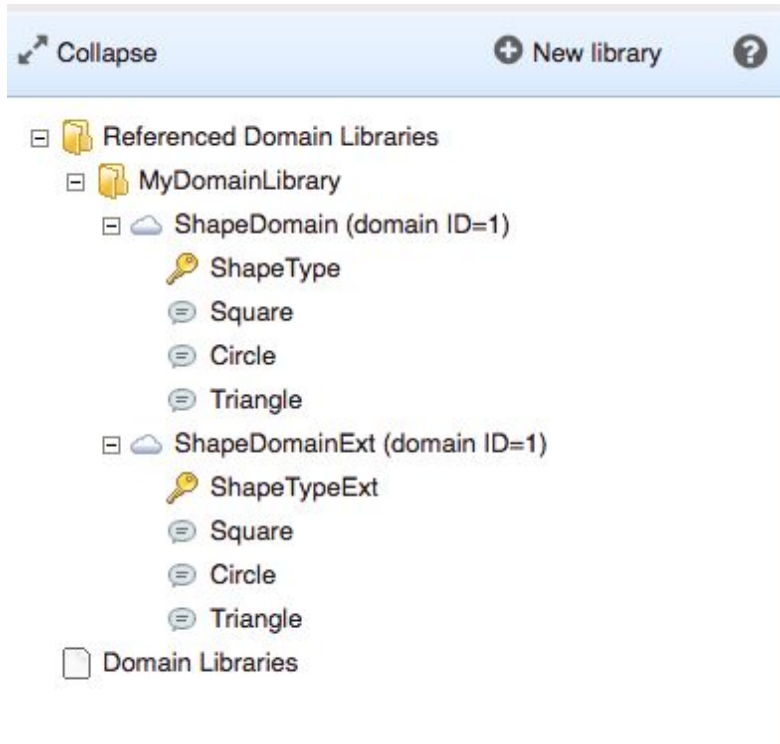
### Exploring Referenced Domain Libraries

Once you uploaded a file as shown above, you will notice, in the left panel, a new tree called “Referenced Domain Libraries” under the “Domain Tab”:



You can expand/collapse the whole tree using the “Expand/Collapse” button on top of the page:





You can then select any element on the tree and see the details on the main panel. Both in a structured view:

The screenshot shows the application interface with the 'Structured' view selected. A red arrow points from the 'Square' element in the tree view to the 'Topics' table in the main panel.

**Domain name:** ShapeDomain **Domain ID:** 1

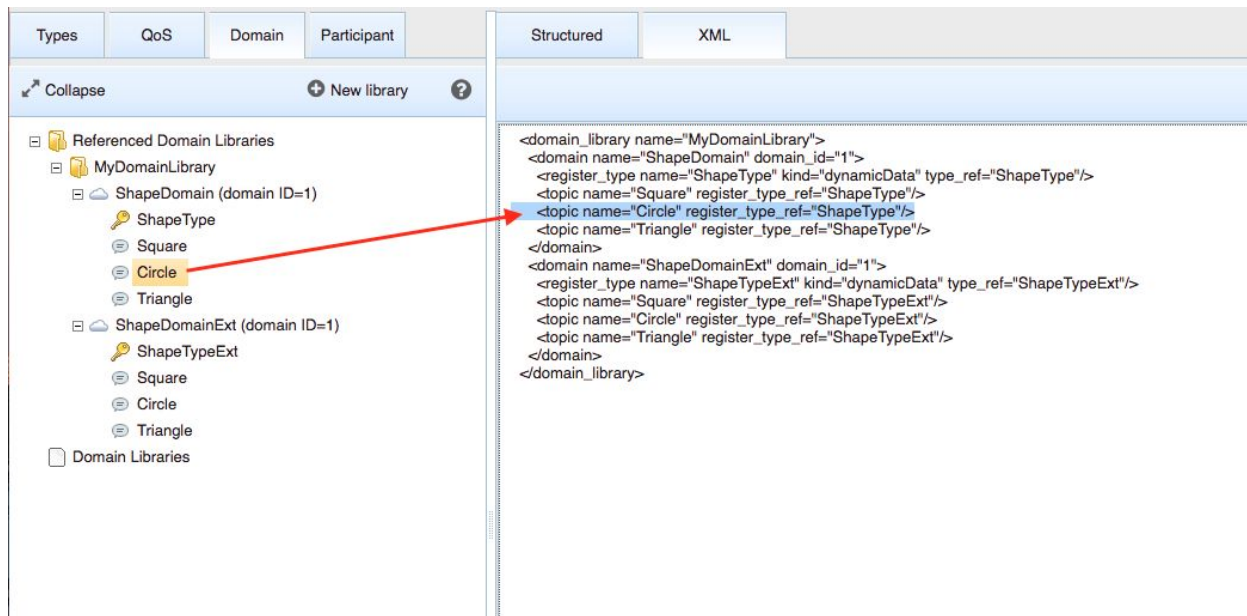
**Registered types**

Registered Name	Kind	Type Name
ShapeType	dynamicData	ShapeType

**Topics**

Topic Name	Registered Type Name	QoS
Square	ShapeType	Default QoS
Circle	ShapeType	Default QoS
Triangle	ShapeType	Default QoS

Or in the XML Format:

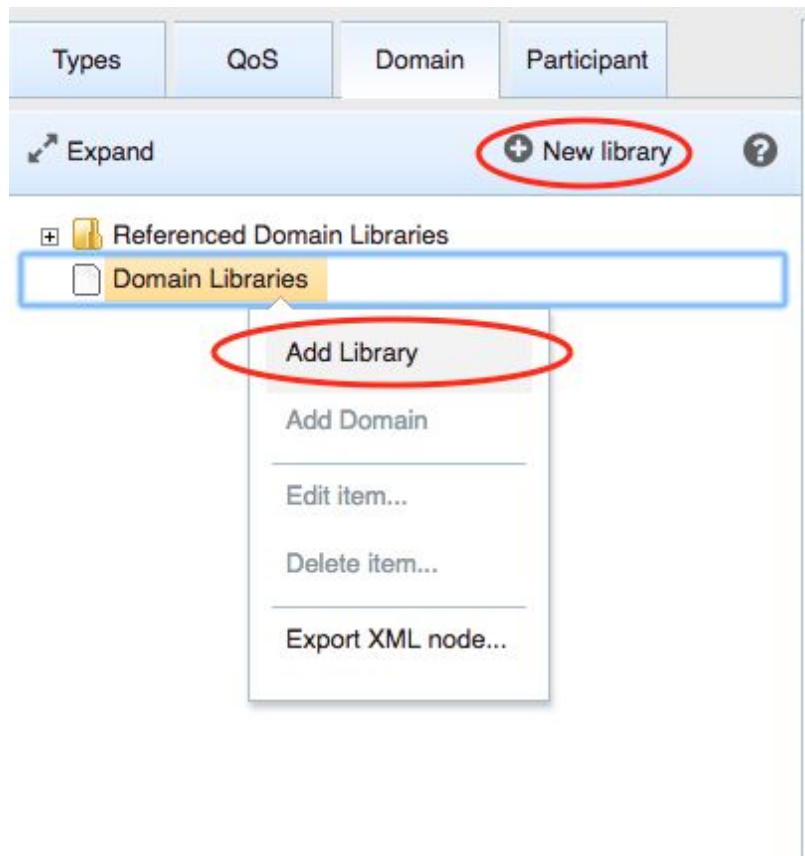


## Adding Local Domains

As you can see, the imported example has a Domain called *Shape Domain* that contains a *ShapeType* and three topics *Square*, *Circle* and *Triangle*.

Let's see how you can create a new Domain Library containing a new Type and a new Topic.

To start, right click on the "Domain Libraries" tree or use the  button:



A dialog box will let you pick a name. In this example will call it *NewDomainLibrary1*. Once the library has been added, right click on it and choose “Add Domain”. A dialog box will be presented:

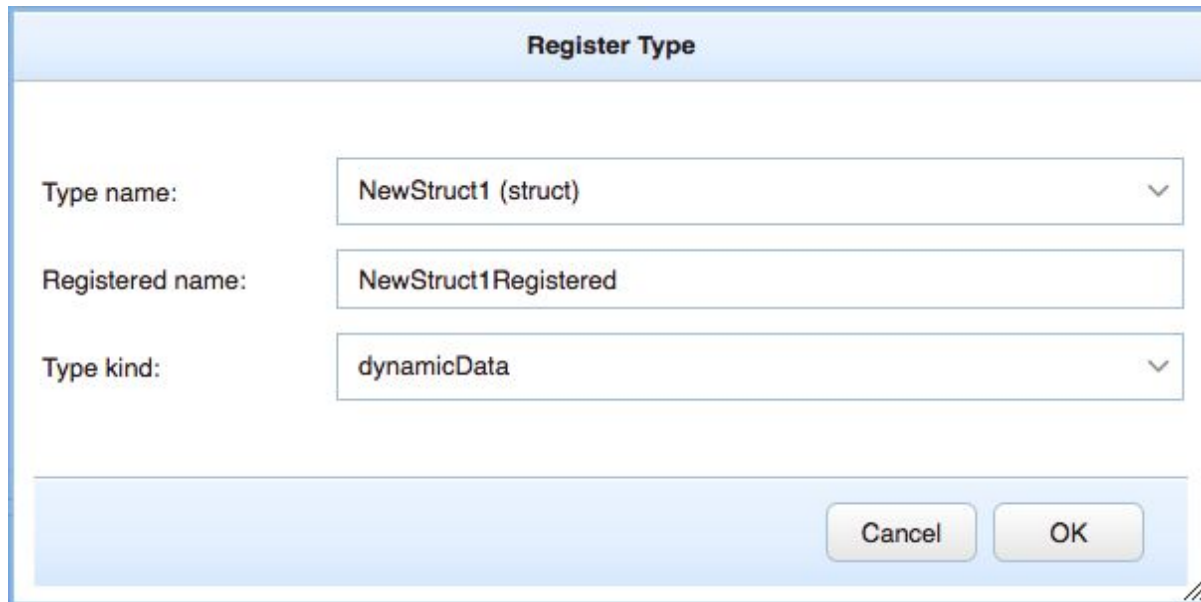
The image shows a dialog box titled 'Domain'. It has two input fields: 'Domain name:' with the text 'NewDomain1' and 'Domain ID:' with the text '0'. At the bottom of the dialog box are two buttons: 'Cancel' and 'OK'.

In this dialog box you can pick the name (*NewDomain1*) and you can select the domain

id (0).

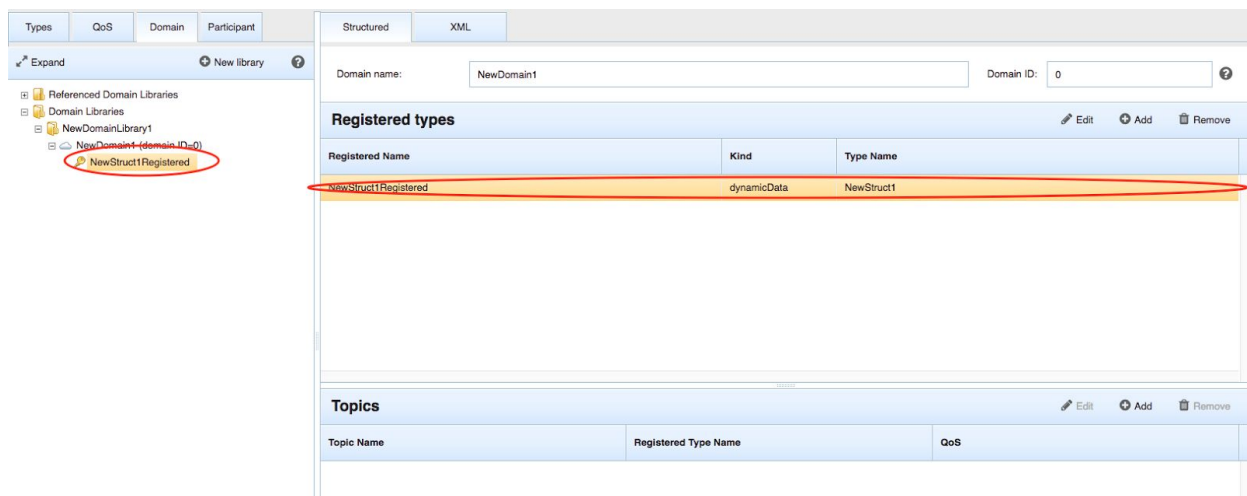
## Register a new Type

To Register a new type, select **+ Add** in the **Registered types** table; a pop-up will be presented allowing the user to choose a name, a kind and the type name from a list:



The 'Register Type' dialog box contains three input fields: 'Type name' with a dropdown menu showing 'NewStruct1 (struct)', 'Registered name' with a text field containing 'NewStruct1Registered', and 'Type kind' with a dropdown menu showing 'dynamicData'. At the bottom right are 'Cancel' and 'OK' buttons.

Once OK is clicked, you will see that the new type is visible in the left tree and in the Registered type table in the context panel:



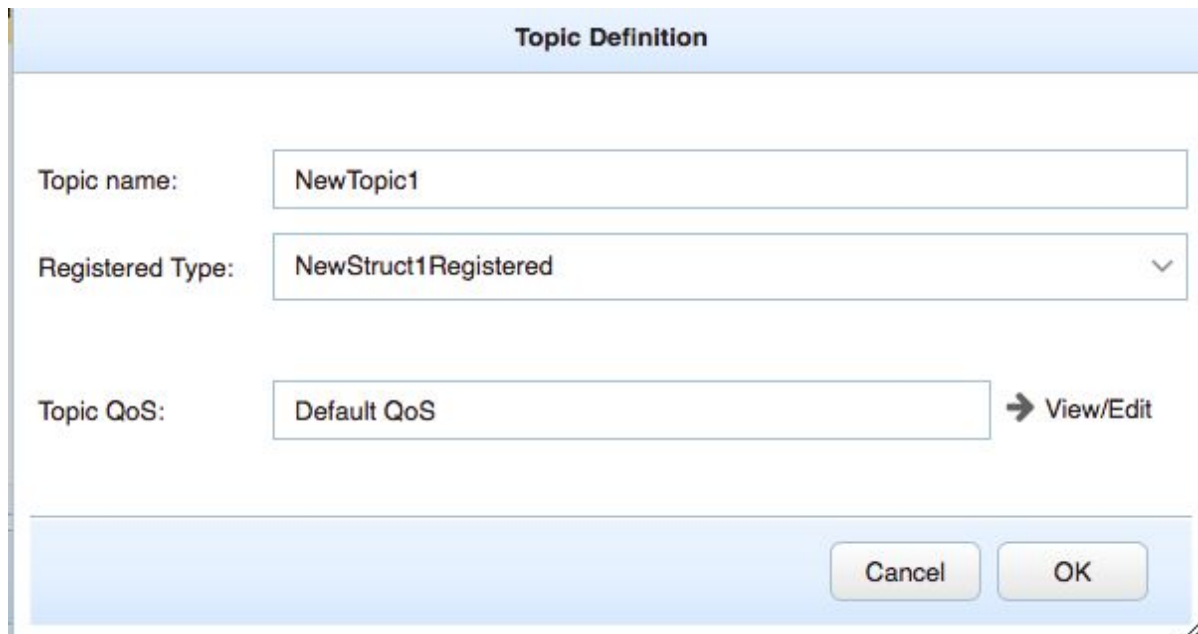
The interface shows a left-hand tree view and a right-hand context panel. In the tree view, under 'Domain Libraries', 'NewDomain1' is expanded, showing 'NewStruct1Registered' circled in red. The context panel has tabs for 'Structured' and 'XML'. The 'Registered types' table is active, showing a single row with 'NewStruct1Registered' in the 'Registered Name' column, 'dynamicData' in the 'Kind' column, and 'NewStruct1' in the 'Type Name' column. This row is also circled in red. Below it is a 'Topics' table with columns 'Topic Name', 'Registered Type Name', and 'QoS', which is currently empty.

Registered Name	Kind	Type Name
NewStruct1Registered	dynamicData	NewStruct1

Topic Name	Registered Type Name	QoS
------------	----------------------	-----

## Register a new Topic

To register a new Topic just click  Add in the Topics table. A pop up will be presented:



The image shows a 'Topic Definition' dialog box. It has a light blue header with the title 'Topic Definition'. Below the header, there are three input fields. The first is 'Topic name:' with a text box containing 'NewTopic1'. The second is 'Registered Type:' with a dropdown menu showing 'NewStruct1Registered'. The third is 'Topic QoS:' with a text box containing 'Default QoS' and a 'View/Edit' link to its right. At the bottom right of the dialog are 'Cancel' and 'OK' buttons.

Topic Definition	
Topic name:	<input type="text" value="NewTopic1"/>
Registered Type:	<input type="text" value="NewStruct1Registered"/>
Topic QoS:	<input type="text" value="Default QoS"/> <a href="#">View/Edit</a>
<div>Cancel OK</div>	

Input your topic name and select one of the available Registered types from the drop down menu.

Once you click ok, the new topic will be available on the left tree and in the context panel:

Domain name:  Domain ID:

**Registered types**

Registered Name	Kind	Type Name
NewStruct1Registered	dynamicData	NewStruct1

**Topics**

Topic Name	Registered Type Name	QoS
NewTopic1	NewStruct1Registered	Default QoS

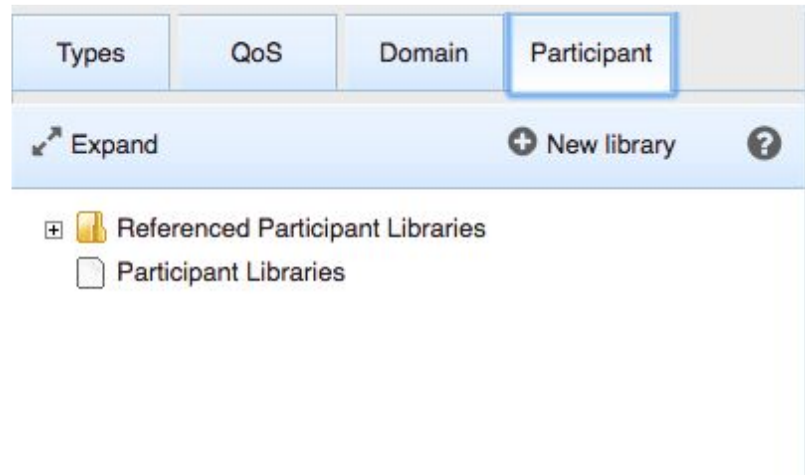
Of course, you can also see the XML definition:

```
<domain_library name="NewDomainLibrary1">
  <domain name="NewDomain1" domain_id="0">
    <register_type name="NewStruct1Registered" kind="dynamicData" type_ref="NewStruct1"/>
    <topic name="NewTopic1" register_type_ref="NewStruct1Registered"/>
  </domain>
</domain_library>
```

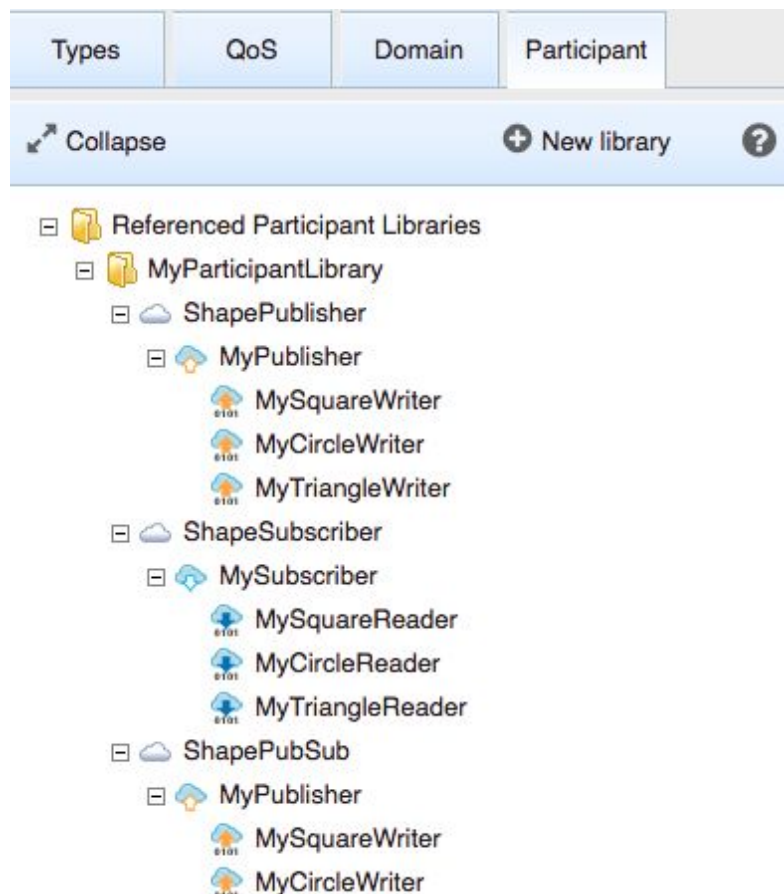
## Participant Library TAB

### Exploring Referenced Participant Library

Once you uploaded a file as shown above, you will notice, in the left panel, a new tree called “Referenced Participants Libs” under the “Participant Tab”:



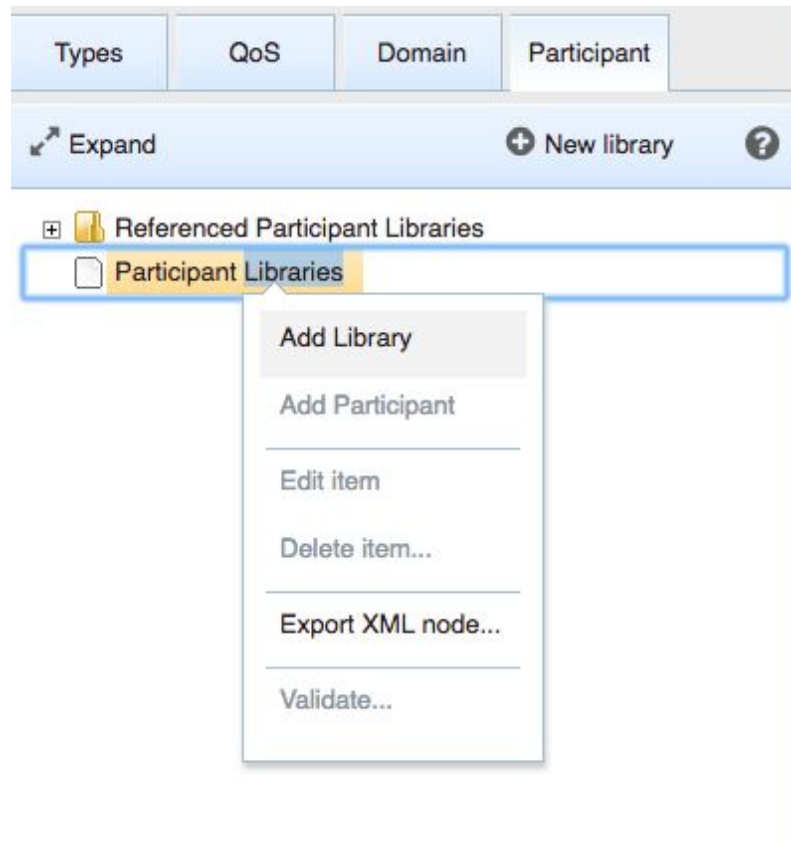
You can expand/collapse the whole tree using the “Expand” button on top of the page:



## Adding Local Participants

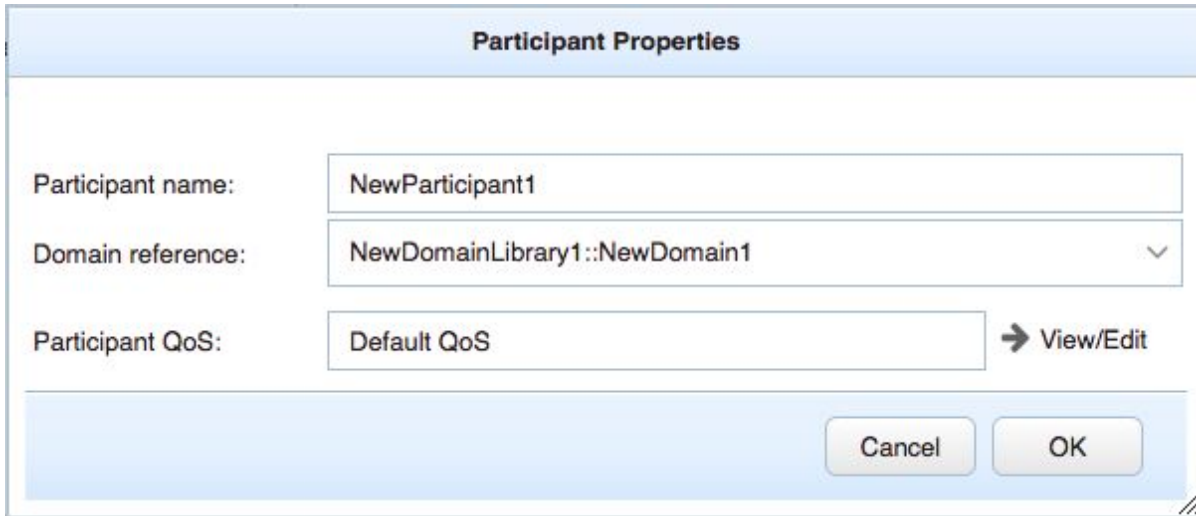
Let's see how you can create a new Participant Library containing readers and writers for another topic.

To start, right click on the "Local Participant Libs" tree:



A dialog will let you pick a name. In this example will call it *NewParticipantLibrary1*. Once the library has been added, right click on it and choose "Add Participant". A dialog box will be presented:





**Participant Properties**

Participant name:

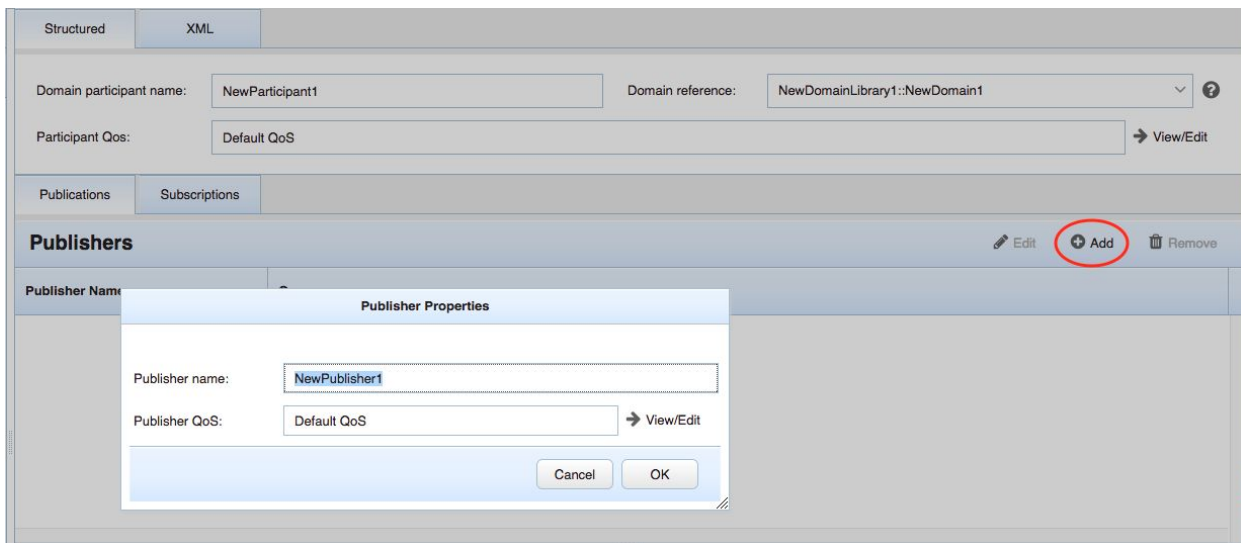
Domain reference:  ▼

Participant QoS:  → View/Edit

Cancel OK

In this dialog box you can pick the name (*NewParticipant1*) and you can select one of the available Domain Participant References. You can also select one of the available QoS profiles and customize them.

To add a new Publisher just select the “Add” button in the **Publishers** table:



Structured XML

Domain participant name:  Domain reference:  ▼ ?

Participant QoS:  → View/Edit

Publications Subscriptions

**Publishers** Edit Add Remove

Publisher Name

**Publisher Properties**

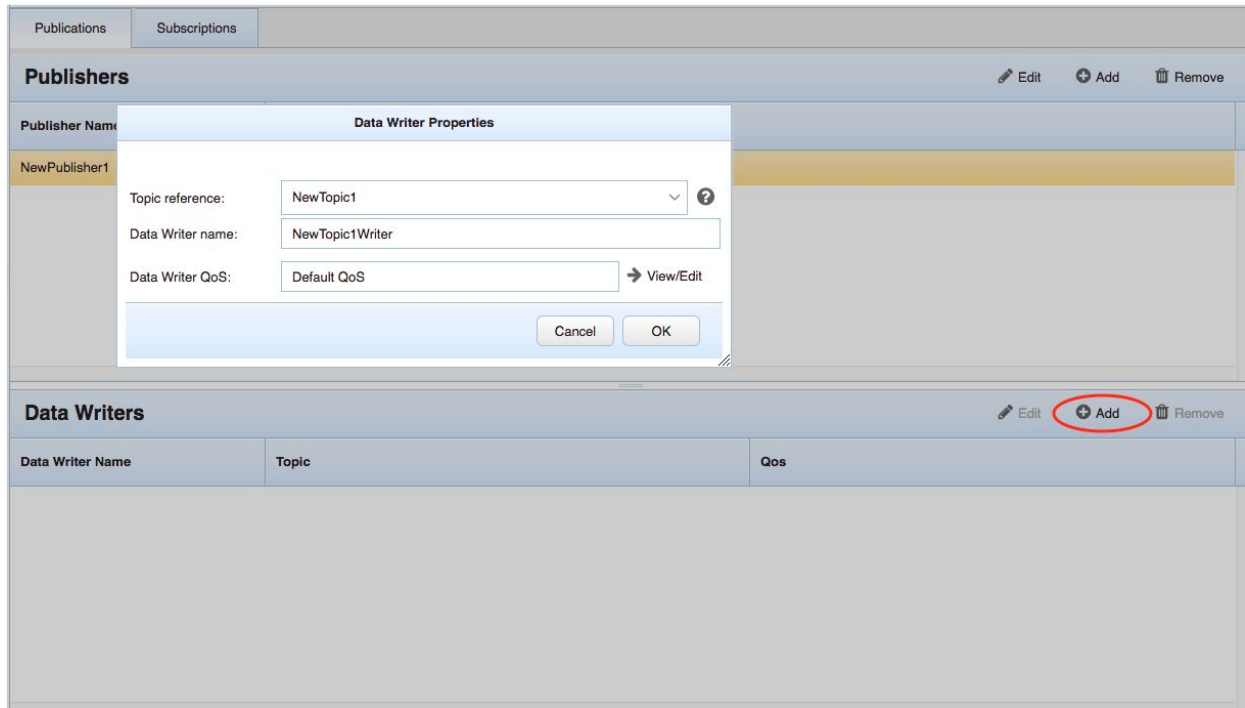
Publisher name:

Publisher QoS:  → View/Edit

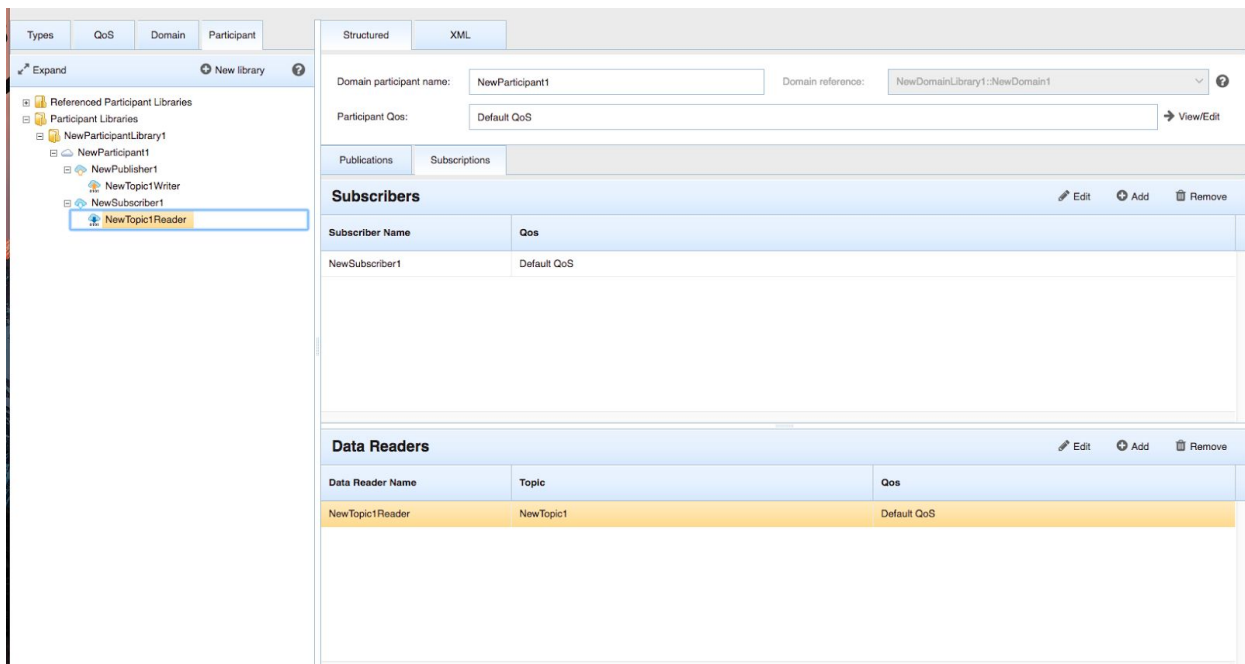
Cancel OK

A dialog box will be presented to allow you to pick a name and the QoS.

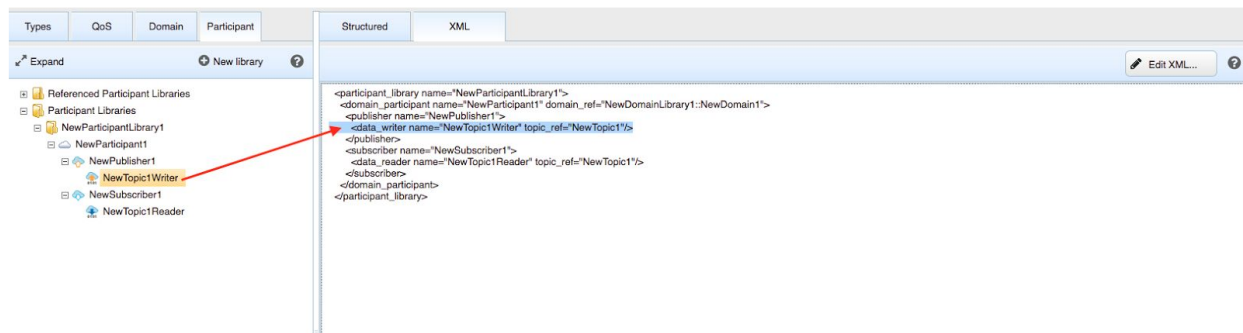
Once you add the publisher (*NewPublisher1*) you can select it and add a writer in the same way by clicking “Add” in the writer’s table. A dialog box will let you select a name and a topic:



You can do the same for adding a subscribers and a readers. This will be the result:



As you can see all the newly created entities are visible in the "Participant Libs" tree on the left panel. You can also see the result in XML:

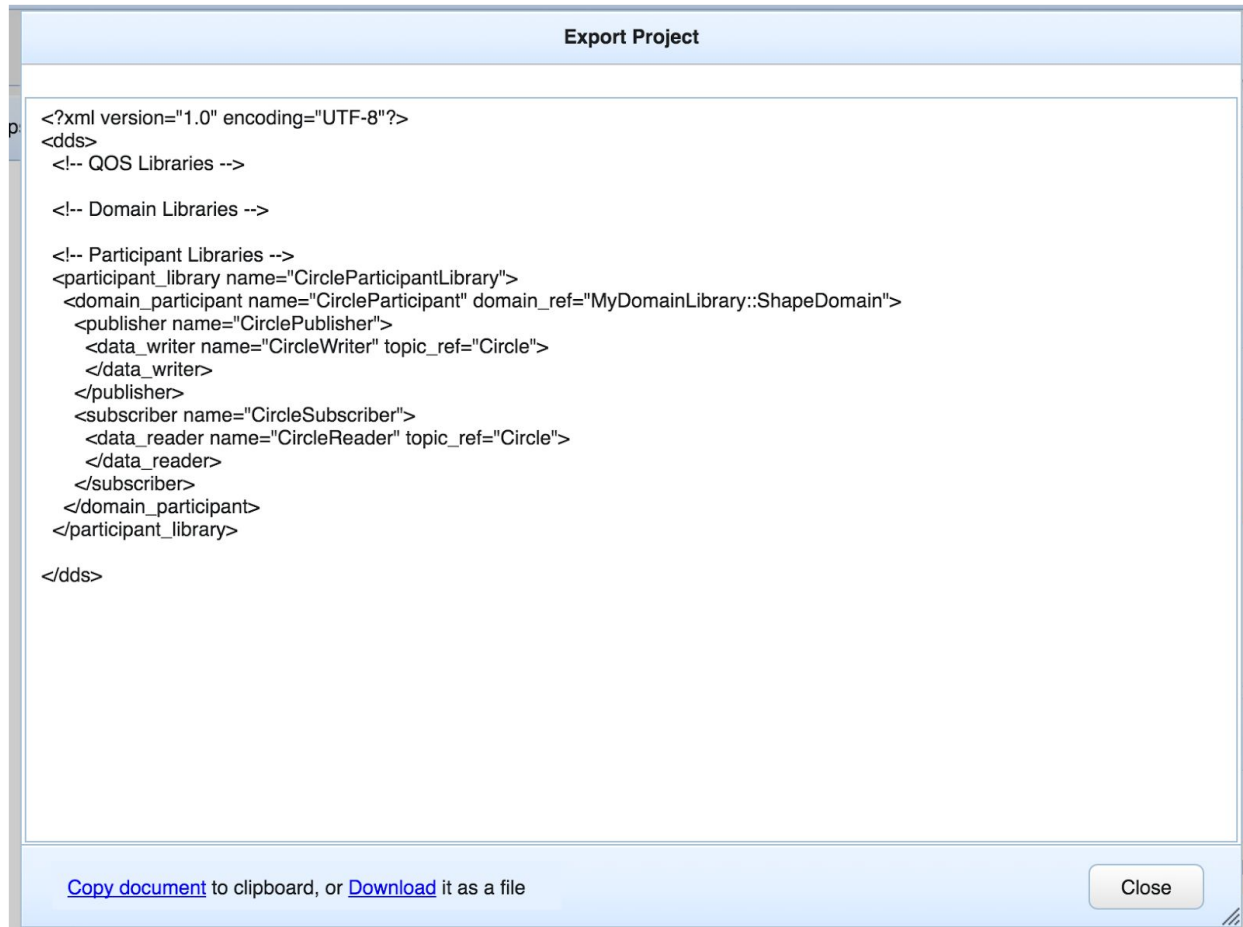


## Export an XML Application Creation File

If you want, you can generate an XML Application Creation File by using the “Export” button on the top right of the UI:



When you click on “export” a dialog box will be presented showing you the XML result:



You can copy or download the file using the links at the bottom of this dialog.

## Additional Notes about RTI System Designer

This section contains additional information that applies to the RTI System Designer (MCT CE)

### Using fully qualified names when referring to objects

When referring to another type, you can either use the target datatype without 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.**

I.e. 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`

Partial qualification like `'AnotherModule::BaseType'` is not accepted by the tool because it is ambiguous.

# Release Notes

## What's New/Fixed in this Version

## Known Issues and Limitations

### System Designer

- ❑ Export: copying functionality not supported in older browser

## Open Source Content

This package contains free and open source software provided license. For components provided in source code form, the associated license is provided with the software. This section lists other open source software used and the license and copyright associated with that software.

### **Lua:**

Copyright © 1994–2013 Lua.org, PUC-Rio.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

### **nodejs:**

Copyright Node.js contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to

deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

**body-parser:**  
(The MIT License)

Copyright (c) 2014 Jonathan Ong <me@jongleberry.com>  
Copyright (c) 2014-2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY

CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

**express:**  
(The MIT License)

Copyright (c) 2009-2014 TJ Holowaychuk <tj@vision-media.ca>  
Copyright (c) 2013-2014 Roman Shtylman <shtylman+expressjs@gmail.com>  
Copyright (c) 2014-2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.

IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY

CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

**express-fileupload:**  
The MIT License (MIT)

Copyright (c) 2015 Richard Girges

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is



furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

**express-session:**  
(The MIT License)

Copyright (c) 2010 Sencha Inc.

Copyright (c) 2011 TJ Holowaychuk <tj@vision-media.ca>

Copyright (c) 2014-2015 Douglas Christopher Wilson <doug@somethingdoug.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,

TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

**fs-extra:**

(The MIT License)

Copyright (c) 2011-2016 JP Richardson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

**nedb:**

(The MIT License)

Copyright (c) 2013 Louis Chatriot <louis.chatriot@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,

EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.

IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY

CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,

TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

**open:**

Copyright (c) 2012 Jay Jordan

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

**jquery:**

/\*! jQuery v1.11.2 | (c) 2005, 2014 jQuery Foundation, Inc. | [jquery.org/license](http://jquery.org/license) \*/

**jquery.xmleditor:**

<https://github.com/UNC-Libraries/jquery.xmleditor#license-information>

**sax:**

The ISC License

Copyright (c) Isaac Z. Schlueter and Contributors

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

**session-file-store:**

Copyright 2014-2015 Valery Barysok <valery.barysok@gmail.com>

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

**webix:**

<http://webix.com/legalnote/developer-license-agreement/>

**shelljs:**

Copyright (c) 2012, Artur Adib <arturadib@gmail.com>  
All rights reserved.

You may use this project under the terms of the New BSD license as follows:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright

notice, this list of conditions and the following disclaimer.

- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of Artur Adib nor the names of the contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ARTUR ADIB BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

#### **xml2js:**

Copyright 2010, 2011, 2012, 2013. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.



