RTI Monitor

for

RTI Data Distribution Service

Getting Started Guide

Version 4.5

The Global Leader in DDS

Printed in U.S.A. First printing.
October 2011.

Trademarks

Real-Time Innovations and RTI are 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

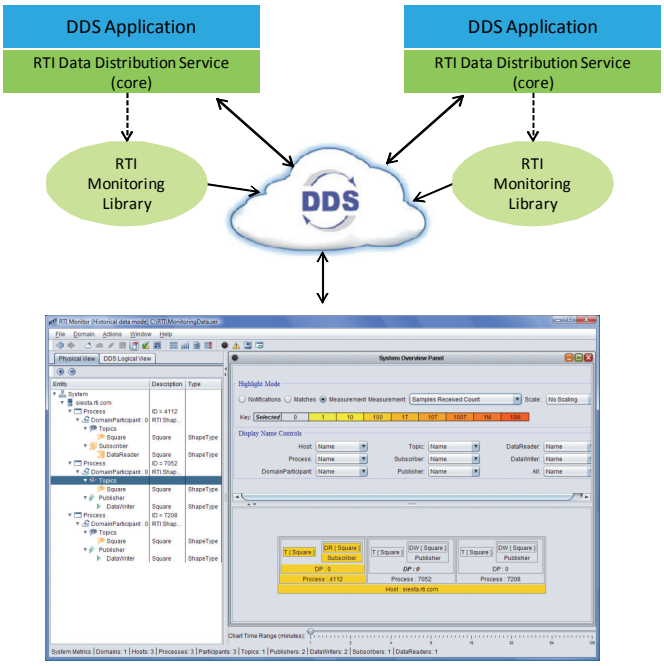
Technical Support

Real-Time Innovations, Inc.
385 Moffett Park Drive
Sunnyvale, CA 94089
Phone: (408) 990-7444
Email: support@rti.com
Website: https://support.rti.com/

Contents

Chapter 1 Welcome

RTI Monitor is a graphical tool that displays monitoring data from *RTI Data Distribution Service* applications.



RTI Monitor

RTI Monitor will help you:

- ❑ **Understand your system** with an easy-to-use graphical view into your entire DDS application.
- ❑ **Verify your design** by making sure your DDS entities are communicating as expected.
- ❑ **Tune performance** by providing deep statistics on every aspect of the middleware's operation.
- ❑ **Optimize integration** with detailed information on every DDS entity in your system.
- ❑ **Monitor real-time operation** with a dashboard of tools to see traffic patterns, errors, lost samples, and more.

You can run *RTI Monitor* on the same host as the DDS application or on a different host.

To enable an *RTI Data Distribution Service* application to provide monitoring data to *RTI Monitor*, the application needs to use the *RTI Monitoring Library* plug-in.

RTI Data Distribution Service notifies *RTI Monitoring Library* every time an entity is created/deleted or a QoS is changed. *RTI Monitoring Library* also periodically queries the status of all DDS entities. *RTI Monitoring Library* sends all the data to *RTI Monitor* once it gets the data from the DDS application.

Monitoring is enabled in the application by setting values in the DomainParticipant's PropertyQosPolicy (programmatically or through an XML QoS profile).

Refer to the *RTI Monitoring Library Getting Started Guide*, included with the *RTI Monitoring Library* bundle, for details. *RTI Monitoring Library* is available from the RTI Customer Portal (accessible from https://support.rti.com/).

Chapter 2 Installation and Start Up

2.1 Instructions for Linux Systems

2.1.1 Installing

1. Untar **RTI_Monitor-<*version*>-Linux.tar.gz** in a directory where you have write access.

   For example, enter:

   ```
   > cd /opt/rti
   > gunzip RTI_Monitor-<version>-Linux.tar.gz
   > gtar xvf RTI_Monitor-<version>-Linux.tar
   ```

   The resulting installation will be in **/opt/rti/RTI_Monitor_<version>**.

2. See Installing the License File (Section 2.3).

3. *RTI Monitor* is designed to monitor DDS applications that are using *RTI Monitoring Library*. If you have not yet installed *RTI Monitoring Library* for use with your DDS applications, you may want to do so now. Refer to the *RTI Monitoring Library Getting Started Guide*, shipped with the *RTI Monitoring Library* bundle, for details.

   RTI Monitoring Library is available from the RTI Customer Portal (accessible from https://support.rti.com/).

2.1.2 Running

Start *RTI Monitor* using the provided **rtimonitor** script**.**

For example, if you installed *RTI Monitor* in **/opt/rti**, start it by entering:

```
> /opt/rti/RTI_Monitor_<version>/scripts/rtimonitor
```

Command-line options are described in Section 2.4.

2.2 Instructions for Windows Systems

2.2.1 Installing

1. Right-click **RTI_Monitor-<*version*>-Win.zip** and select **Extract All...** to install *RTI Monitor* in the directory of your choice. You must have write access to the directory.

   If you choose **c:\RTI**, the resulting installation will be in **c:\RTI\RTI_Monitor_<*version*>**.

2. See Installing the License File (Section 2.3).

3. *RTI Monitor* is designed to monitor DDS applications that are using *RTI Monitoring Library*. If you have not yet installed *RTI Monitoring Library* for use with your DDS applications, you may want to do so now. Refer to the documentation in the *RTI Monitoring Library* bundle for details.

   RTI Monitoring Library is available from the RTI Customer Portal (accessible from https://support.rti.com/).

2.2.2 Running

Start *RTI Monitor* by double-clicking **<*installation directory*>\scripts\rtimonitor.bat**.

You can also start it from the command-line if you need to use any of the options described in Section 2.4.

2.3 Installing the License File

RTI Monitor requires a valid license to run. You will receive one via email after you download the software.

Save the license file in any location of your choice. When *RTI Monitor* starts, it will look in these locations until it finds a valid license:

1. The file **<*installation directory*>/RTI Monitor <*version*>/rti_license.dat**.

2. The file specified in the environment variable RTI_LICENSE_FILE, which you may set to point to the full path of the license file, including the filename (for example, **C:\RTI\my_rti_license.dat**).

3. The file **rti_license.dat** in the current working directory.
4. The file **rti_license.dat** in the directory specified by the environment variable NDDSHOME.

If *RTI Monitor* cannot find a valid license file automatically, it will prompt you to enter the location of a license file.

If you have any questions about license installation, please contact **support@rti.com**.

2.4 Command-line Options

RTI Monitor accepts the command-line options in Table 2.1.

Table 2.1 **Command-line Options**

Option	Description
-aggregationPeriodSeconds <seconds>	*RTI Monitor* periodically goes through all the monitored entities in the system (this information is saved in its own database) to calculate aggregated statistics and states. This value controls that minimum period (specified in seconds). Default: 5 seconds
-help	Displays all command-line options.
-historyDepth <value>	*RTI Monitor* saves some statistics' history, so it can be displayed in the charts. This option controls how much historical data (number of samples) is saved per monitoring topic. Default: 12 samples
-ignoreTypeConflicts	Instructs *RTI Monitor* to ignore any type conflicts. In *RTI Monitor*, type conflicts are based on type-code equality rather than type compatibility. This command-line option can be useful if you have types that have different type-code but are compatible. Default: Not specified (do not ignore type conflicts)

Table 2.1 **Command-line Options**

Option	Description
-initialDomainIds <domain_id_list>	Specifies which domains *RTI Monitor* will join when it starts up. <domain_id_list> is a list of domain IDs, each separated by a comma. **To specify multiple domain IDs on a Windows system**, enclose the comma-separated IDs in quotation marks. For example: **-initialDomainIds "31, 32"**. Default: If not specified, you will be prompted to enter a domain ID when *RTI Monitor* starts.
-matchRefreshPeriodSeconds <seconds>	Specifies the period at which to refresh the system overview panel's matches. Default: 5 seconds
-notificationHistoryDepth <value>	Specifies the number of notifications to keep per entity. Default: 12 notifications
-pruneDeadObjectsPeriodSeconds <seconds>	Sets the period at which *RTI Monitor* should clean up user-interface objects (such as the Host, and Process nodes in the tree views) that are no longer current (have no more children nodes in the tree view). This value should be increased when dealing with very large systems where the time to complete discovery is longer than the default value of 3 seconds. Default: 3 seconds
-spawnReadThreads	Instructs *RTI Monitor* to use multiple threads (according to the number of cores on the host) to retrieve data from its DDS DataReaders (which contain monitoring data). This is typically only needed for very large systems. Default: Not specified (use a single read thread to retrieve data at a period of 1 second)
-verbosity <value>	Sets the verbosity for *RTI Monitor* and DDS. 0: silent (both DDS and *RTI Monitor*) 1: errors (both DDS and *RTI Monitor*) 2: warnings (*RTI Monitor* only) 3: warnings (both DDS and *RTI Monitor*) 4: information (*RTI Monitor* only) 5: tracing (*RTI Monitor* only) 6: tracing (both DDS and *RTI Monitor*) Default: 1

Chapter 3 A Demo using RTI Shapes Demo

Before going through the steps in this chapter, make sure that:

- ❑ *RTI Monitor* is installed.
- ❑ *RTI Monitoring Library* is installed (can be on a different host than *RTI Monitor*).
- ❑ *RTI Shapes Demo* (dynamic version) is installed (architecture should match the one for *RTI Monitoring Library*)

In each of the following scenarios, make sure you use the special dynamic version of *RTI Shapes Demo*:

- ❑ Showing System Topology, Sample Counts and Rates (Section 3.2)
- ❑ Showing Content-Filtered Samples (Section 3.3)
- ❑ Showing Deadlines (Section 3.4)
- ❑ Showing a 'Samples Rejected' Scenario (Section 3.5)

When you are done with the above scenarios, please see Restoring RTI Shapes Demo to its Default (optional) (Section 3.6).

3.1 Setting Up Your Environment

In this demo, you will use the dynamic version of *RTI Shapes Demo* to dynamically load the monitoring library. Therefore, the monitoring library needs to be set in your path.

- ❑ **On Linux systems**, on the host where you will be running the dynamic version of *RTI Shapes Demo*, set your **LD_LIBRARY_PATH** environment variable to contain both the path to the monitoring library and to the DDS libraries (which are included in the *RTI Shapes Demo* bundle). For example:

```
setenv LD_LIBRARY_PATH <monitoring library install directory>/lib/
  <architecture>:<shapes demo install directory>/bin/
  <architecture>:${LD_LIBRARY_PATH}
```

where *<architecture>* might be i86Linux2.6gcc4.1.1, for example. The *<architecture>* for both the Monitoring Library and Shapes Demo must be the same.

- **On Windows systems**, on the host where you will be running the dynamic version of *RTI Shapes Demo*, set your Path environment variable to contain the path to the monitoring library.

  There are two ways to change your Path:

  1. You can change your Path in a command shell. This change is only effective for commands run from that particular command shell. Therefore you will have to use that same command shell to start each instance of *RTI Shapes Demo* required in the steps throughout this chapter.

     To change your Path in a command shell:

     ```
     set PATH=<monitoring library install directory>\lib\<arch>;%PATH%
     ```

     where *<arch>* might be i86Win32VS2005, for example, and must match the architecture for your installation of *RTI Shapes Demo*.

  2. Alternatively, you can change your Path at the system level through the Control Panel. (For example, select **Start, Control Panel, System, Advanced System Settings, Environment Variables**.[1]) If you use this method, the change is effective regardless of how you start *RTI Shapes Demo* (from a command shell or the Start menu.)

3.2 Showing System Topology, Sample Counts and Rates

1. Starts two instances of the dynamic version of *RTI Shapes Demo*:

   On a Windows system, from the Start menu, select **RTI**, **RTI Shapes Demo *<version>***, **Shapes Dynamic**.

   On a Linux system:

   ```
   > cd <RTI Shapes Demo installation directory>
   ```

1. This selection path varies slightly on different versions of Windows.

```
> bin/<architecture>/rtishapesdemo_dynamic
```

2. Configure both instances of *RTI Shapes Demo* to use the profile **MonitorDemoLibrary::MonitorDefault**, which is the default profile when you choose ***<Monitoring Library installation directory>*/resource/monitor/xml/MONITORING_DEMO.xml**:

   (Notice that we are using an XML file from *RTI Monitoring Library's* installation directory, not from *RTI Monitor's*.)

   a. Select **Controls, DDS Configuration**, **Stop DDS**, **Manage QoS**, **Add**.

   b. Add ***<Monitoring Library installation directory>*/resource/monitor/xml/MONITORING_DEMO.xml** to the list of profiles.

   c. Uncheck the box for **RTI_SHAPES_DEMO_QOS_PROFILES.xml** in the list of profiles if it is checked.

   d. Click **OK**.

   e. Choose the profile **MonitorDemoLibrary::MonitorDefault**.

   f. Select **Start DDS**.

3. In one instance of *RTI Shapes Demo*, create a reliable square publisher as follows:

   a. Select **Publish, Square**.

   b. Choose the profile **MonitorDemoLibrary::MonitorDefault**.

   c. Make sure the **Reliability** box is checked.

   d. Select **OK**.

4. In the other instance of *RTI Shapes Demo*, create a reliable square subscriber as follows:

   a. Select **Subscribe, Square**.

   b. Choose the profile **MonitorDemoLibrary::MonitorDefault**.

   c. Check the **Reliability** box.

   d. Select **OK**.

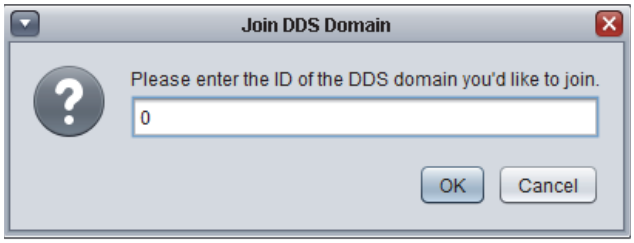
5. Start *RTI Monitor:*

   On a Windows system, double-click ***<installation directory>*\scripts\rtimonitor.bat** in your browser.

   Or on a Linux system, run the **rtimonitor** script:

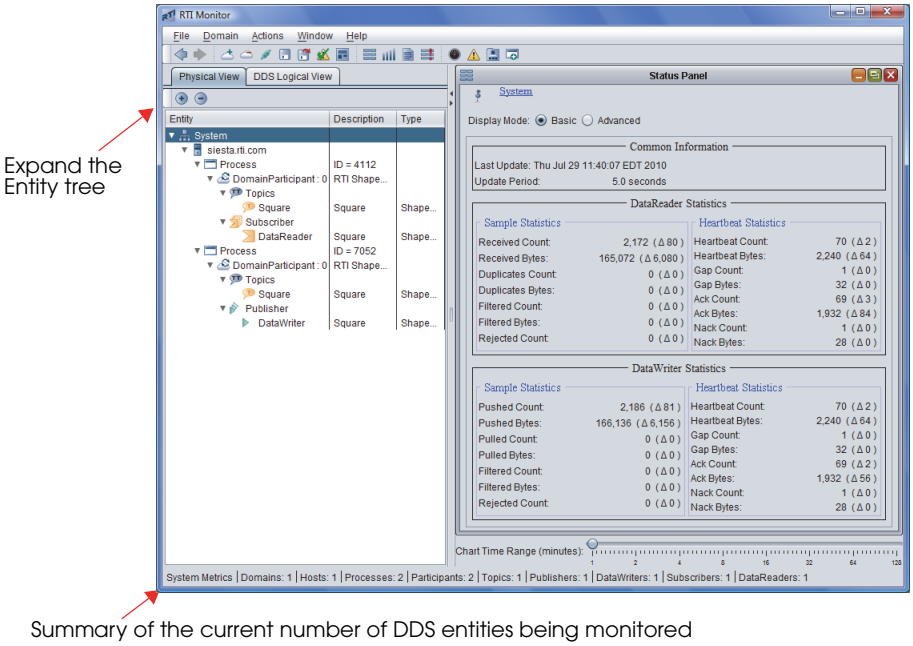
```
> <installation directory>/scripts/rtimonitor
```

6. When *RTI Monitor* starts, you will be prompted to enter a domain ID. If you used the default domain ID (0) when you started *RTI Shapes Demo*, enter 0 at the prompt. Otherwise, enter the same value you used when starting *RTI Shapes Demo*.



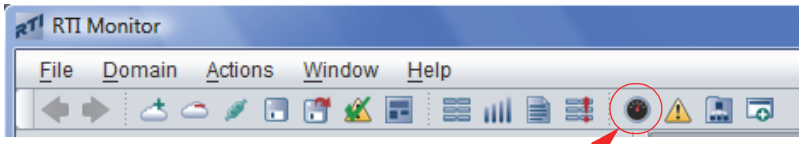
7. Review the system topology:

   a. Expand the Physical View tree by clicking the button below the **Physical View** tab.

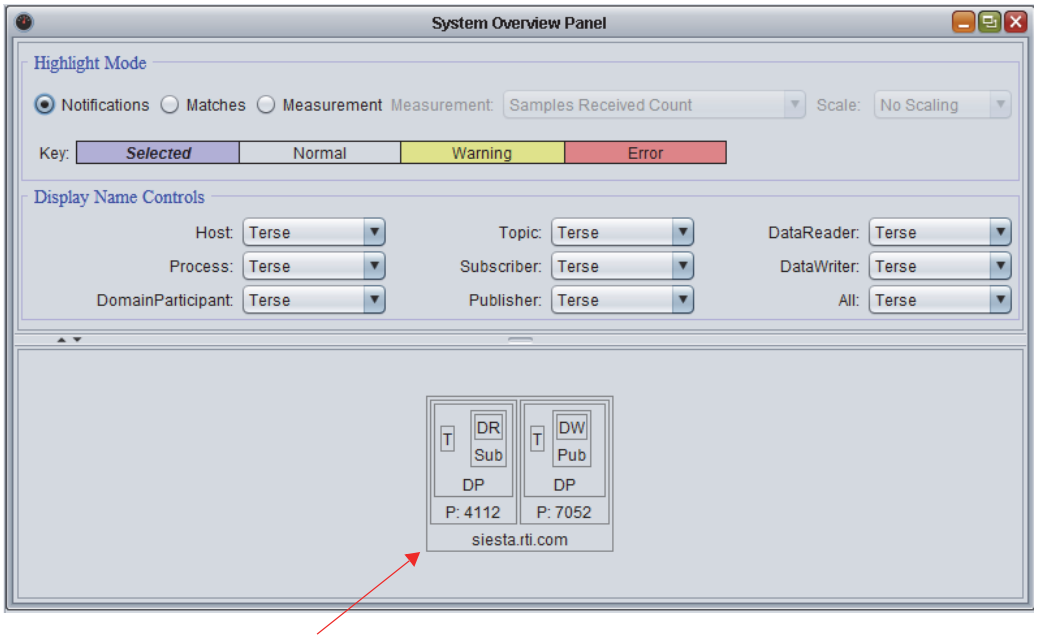


   b. Close the Status Panel on the right (select the red at the top-right corner of that panel).

c. Select the **System Overview** button in the toolbar to see a summary of the monitored DDS domain.



Create a new System Overview panel

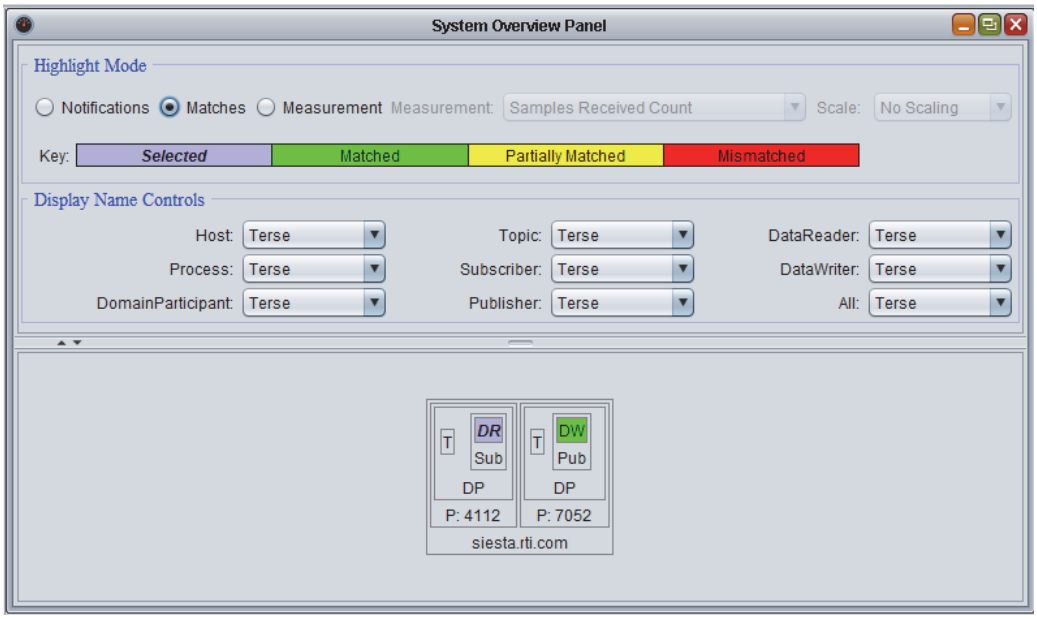


A visual map of the system.

The outer-most box represents the host.

T = Topic
DR = DataReader
DW = DataWriter
Sub = Subscriber
Pub = Publisher
DP = DomainParticipant
P:# = process ID

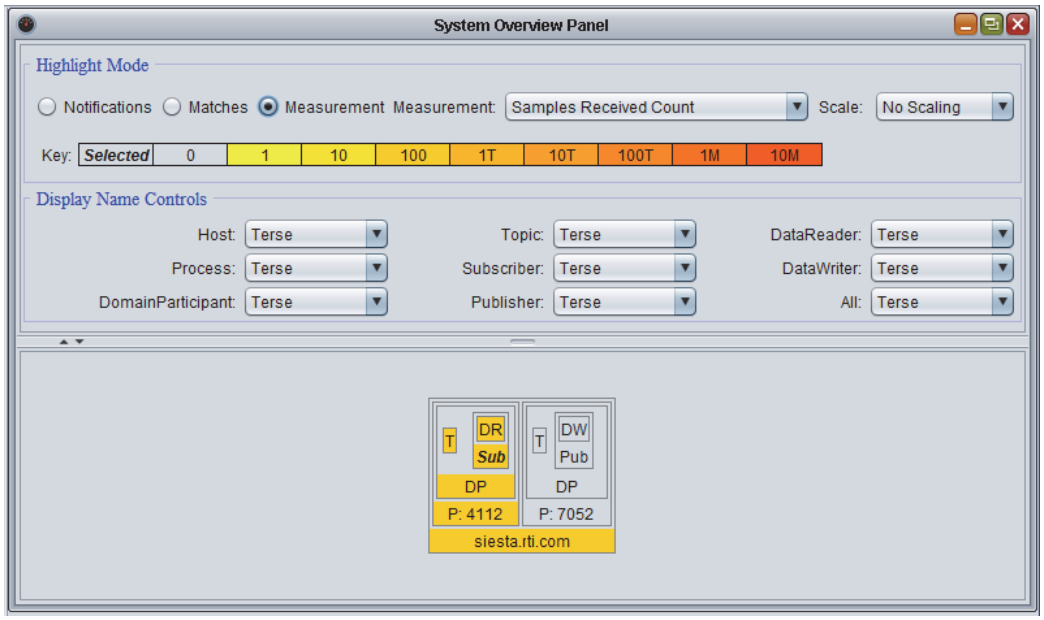
d. Select the **Matches** option in the System Overview panel. Select **DW** or **DR** in the system map to see what entities are matched in the system.



Notice that when you select an entity in the system map, that entity also becomes selected in the Physical View tree.

e. Click the Back button on the toolbar; it will change the selection back to the previously selected entity—in the Physical View tree and the System Overview panel. Try the Forward button too.

f. In the System Overview panel, select the **Measurement** option and **Samples Received Count** in the drop-down menu. You will see a color map that indicates the number of samples received by various entities in the system.

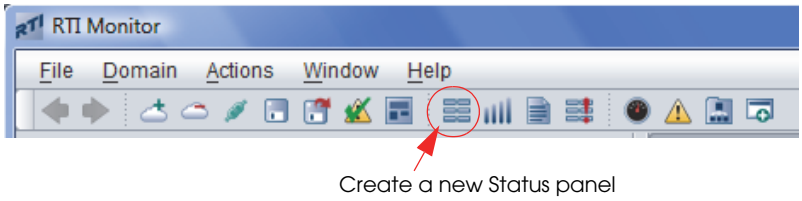


Notes:

- To change the scaling of the numbers indicated in the color map, use the **Scale** drop-down menu.
- Move the mouse over the entities in the map to see a tooltip of the actual value.
- You can select different **Display Name Controls** for each kind of entity (or all entities) to control how much detail will appear in the map in the System Overview panel. You can either **Hide** an entity kind, show a **Terse** form of the entity name, or show the full **Name.**

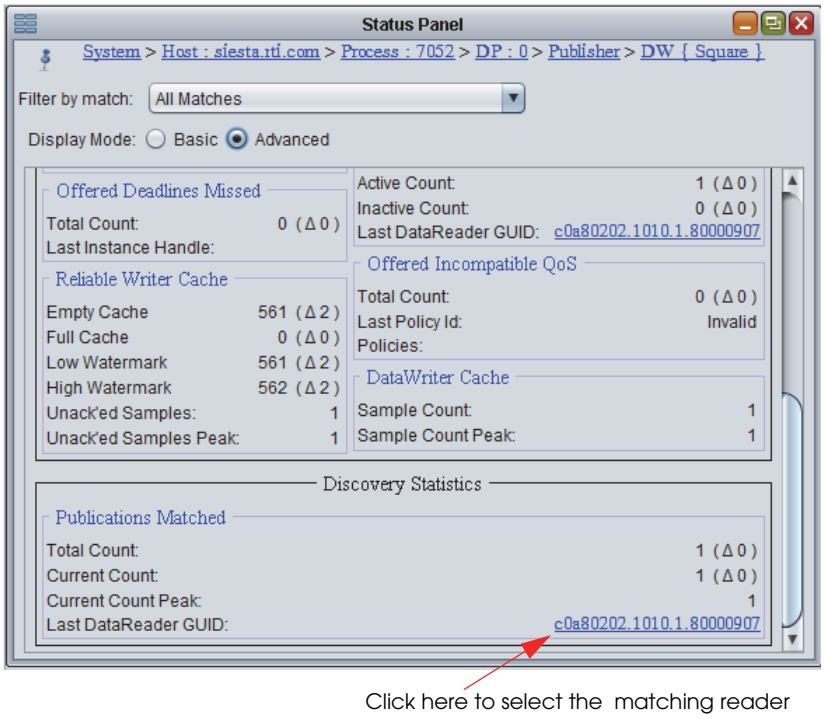
g. Close the System Overview panel.

h. In the Physical View tree, select the **DataWriter**, then select the **Status Panel** button in the toolbar.

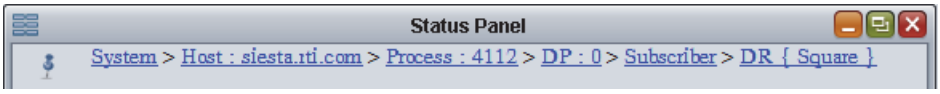


i. Select the **Advanced** display mode.

j. Scroll down to see Discovery Statistics. Click the link next to **Last DataReader GUID**. This will select the matching DataReader in the Physical View tree and the panel will switch to show DataReader status instead.



The top of the Status panel (and some of the other entity-specific panels that you will see later in this demo) shows a list of parent entities to which the selected entity belongs. For example:

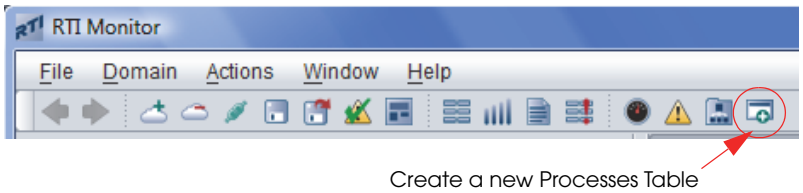


k. Click on **DP:0** in the list of parent entities. This will select the Domain Participant in the Physical View tree and the Status panel will change to show the participant's status. Notice that the Status panel now shows an aggregation of the statuses of all the data writers and data readers that belong to the selected participant.

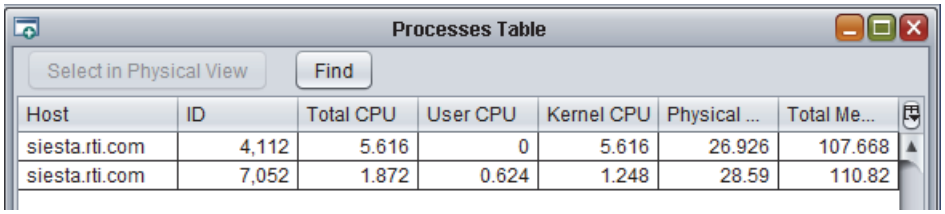
l. Close the Status panel.

8. Review all the processes in the system:

a. Select the **Processes Table** button from the toolbar.



b. This will display a panel that shows the processor and memory usage of all monitored processes. Select one of the processes and click on the **Select in Physical View** button; this will select the same process in the Physical View tree.

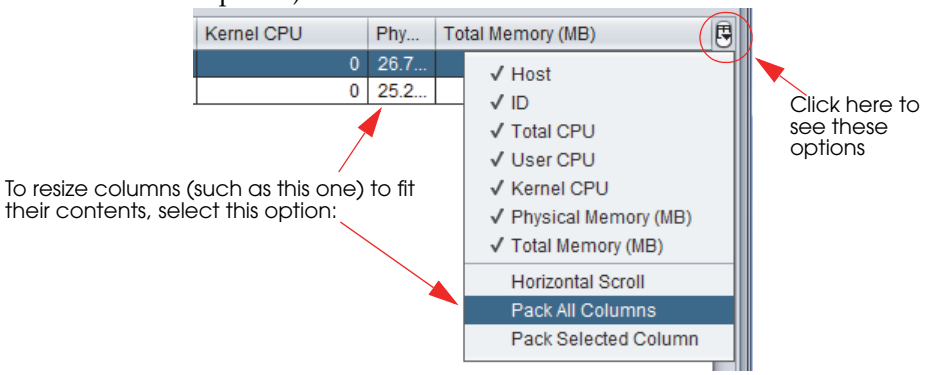


Host	ID	Total CPU	User CPU	Kernel CPU	Physical ...	Total Me...
siesta.rti.com	4,112	5.616	0	5.616	26.926	107.668
siesta.rti.com	7,052	1.872	0.624	1.248	28.59	110.82

The **Find** button is useful for searching a large table for a specific process. (This is a simple string search, so you must use the same format displayed in the table; for example, notice that the process ID includes a comma.)

c. Click on the **Total CPU** column heading. This will sort the table by the values in this column. Clicking it again will sort in the opposite order. This is useful to watch in real time to see which processes are using a lot of CPU. You can sort based on any of the columns.

d. Click the button just above the vertical scrollbar. This allows you to choose which columns appear in the table. For instance, to remove the **ID** column, uncheck it. (Note: to enable the 'Pack Selected Column' option, select a cell in the top row.)

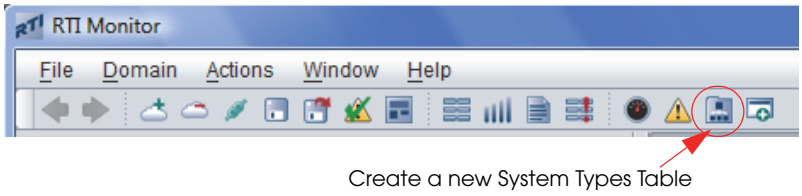


You can also change the order of the columns by simply dragging them to a new place in the table.

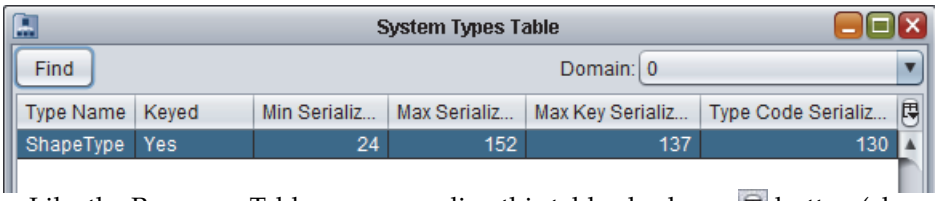
e. Close the Processes Table.

9. Review all the data types in the system:

a. Select the **System Types Table** button from the toolbar.



This will display a panel that shows all the known data types in the selected domain. In this case, there is only one data type called **ShapeType**.



Like the Processes Table we saw earlier, this table also has a button (above the vertical scrollbar) to control the columns that appear in the table. You can also sort the table based on any of the columns by clicking the column heading.

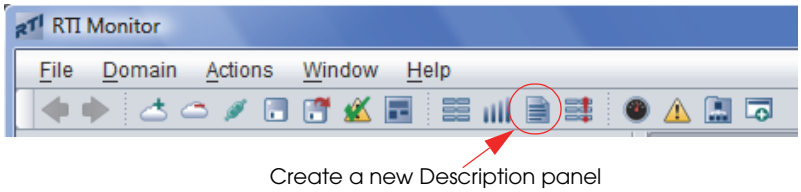
b. Select the **DDS Logical View** tab on the left. This is another display of the system tree, arranged by domain and topics. Select the **Square** topic from the tree and the corresponding row will be highlighted in the System Types Table.

c. Close the System Types Table.

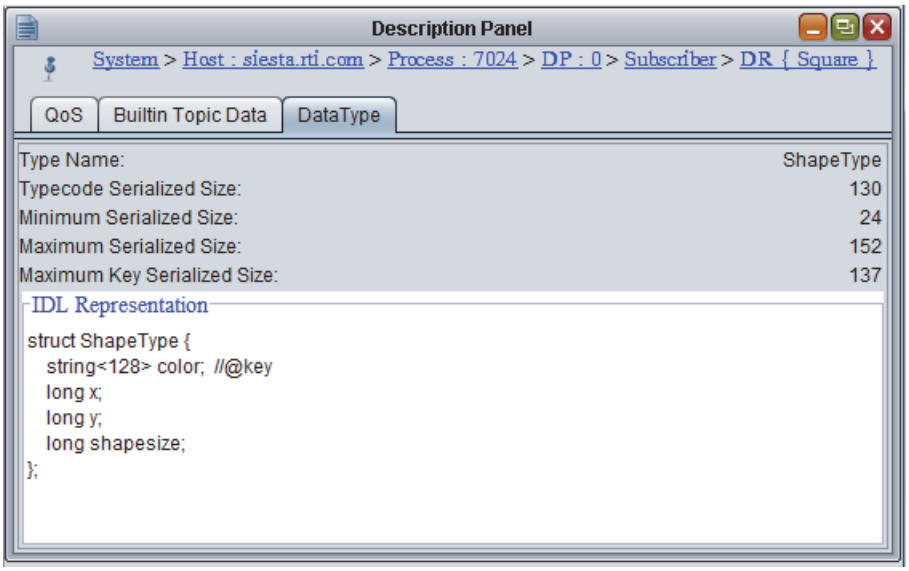
10. Show details of each data type

a. Select the **Physical View** tab on the left.

b. Select **DataReader** in the tree, then select the **Description Panel** button from the toolbar.



c. In the Description panel, select the **DataType** tab to see the data type for the data reader in IDL. You can also see other properties related to the data type.



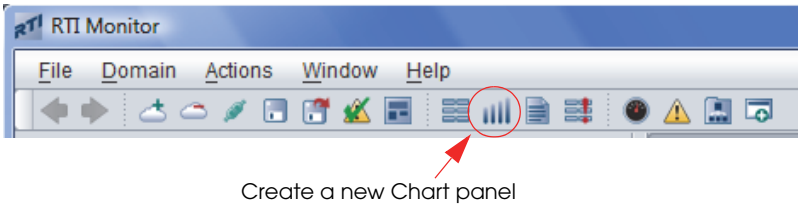
d. Close the Description Panel.

11. To show panels for two or more entities side-by-side for comparison, you can press the **pin** button at the top-left corner of all the entity-specific panels. The panel is then pinned to the entity and the panel will periodically receive updated data for the pinned entity—even when another entity is selected in the tree.

a. Select the **Physical View** tab on the left.

b. From the menu, make sure **Window**, **Auto Tile** is checked.

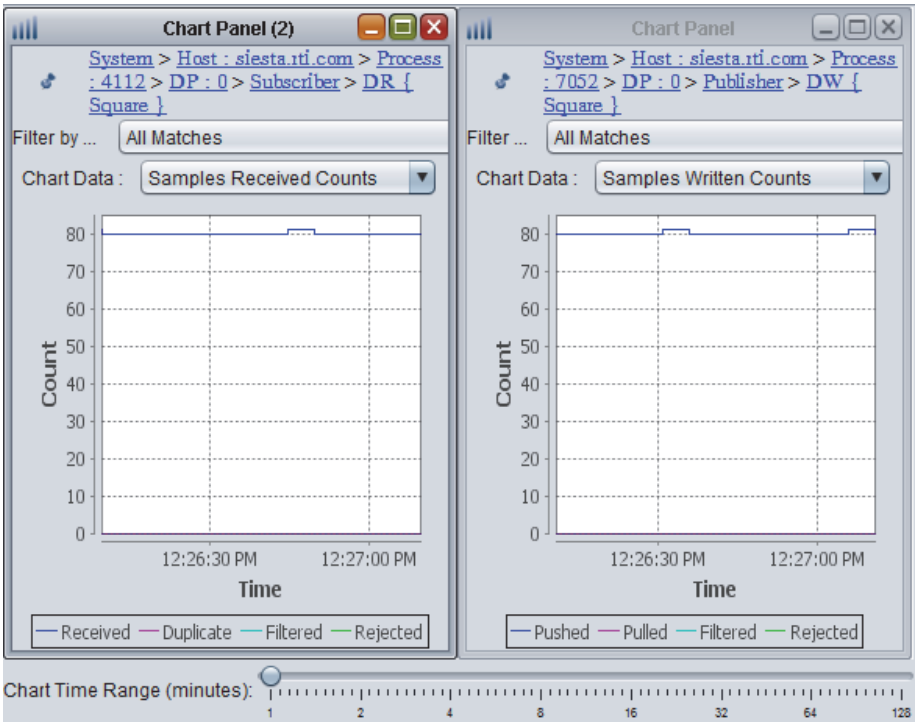
c. In the Physical View tree, select the **DataWriter**, then press the **Chart** button in the toolbar.



d. In the Chart panel, press the **pin** button on the top-left corner. Notice that the button has changed to **pinned** to indicate that the panel is pinned to a specific entity. For Chart Data, select **Samples Written Counts**.

e. In the Physical View tree, select the **DataReader**, then press the **Chart** button in the toolbar to create the second Chart panel and press its **pin** button. For Chart Data, select **Samples Received Counts**.

f. Now you can compare the DataWriter's Samples Written Counts and the DataReader's Samples Received Counts side-by-side. Notice that the send and receive sample counts are about the same.



Notes:

- The default monitoring library settings set the publish rate of the monitoring topics to 5 seconds. Therefore, you may need to wait 5 seconds for the *RTI Monitor* data to be updated.

- The Samples Written Counts and Samples Received Counts charts show the number of samples sent/received in the last sample period. In this case, the sample period is 5 seconds. Since *RTI Shapes Demo* publishes 16 samples per second, you will see approximately 80 (5 x 16) samples per sample period.
- The **Chart Time Range** slider (at the bottom of *RTI Monitor*) changes the time scale of the graphs.
- To unpin the panels, press their **pin** buttons again. Notice that now both chart panels are showing **DR** as the current entity at the top, since that entity is selected in the Physical View tree.

12. Start a third instance of the dynamic version of *RTI Shapes Demo* to create a reliable publication with a longer publication interval (250 ms).

For example, on a Windows system:

```
> cd <RTI Shapes Demo installation directory>
> bin\<arch>\rtishapesdemo_dynamic.exe -pubInterval 250
```

Or, on a Linux system:

```
> cd <RTI Shapes Demo installation directory>
> bin/<arch>/rtishapesdemo_dynamic -pubInterval 250
```

13. In the new *RTI Shapes Demo* window, create a reliable square publisher of a different color:

a. Select **Publish, Square**.

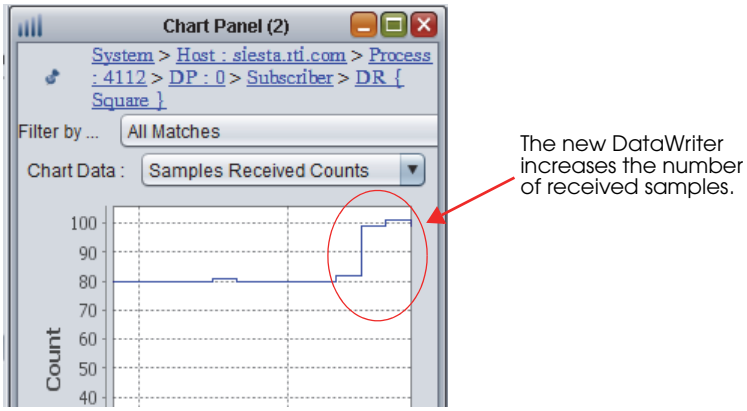
b. Select **YELLOW**.

c. Make sure the **Reliability** box is checked.

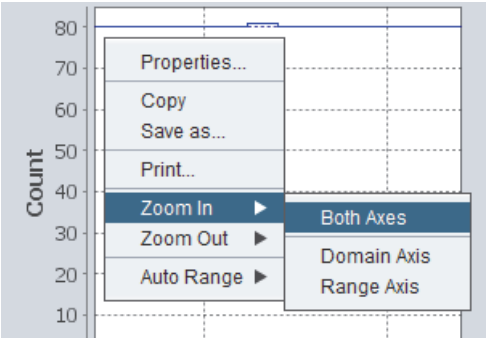
d. Select **OK**.

14. Examine the data in chart:

a. In *RTI Monitor*, notice the number of received samples increases in the chart..



b. Right-click in the white space of one of the charts to see how you can change the chart:



c. Close the two Chart panels.

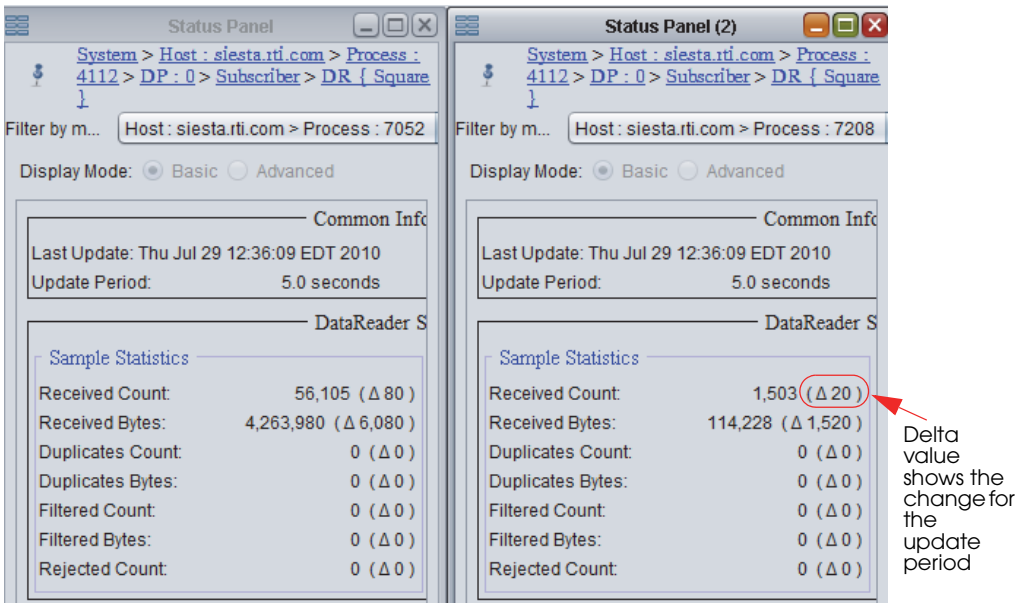
15. Let's see how to determine which writer is contributing more received samples:

a. In the Physical View tree, select the **Expand All** button.

b. Select the **DataReader** in the tree, then select the **Status Panel** button from the toolbar. For **Filter by match**, select the first matching endpoint in the drop-down menu.

c. Select the **Status Panel** button again to open a second status panel for the reader. For **Filter by match**, select the second matching endpoint in the drop-down menu.

Now you have status for both of the reader's matching writers side-by-side. The Received Count values will point out which one is contributing more samples.



16. Let's see how to save the current data to be used for further analysis later:

a. Select the **Save Data** button in the toolbar.

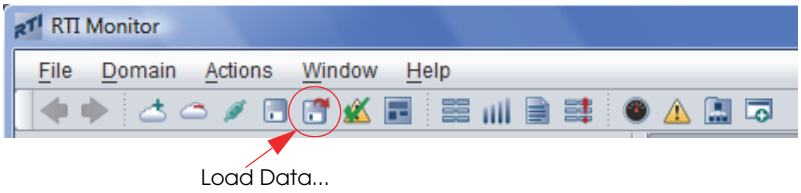
b. Select a location and enter a filename in the file dialog.

c. Close the third instance of *RTI Shapes Demo* (the one started with -pubInterval 250).

d. In the remaining two instances of *RTI Shapes Demo*, select **Controls, Delete All.**

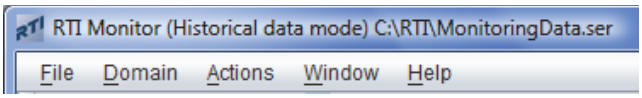
No shapes publications or subscriptions should be running in the system at this point.

e. Select the **Load Data** button from the toolbar.



f. You will see a prompt asking if you want to lose the current data and leave the domain. Select **OK** to continue. In the file dialog, select the file in which you previously saved the data.

Notice that the title of the *RTI Monitor* window has changed to **Historical data mode** and show the name of the loaded data file. Now you are seeing a snapshot of the system.



g. Select the **Expand All** button for the Physical View tree. You can see all the previously created entities, even though no publications or subscriptions are currently running.

h. Select **Domain, Show Current Domain** from the menu. Notice that you are not joined to any domain now because *RTI Monitor* is showing historical data instead of live data. Click **OK** to close the dialog box.

17. Prepare for the next demo:

a. Select the **Join Domain** button from the toolbar. You will see a prompt asking if you want to lose the currently loaded data. Click **OK** to continue. Rejoin your original domain by entering the domain ID, then click **OK**.

Notice that the title of *RTI Monitor* is no longer showing **Historical data mode**. *RTI Monitor* is showing live data again.

b. Close all the panels.

3.3 Showing Content-Filtered Samples

The steps in this section assume you are using the same profile used in Section 3.2 for the two instances of *RTI Shapes Demo* (dynamic version).

1. In one instance of the dynamic version of *RTI Shapes Demo*, create a reliable square publisher:
   - **a.** Select **Publish**, **Square**.
   - **b.** Make sure the **Reliability** box is checked.
   - **c.** Select **OK**.
2. In the other instance of the dynamic version of *RTI Shapes Demo*, create a content-filtered, reliable square subscriber:
   - **a.** Select **Subscribe**, **Square**.
   - **b.** Check the **Reliability** box.
   - **c.** Check the **Use Filter** box under Content Filter Topic.
   - **d.** Select **OK**.

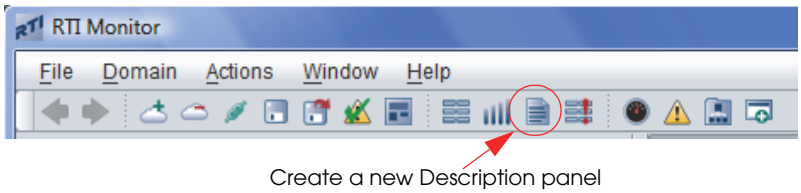
     Notice that the subscriber only receives samples that are within the filtering square.
3. Observe the filtered samples in *RTI Monitor*:
   - **a.** Select the **Expand All** button for the Physical View tree in *RTI Monitor*.
   - **b.** In the Physical View tree, select the **DataWriter**, then select the **Status Panel** button from the toolbar.

     In the Status panel, notice that the **Sample Statistics**, **Filtered Count** is non-zero and keeps changing.
   - **c.** In the Physical View tree, select the **DataReader**.

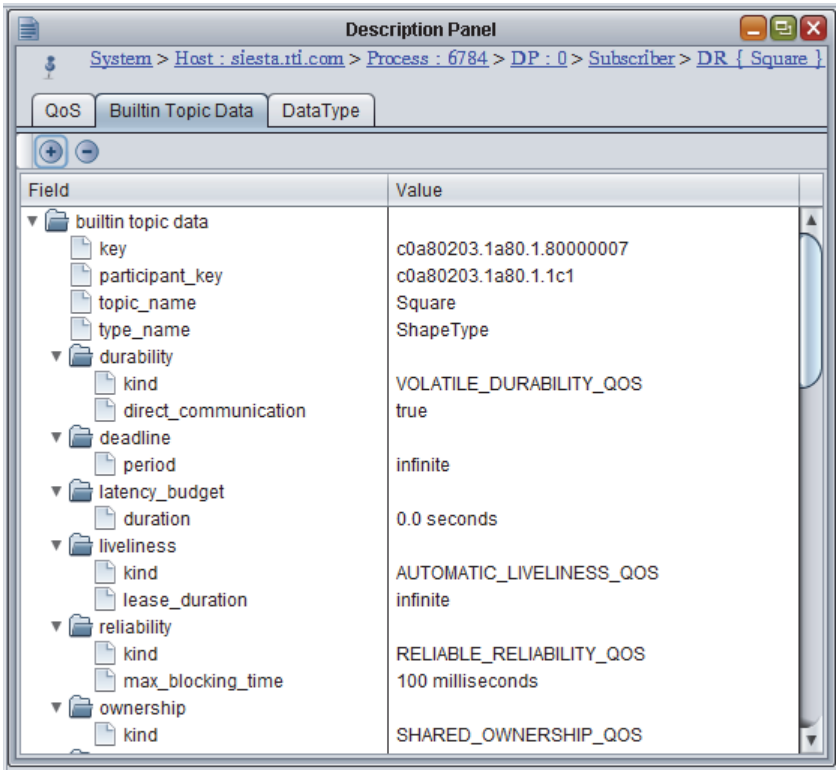
     Now the Status panel is showing values for the **DataReader**. Notice that the **Sample Statistics**, **Filtered Count** is zero. This shows that content filtering is only happening on the writer side in this case.

4. Show the content filter expression

   a. In the Physical View tree, select the **DataReader**, then select the **Description Panel** button from the toolbar.



   b. Select the **Builtin Topic Data** tab in the Description Panel.



   c. Notice the content filter expression under **builtin topic data/content_filter_property/filter_expression** and **builtin topic data/content_filter_property/expression_parameters**.

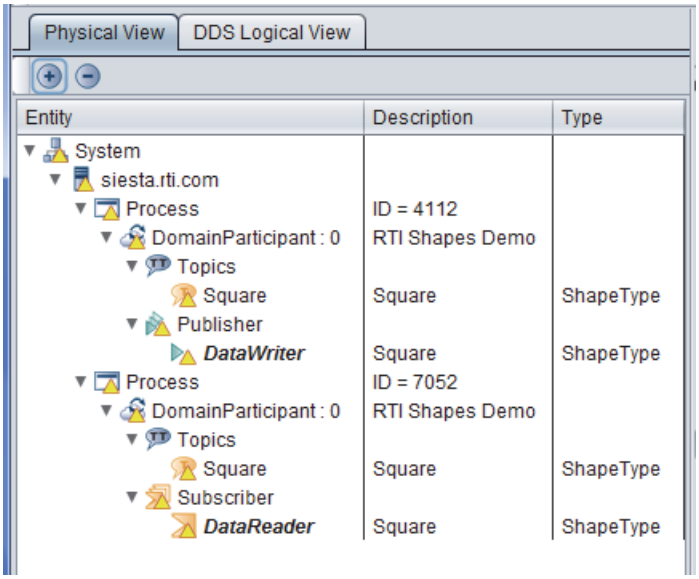
5. Prepare for the next demo:
   a. Close the Status panel.
   b. In the two instances of *RTI Shapes Demo*, select **Controls, Delete All.**

3.4 Showing Deadlines

The steps in this section assume you are using the same profile used in Section 3.2 for the two instances of *RTI Shapes Demo* (dynamic version).

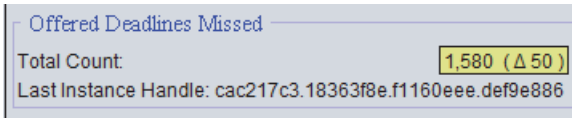
1. In one instance of *RTI Shapes Demo*, create a reliable square publisher with a 100ms deadline:
   a. Select **Publish, Square**.
   b. Make sure the **Reliability** box is checked.
   c. Set **Deadline** to **100**.
   d. Select **OK**.
2. In the other instance of *RTI Shapes Demo*, create a reliable square subscriber with a 250ms deadline:
   a. Select **Subscribe, Square.**
   b. Check the **Reliability** box.
   c. Set **Deadline** to **250**.
   d. Select **OK**.
3. In the publisher shapes demo instance, select **Controls, Pause Publishing**.
4. Observe the results in *RTI Monitor*:
   a. Select the **Expand All** button for the Physical View tree in *RTI Monitor*.

Notice that all the entities in the Physical View tree are marked with yellow triangles to show there is a potential problem. The root cause of the problem is in bold (the **DataWriter** and **DataReader** in this case). The parent entities are also marked with yellow triangles, but not in bold.



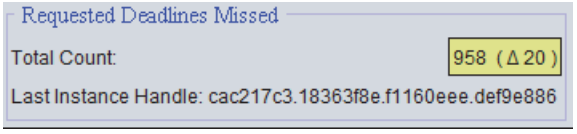
b. In the Physical View tree, select the **DataWriter**, then select the **Status Panel** button in the toolbar.

c. In the Status panel, select the **Advanced** display mode. Under **Offered Deadlines Missed**, notice the non-zero **Total Count** highlighted in yellow.



d. In the Physical View tree, select the **DataReader**. Now the Status Panel is showing values for the **DataReader**.

Under **Requested Deadlines Missed**, you will see a non-zero **Total Count** highlighted in yellow.

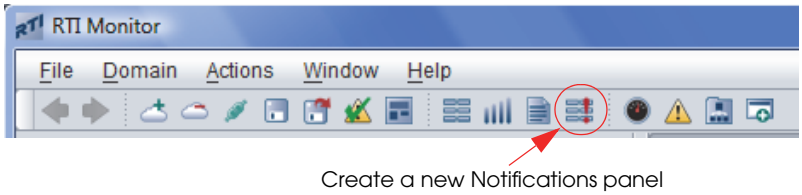


You may notice that the **DataWriter's Offered Deadlines Missed Total Count** is different than the **DataReader's Requested Deadlines Missed Total Count**. That's because these entities were created with different deadline values (100ms for the writer, 250ms for the reader).

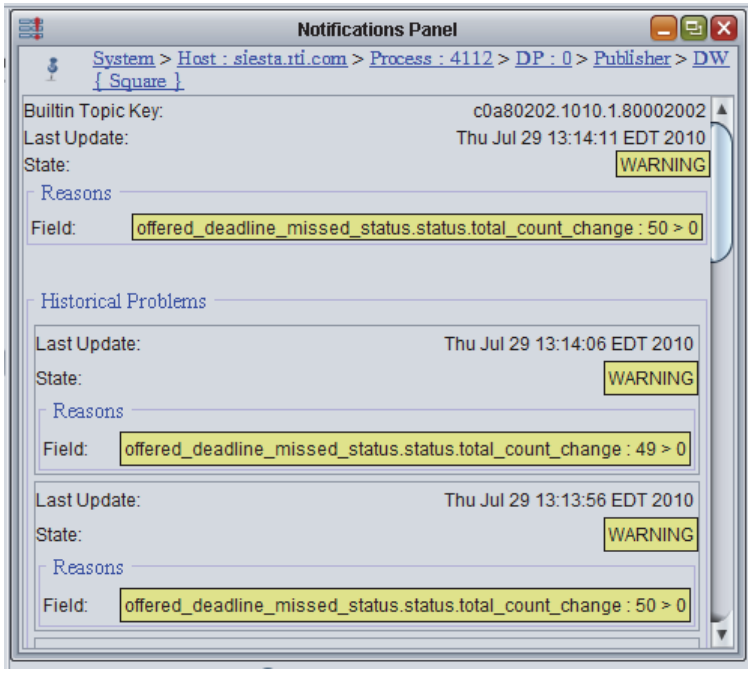
e. Close the Status panel.

5. Look at the notifications:

a. In the Physical View tree, select the **DataWriter**, then select the **Notifications Panel** button in the toolbar.

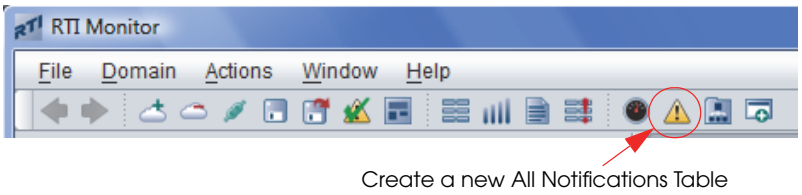


The Notifications panel displays the selected entity's current status and a historical list of all alarm statuses related to the selected entity.

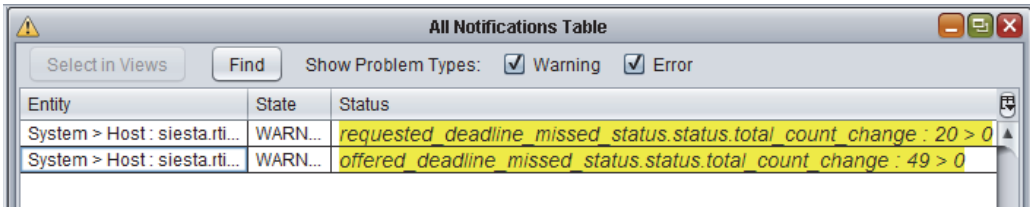


b. Close the Notifications panel.

c. Select the **All Notifications Table** button from the toolbar.



The All Notifications Table displays all the notifications *in the entire system* (not just for the selected entity). By default, it shows both warnings and errors. You can choose to see either just the warnings or just the errors by checking/unchecking the options.



If a row is selected in the All Notifications Table, clicking the **Select In Views** button in the All Notifications Table will select the corresponding entity in the tree views.

d. Close the All Notifications Table panel.

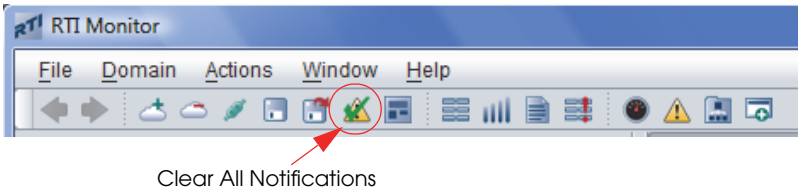
e. Select the **System Overview Panel** button from the toolbar. In the System Overview Panel, with the **Notifications** option selected, all the entities in the system that have caused a notification are highlighted in the system map.

f. Close the System Overview Panel.

6. Clear the notifications:

a. In the publisher shapes demo instance, select **Controls, Resume Publishing**.

b. From *RTI Monitor's* menu, select the **Clear All Notifications** button from the toolbar. This will clear all the yellow markers in the tree.



7. Prepare for the next demo:
   a. In one instance of *RTI Shapes Demo*, select **Controls, Delete All**.
   b. Close the other *RTI Shapes Demo* instance.

3.5 Showing a 'Samples Rejected' Scenario

1. Configure the existing instance of *RTI Shapes Demo* to use the profile, **MonitorDemoLibrary::SamplesRejectedScenario**[1].
   a. Select **Controls, DDS Configuration**, **Stop DDS**.
   b. Choose the profile **MonitorDemoLibrary::SamplesRejectedScenario**.
   c. Select **Start DDS**.
2. Create a reliable square publisher with the **MonitorDemoLibrary::SamplesRejectedScenario** profile:
   a. Select **Publish, Square**.
   b. Choose the profile **MonitorDemoLibrary::SamplesRejectedScenario.**
   c. Make sure the **Reliability** box is checked.
   d. Select **OK**.
3. Create a new instance of *RTI Shapes Demo* with a reliable subscribing rate of 1,000 ms:

   For example, on a Windows system:

   ```
   > cd <RTI Shapes Demo installation directory>
   > bin\<arch>\rtishapesdemo_dynamic.exe -subInterval 1000
   ```

   Or, on a Linux system:

   ```
   > cd <RTI Shapes Demo installation directory>
   > bin/<arch>/rtishapesdemo_dynamic -subInterval 1000
   ```
4. In the new *RTI Shapes Demo* instance, create a reliable square subscriber that uses **take()** and the **MonitorDemoLibrary::SamplesRejectedScenario** profile:
   a. Select **Subscribe, Square**.
   b. Choose the profile **MonitorDemoLibrary::SamplesRejectedScenario.**

1. The **SamplesRejectedScenario** profile is in the same file that you already selected in Section 3.2.

c. Check the **Reliability** box.

d. For the Read method to use, select **Take()**.

e. Select **OK**.

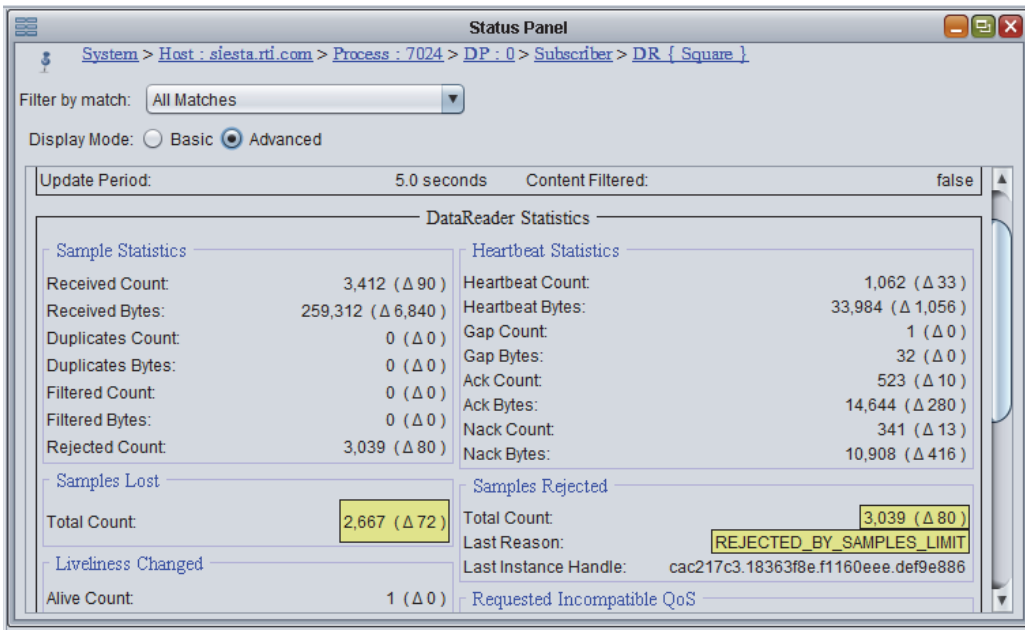
5. Notice that the subscriber is not receiving all the samples. Let's see why.

   a. Select the **Expand All** button for the Physical View tree in *RTI Monitor*.

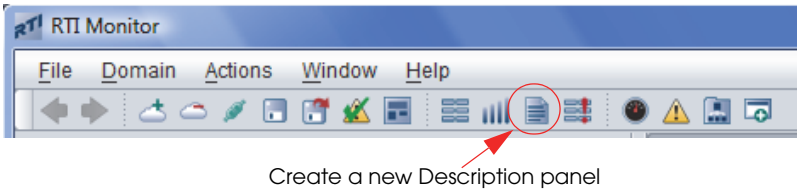
   b. In the Physical View tree, select the **DataReader**, then select the **Status Panel** button from the toolbar.

   c. Select the **Advanced** display mode.

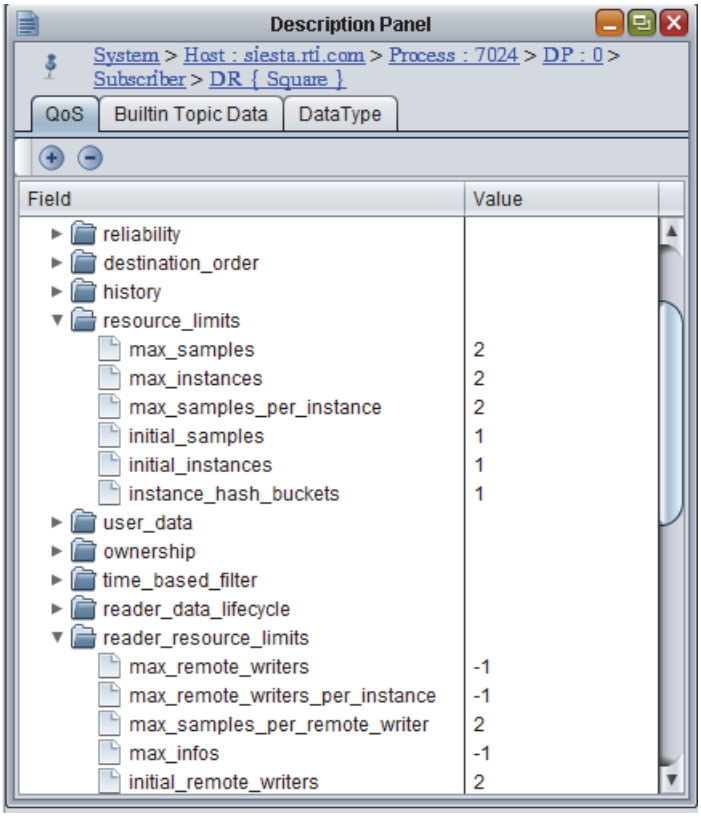
   Notice that the **Total Count** values for **Samples Lost** and **Samples Rejected** are non-zero. This indicates that not all samples are being received. Samples Lost and Samples Rejected are both yellow to indicate that this may or may not be a problem, depending on your use case. Also notice further down under **DataReader Cache** that both **Sample Count** and **Sample Count Peak** are 2 (you will see why this is significant later).



d. In the Physical View tree, select the **DataReader**, then select the **Description Panel** button from the toolbar.



In the QoS tab, scroll down and notice that the values for **qos/resource_limits/max_samples** and **qos/reader_resource_limits/max_samples_per_remote_writer** are both 2, which is the same as the **Sample Count** and **Sample Count Peak** values we saw in the Status panel previously. This shows that the reader's queue for receiving samples is full.



6. Fix the problem by creating a reader with a larger queue size:
   a. In the subscriber shapes demo instance, select **Controls, Delete All**.
   b. Create a new reliable square subscriber that uses **take()** and the **MonitorDemoLibrary::FixedSamplesRejectedScenario** profile, which will fix the problem by increasing the queue size.
      c. Select **Subscribe, Square**.
      d. Choose the profile **MonitorDemoLibrary::FixedSamplesRejectedScenario**.
      e. Check the **Reliability** box.
      f. For Read method to use, select **Take()**.
      g. Select **OK**.

      This profile uses a larger queue size. Notice that the subscriber is now receiving all the samples.
7. Verify the new reader queue size in *RTI Monitor*:
   a. Select the **Expand All** button under the Physical View tab.
   b. In the Physical View tree, select the **DataReader**.
   c. In the Status panel that is already open, notice that the **Total Count** values for **Samples Lost** and **Samples Rejected** are now zero.

      In the **DataReader Cache** section, notice the values for **Sample Count** and **Sample Count Peak**.

      In the Description panel that is already open, notice in the QoS tab that the values for **qos/resource_limits/max_samples** and **qos/reader_resource_limits/max_samples_per_remote_writer** are now 100. This is larger than the **SampleCount** and **Sample Count Peak** values seen in the Status panel. This shows that now the reader queue still has a lot of room before it becomes full and starts dropping samples.

   This concludes the demo. Please see Section 3.6.

3.6 Restoring RTI Shapes Demo to its Default (optional)

Once you've gone through the scenarios in the previous sections, your installation of *RTI Shapes Demo* is configured to turn on monitoring by default (the settings are saved in the demo's workspace file in your system). Now if you run the regular version of *RTI Shapes Demo,* which is statically linked, it may cause a segmentation fault. This is because the default saved profile in *RTI Shapes Demo's* workspace is trying to load the dynamic version of the monitoring library, which isn't compatible with the static/regular version of *RTI Shapes Demo.*

To restore RTI Shapes Demo to its original state:

1. Start an instance of the dynamic version of *RTI Shapes Demo* (or use the previous instance of the dynamic version from the previous demo steps if it is still open).
2. Select **Controls, DDS Configuration**, **Stop DDS, Manage QoS.**
3. Uncheck the box for **MONITORING_DEMO.xml.**
4. Check the box for **RTI_SHAPES_DEMO_QOS_PROFILES.xml** (the default profile).
5. Click **OK**.
6. Select **Start DDS**.

   Now *RTI Shapes Demo* (the dynamic and regular versions) will no longer have monitoring turned on by default.

Chapter 4 Troubleshooting

4.1 Debugging Problems with RTI Monitor on Windows System

If you run *RTI Monitor* on a Windows system and it is not showing any error messages, but the *RTI Monitor* window is not showing up, or you are not seeing any data in the *RTI Monitor* window, you can modify the file ***<installation directory>*\scripts\rtimonitor.bat** to change **javaw.exe** to **java.exe**, so that error messages will be displayed in a command prompt to help you debug the issue.

4.2 Running RTI Monitor on a System with Limited Memory

RTI Monitor runs with Java and a default maximum Java heap size of 500m. If you are monitoring on a system with very little memory and you are only monitoring a very small DDS system, you may be able to reduce memory usage by modifying the file in ***<installation directory>*\scripts\rtimonitor.bat** (on Windows systems) or ***<installation directory>*/scripts/rtimonitor** (on Linux systems) to decrease the maximum Java heap size usage.

For example, in the script change "**-Xmx500m**" to "**-Xmx300m**".

RTI Monitor will save some history of statistics to be displayed in the charts. By default, this value is 12. If you are running on a system with limited memory, you can decrease this value with the command-line option, -historyDepth <value> (see page 2-3).

4.3 Running RTI Monitor with a Large DDS System

RTI Monitor runs with Java and a default maximum Java heap size of 500m. If you are monitoring a very large DDS system, you may need to modify the file ***<installation directory>*\scripts\rtimonitor.bat** (on Windows systems) or ***<installation directory>*/scripts/rtimonitor** (on Linux systems) to increase the maximum Java heap size usage. For example, in the script change "**-Xmx500m**" to "**-Xmx1536m**".

4.4 Error Regarding 'Incompatible Shared Memory Segment'

If you see the following error messages:

```
[D0000|ENABLE]NDDS_Transport_Shmem_attach_writer:incompatible shared
memory segment found.
Found segment with max message size 9216. Needed 65530.
```

These messages likely mean either:

a. Another application is currently running on the same host, in the same domain, with different shared-memory transport settings, or

b. If you are on a Linux system, there was an old application running on that domain ID before—with different shared-memory transport settings—that was not terminated gracefully.

To correct problem (a), if you do not intend to monitor the application that has different shared-memory settings on the same host, you can use another domain ID for the monitoring topics, both in *RTI Monitor* and in the DDS applications that you want to monitor. If you intend to monitor all the DDS applications in that domain on the same host, make sure that all the applications running on the same host with the same domain ID have consistent shared-memory transport settings.

The QoS profile used by *RTI Monitor* is in ***<installation directory>*/config/rtimonitor_qos_profiles.xml**. The transport settings in this profile need to be consistent with the transport settings in all the DDS applications that are running on the same host with the same domain ID. All shared-memory transport settings are specified under the **participant_qos** and have property names that begin with **dds.transport.shmem.builtin**. See the documentation for *RTI Monitoring Library* for an explanation of the transport settings.

To correct problem (b), use the **ipcrm** command to clean up the shared-memory and shared-semaphore resources. See the *RTI Data Distribution Service Platform Notes* for

details. You can also run *RTI Monitor* and the DDS application that you want to monitor with another domain ID that doesn't have any shared-memory or shared-semaphore resources left-over from previous runs.

4.5 Unable to Create Participant in DDS Application

If you see error messages similar to the following:

```
[CREATE Participant]RTIOsapiLibrary_open:error opening library
rtimonitoringnothing.dll
[CREATE Participant]DDS_DomainParticipantMonitoring_
initializeMonitoringLibrary:
ERROR: Failed to get load monitoring library
[CREATE Participant]DDS_DomainParticipantMonitoring_initializeI:
!create monitoring library instance
[CREATE Participant]DDS_DomainParticipant_createI:!create builtin
monitoring support
[CREATE Participant]DDS_DomainParticipantFactory_create_participant_
disabledI:!create participant
```

These messages most likely mean that your DDS application is configured to load the monitoring library dynamically, but you don't have the monitoring library in your path.

If you are running on a Linux system, make sure that your **LD_LIBRARY_PATH** environment variable includes the monitoring library. If you are running on Windows, make sure that your **PATH** environment variable includes the monitoring library.

If you are seeing error messages similar to the following:

```
[CREATE Participant]DDS_DomainParticipantFactory_set_default_-
participant_qos:ERROR: Inconsistent QoS (more information at WARN ver-
bosity level)
[CREATE Participant]DDS_DomainParticipantFactory_load_profilesI:ERROR:
loading profiles
[CREATE Participant]DDS_DomainParticipantFactory_create_participant_-
disabledI:ERROR: loading profiles
```

These messages most likely mean that you are using a lot of properties in ParticipantQos to configure monitoring, and **participant_property_string_max_length** or **participant_property_list_max_length** in the **ResourceLimitsQosPolicy** in DomainParticipantQos is not large enough to accommodate all the properties. Try increasing those values in your DDS application to fix the problem.

4.6 Regular Version of RTI Shapes Demo Stops Working

If you have gone through the demo in Chapter 3 and now notice that your regular *RTI Shapes Demo* executable (the one that is statically linked) is no longer working, follow the instructions in Section 3.6 to restore the demo to its original state.

4.7 Not Receiving Monitoring Data due to Inconsistent QoS

If you see an error message similar to the following:

```
WARN : com.rti.dds.monitor.util.DebugDataReaderLis-
tener.on_requested_incompatible_qos(Unknown Source) : - topic: rti/
dds/monitoring/domainParticipantDescription : RequestedIncompati-
bleQosStatus[total_count=1, total_count_change=1,
last_policy_id=Durability, policies=[QosPolicyCount[policy_id=Dura-
bility, count=1]]]
```

This message most likely means that the internal DDS DataWriters created by *RTI Monitoring Library* for publishing monitoring topics have QoS that are incompatible with the the QoS of the internal DDS DataReaders created by *RTI Monitor* for subscribing to monitoring topics.

If you see this error message, try specifying the **rti.monitor.config.qos_library** and **rti.monitor.config.qos_profile** properties in the DDS application that has monitoring turned on, to ensure that the internally created DDS DataWriters are using the correct QoS values. The default QoS values used for the internally created DDS DataReaders are listed in **RTIMonitoringQosLibrary** and **RTIMonitoringPublishingQosProfile** in the file ***<RTI Monitoring Library installation directory>*/resource/xml/MONITORING_QOS_PROFILES.xml** in the *RTI Monitoring Library* bundle. Refer to *RTI Monitoring Library*'s documentation in the *RTI Monitoring Library* bundle for an explanation of the QoS settings that are required to specify the QoS library and profile.

4.8 Not Receiving Monitoring Data for DDS Entities

Some of the monitoring topics (the *description* monitoring topics) can have data that is larger than what is supported by the default DDS transport settings, especially for cases in which a lot of propagated properties are added to the PropertyQosPolicy, or a large UserDataQosPolicy, TopicDataQosPolicy or GroupDataQosPolicy is involved. By default, asynchronous publishing is used for the writers in *RTI Monitoring Library* for these monitoring topics to resolve the large data issue—transport settings and the maximum type-code serialized size are left at the default values.

The maximum type-code serialized size and transport settings *must be* consistent between *RTI Monitor* and the DDS application that has monitoring turned on. By keeping the maximum type-code serialized size and all the transport settings at default values in the QoS profile used by *RTI Monitor*, all monitored DDS applications that use default settings will work with *RTI Monitor* out of the box.

If you are not receiving monitoring data, it is most likely because you do have inconsistent transport settings or inconsistent maximum type-code serialized size settings between *RTI Monitor* and your DDS application that has monitoring turned on (maybe you are not using the default maximum type-code serialized size or transport settings in the monitored DDS application).

If your monitored DDS application is required to use a large maximum serialized type-code size or transport settings that support large data, you will need to change the corresponding settings in the QoS profile used by *RTI Monitor.* The maximum type-code serialized size is configured under **resource_limits** for the **participant_qos**; transport settings are configured under **property** for the **participant_qos**. The QoS profile used by *RTI Monitor* is in **<installation directory>/config/rtimonitor_qos_profiles.xml**.

A sample large-data QoS profile is shipped with the *RTI Monitoring Library* bundle for your reference; it has large-data support turned on for both the UDPv4 and shared-memory transports, and uses large maximum type-code serialized size support. If you need to use large data or large type-code in your DDS application, you can use this provided large-data QoS profile in the monitored application and also uncomment the corresponding transport and maximum type-code serialized size settings in the *RTI Monitor* QoS profile; this will enable consistent large maximum type-code serialized size and large-data transport settings. Please see the documentation in the *RTI Monitoring Library* bundle for an explanation of this large-data QoS profile.

4.9 Not Seeing Type Code for Some Entities in Description Panel

If the type code for your user data type is larger than the default maximum type-code serialized size, the IDL for that data type may not show up in the DataType tab in *RTI Monitor's* Description Panel. However, it should not affect the rest of the monitoring data.

To see the IDL representation of large type-code in *RTI Monitor,* you can increase the maximum type-code serialized size, both in the monitored DDS application and in *RTI Monitor.* However, if you do that, you will also need to increase the values in the transport settings to support large data in DDS discovery traffic—both in the monitored DDS application and in *RTI Monitor*.

A sample large-data QoS profile is shipped with the *RTI Monitoring Library* bundle; it has large-data support turned on for both the UDPv4 and shared-memory transports, and a larger maximum type-code serialized size. Please see the documentation in the *RTI Monitoring Library* bundle for an explanation of the large-data QoS profile. If you use the provided large-data QoS profile, you can uncomment the corresponding settings in the *RTI Monitor* QoS profile to enable support for large type-code and large data. Both the monitored DDS application and *RTI Monitor* must have a consistent maximum type-code serialized size and consistent transport settings. The QoS profile used by *RTI Monitor* is in **<installation directory>/config/rtimonitor_qos_profiles.xml**.

4.10 Running out of Memory

If *RTI Monitor* is running out of memory, use a smaller value for the **-historyDepth** command-line option or run *RTI Monitor* on a 64-bit machine.

4.11 Running without an Active Network Interface

If you run *RTI Monitor* on a computer that does not have an active network interface, you may see an error message stating "No interface found enabled for multicast."

Modify the QoS profile used by *RTI Monitor* to turn off UDPv4 and only use the shared-memory transport:

```
<participant_qos>
......
    <transport_builtin>
            <mask>SHMEM</mask>
    </transport_builtin>
</participant_qos>
```

The QoS profile used by *RTI Monitor* is in ***<installation directory>*/config/rtimonitor_qos_profiles.xml**.