

RTI Analyzer

Getting Started Guide

Version 5.0



Your systems. Working as one.



© 2006-2012 Real-Time Innovations, Inc.
All rights reserved.
Printed in U.S.A. First printing.
August 2012.

Trademarks

Real-Time Innovations, RTI, and Connexant 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.

Technical Support

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

Contents

1 Introduction

1.1 Overview	1-1
--------------------	-----

2 Installation

2.1 Installing Analyzer.....	2-1
2.2 Installing the License File.....	2-2
2.3 Uninstalling Analyzer	2-3

3 Starting Analyzer

3.1 Windows Systems	3-1
3.2 Solaris and Linux Systems	3-2
3.3 Running without an Active Network Interface	3-2

4 Configuring Analyzer

4.1 Specifying a QoS Profile	4-2
4.2 Configuring Analyzer from the GUI.....	4-3

5 Tutorial

5.1 Setting Up Shapes Demo.....	5-2
5.2 Connecting Analyzer to Shapes Demo	5-5
5.3 Working with the Entity Tree	5-7
5.3.1 Changing the Tree Layout	5-8
5.3.2 Filtering Out Specific Entities.....	5-9
5.4 Viewing Entity Details.....	5-12
5.5 Debugging Object Communication Problems with “Match Analysis”	5-13
5.5.1 Understanding Type Consistency.....	5-16
5.6 Filtering the Match Results Tree.....	5-17
5.6.1 Filtering the Contents of the Match Analysis Tree.....	5-17
5.6.2 Analyzing Selected Objects Only.....	5-18
5.7 Correcting the Circle’s Liveliness QoS	5-20
5.8 Saving and Comparing Snapshots.....	5-21

Chapter 1 Introduction

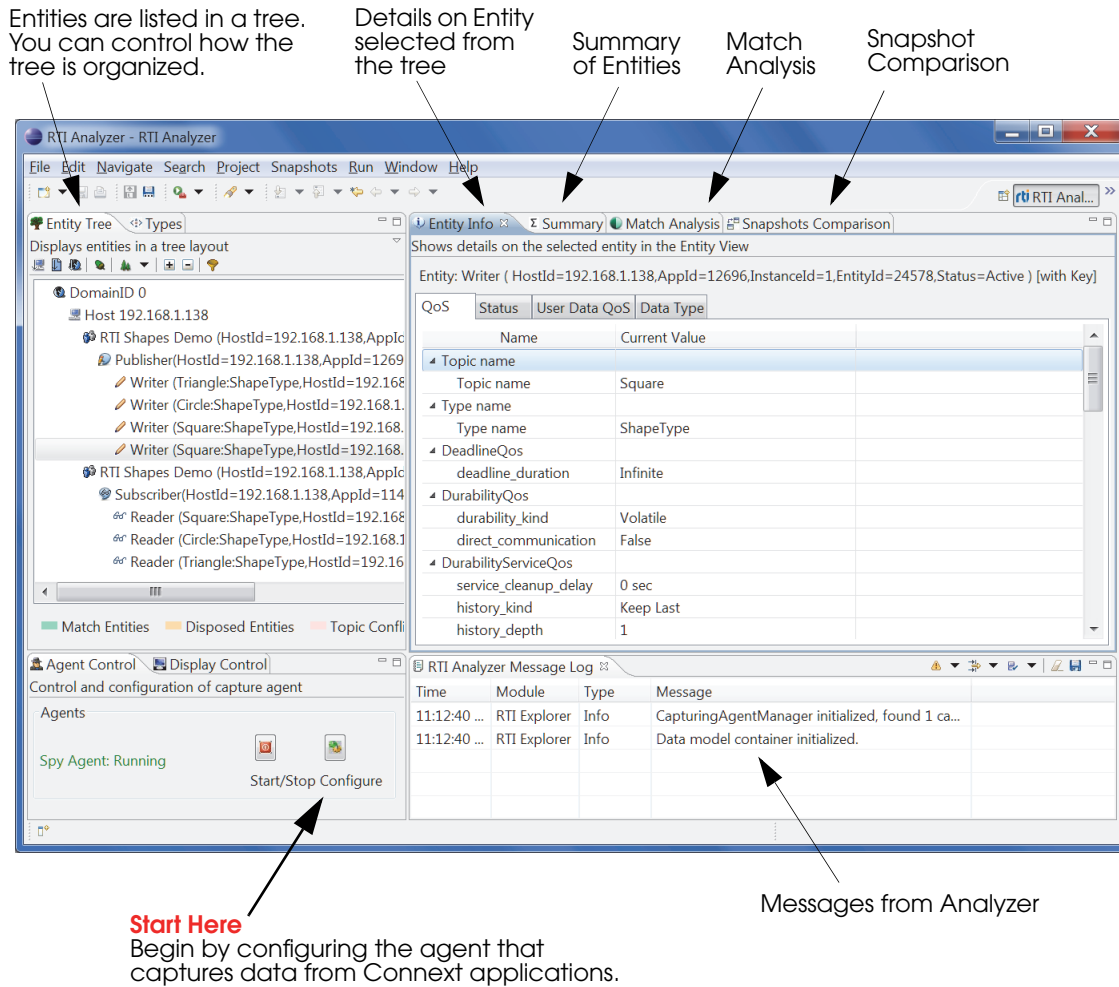
Welcome to *RTI[®] Analyzer*—the debugging and analysis tool for *RTI Connex[™]* (formerly *RTI Data Distribution Service*) applications. *Analyzer* provides an easy way to see the topology of a *Connex* application and diagnose peer-to-peer communication problems.

1.1 Overview

Analyzer “listens” to the declarations (meta-traffic) sent from *Connex* applications running in a specific domain. It builds an internal database of the nodes, applications, topics, and *Connex* objects. This information is then displayed in one or more tree view windows, each organized according to your specifications. For example, you could have one window that lists the *Connex* objects (*DomainParticipants*, *DataWriters*, *DataReaders*, etc.) associated with each Topic; in another window, you could list the objects by host or domain.

If you have applications that are not fully communicating, the Match Analysis feature can help you determine which objects (*DataWriters* and *DataReaders*, or *Publishers* and *Subscribers*) are ‘matched’ up and which ones are not. If you have objects with incompatible QoS settings, this feature will point them out.

The *Connex* objects are shown in a familiar explorer-style tree that organizes the application and object relationships. *Analyzer* also allows you to select an object from the tree and display its status and QoS properties. This facility makes it easy to understand your distributed application.



In this example, Shapes Demo has created Writers for 2 squares, 1 triangle, and 1 circle. The squares use the same Topic name, with color as the 'key.' You can modify how the tree is organized and filter out Entities that are not of interest.

Analyzer's main window is composed of the following sections.

- ❑ **Agent Control** Allows you to start, stop, and configure *Analyzer's* Spy agent, which collects discovery data from *Connex*t applications.
- ❑ **Display Control** Allows you to pause the automatic redisplay of data, and to control how often data on the display is refreshed.
- ❑ **Entity Tree** The tree is a sorted list of *Connex*t objects. *Analyzer* “listens” to the declarations and lists the discovered objects. The Entity Tree uses Windows Explorer-style displays and techniques for expanding and collapsing the tree. You can double-click the folders to expand the lists of hosts, Participants, Writers, Topics, etc.
- ❑ **Types** Displays the data types for each discovered Topic. This view highlights type-related conflicts such as two Topics with the same name that use different data types, or two data types with the same name that have different structures.
- ❑ **Entity Info** Contains subordinate tabs for QoS, Status, and Data Type. The contents of these tabs are determined by the object selected in the tree. The QoS tab can also be used to temporarily change mutable QoS policies.
- ❑ **Summary** Shows you how many objects of each type have been discovered.
- ❑ **Match Analysis** Shows you which Writers/Readers or Publishers/Subscribers are ‘matched’ to each other. This information can help you diagnose communications problems between applications, such as incompatible QoS settings.
- ❑ **Snapshots Comparison** Provides a way to quickly compare the hierarchy of *Connex*t entities in two different systems. You can take snapshots of various systems, then compare them in a graphical manner to see how they differ.
- ❑ **Message Log** Displays messages from *Analyzer*. You can control which type of messages are displayed (severe, warning, info, etc.).

Context-sensitive help is available when you press F1 on Windows systems or Ctrl-F1 on Solaris and Linux systems, or select the help button  found in the lower left corner of some dialogs.

[Chapter 2](#) describes the installation process and important information about the required license file.

[Chapter 3](#) explains how to start *Analyzer*.

[Chapter 4](#) describes how to configure *Analyzer*.

[Chapter 5](#) presents a short tutorial to introduce the major *Analyzer* features. To follow along with the tutorial, you will start the *Shapes Demo* application and then walk through the various *Analyzer* windows and options. Once you have completed the tutorial, *Analyzer* operation is intuitive.

Chapter 2 Installation

Analyzer is included with *RTI Connex Messaging*. Use the installation instructions in this chapter only if you are installing *Analyzer* independently (not as part of *Connex Messaging*).

After installation, see [Chapter 3: Starting Analyzer](#); then we recommend that you walk through [Chapter 5: Tutorial](#).

2.1 Installing Analyzer

In the following instructions, replace *version* to match your version of *Analyzer*.

1. To install *Analyzer*, simply run the installer and follow the on-screen instructions.
2. Optional: Download and install *RTI Shapes Demo*. This is only required if you want to follow the steps in [Chapter 5: Tutorial](#). *Shapes Demo* is available from RTI's Downloads page (<http://www.rti.com/downloads>) in the section titled Interactive Demonstration. *Shapes Demo* runs on Windows and Linux systems. Follow the installation instructions that accompany the distribution file.

Additional Steps for Linux Systems

1. `cd <location of RTI_Analyzer-<version>-SetupLinux.sh>`
2. `chmod 755 RTI_Analyzer-<version>-SetupLinux.sh`
3. `./RTI_Analyzer-<version>-SetupLinux.sh`

Additional Notes for Solaris Systems

If you are root/administrator when you install *Analyzer*, then only the root user/administrator will be able to run *Analyzer* unless you do the following:

- Make sure other users have read access to all files in **<Analyzer installation directory>/apps/RTI Analyzer <version>**.
- Provide other users with 'execute' permission on directories and executables under the **<Analyzer installation directory>/apps/RTI Analyzer <version>** directory.
- Make the **<Analyzer installation directory>/apps/RTI Analyzer <version>/eclipse/configuration** directory writable by users other than root/administrator.

2.2 Installing the License File

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

If you have *Connex* Messaging and you want to use *RTI Connex Launcher*¹ to start *Analyzer*²:

By default, *Launcher* looks for the license file **rti_license.dat** in the top-level directory where you installed *Connex* Messaging. If you choose to save the license file elsewhere, you can configure *Launcher* to look in a different location by using its Configuration tab.

Otherwise:

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

1. The last saved, valid license location for the current user (if *Analyzer* has been run before).
2. The file **<Analyzer installation directory>/rti_license.dat**
3. 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**).
4. The file **rti_license.dat** in the current working directory.

1. *Launcher* is a convenient GUI-based tool that can start and configure all of your *Connex* Messaging components, including *Analyzer*.

2. Even if your distribution of *Connex* Messaging is not license-managed, you always need a license file to run *Analyzer*.

5. The file **rti_license.dat** in the directory specified by the environment variable **NDDSHOME**.

If *Analyzer* 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**.

2.3 Uninstalling Analyzer

On a Windows system, select **Start, Programs, RTI, RTI Analyzer <version>, RTI Analyzer Uninstaller**. (Or you can select **Start, Control Panel, Add/Remove Programs**¹.)

On a Linux or Solaris system, run:

```
<Analyzer installation directory>/uninstall
```

Note: If *Analyzer* was installed on a remote file system, the following files will not be removed:

- ☐ .install4j
- ☐ jre
- ☐ jre/bin
- ☐ jre/lib
- ☐ jre/lib/ext
- ☐ jre/lib/i386
- ☐ jre/lib/i386/xawt
- ☐ jre/lib/i386/native_threads
- ☐ jre/lib/i386/client
- ☐ jre/lib/font

To completely uninstall *Analyzer* on a remote file system, run:

```
rm -rf <install-dir>
```

1. On Windows Vista systems, select **Start, Control Panel, Programs and Features**.

Chapter 3 Starting Analyzer

- ❑ Workspaces created with older versions of *Analyzer* will not work. You must use a new workspace with *Analyzer*.
- ❑ *Analyzer* requires more memory as the size of data set it has to process increases. *Analyzer* is configured for 256 MB of memory by default, which is sufficient for a total of 2,000 data readers and data writers.

For larger data sets, make this change before starting *Analyzer*: In the file `<install-dir>/apps/RTI_Analyzer_<version>/eclipse/rdp.ini`, change `"-Xmx256m"` to `"-Xmx512m"` or `"-Xmx1024m"`, depending on the total number of endpoints your system contains.

3.1 Windows Systems

If you installed *Analyzer* as part of *RTI Connexx Messaging*, either:

- ❑ From the **Start** menu, select **All Programs, RTI Connexx <version>, RTI Connexx <version> Launcher**. When Launcher starts, select **Analyzer** from the Tools tab.
- or
- ❑ From the **Start** menu, select **All Programs, RTI Connexx <version>, RTI Connexx Messaging <version> Components, RTI Analyzer <version>, RTI Analyzer**.

If you installed *Analyzer* separately:

- ❑ From the **Start** menu, select **All Programs, RTI, RTI Analyzer <version>, RTI Analyzer**.

3.2 Solaris and Linux Systems

Enter the following command:

```
$ <Analyzer installation directory>/scripts/rtianalyzer
```

3.3 Running without an Active Network Interface

To use *Analyzer* on a computer that does not have an active network interface, you must configure the tool so that it only uses shared memory, as follows:

1. Click the **Configure**  button to open the Domain Configuration dialog.
2. Select the **Transport Properties** tab and clear the check-box for **Enable Multicast**. (If this option is disabled, click the **Add** button first.)
3. Click **OK**.

Chapter 4 Configuring Analyzer

There are two parts to *Analyzer* configuration—domain configuration and QoS parameters.

Domain configuration consists of providing *Analyzer* with the list of domain IDs that you would like to analyze. To access the dialog seen in [Figure 4.1](#), select the **Configure** button  in the **Agent Control** view (lower left corner). From the Agent Configuration window, in the lower left portion of this view, you can select Add or Delete to add or delete domain IDs to be analyzed. Each domain ID can have separate QoS parameters.

There are two ways to configure *Analyzer* QoS parameters:

- ❑ [Specifying a QoS Profile \(Section 4.1\)](#)
- ❑ [Configuring Analyzer from the GUI \(Section 4.2\)](#)

You can use either method; however you cannot use both at the same time.

The QoS profile method gives you greater control over the configuration because you can change *all* DomainParticipant QoS settings. It's also a simple way to use the exact same profile as the *Connex*t application that you are analyzing. The GUI method allows you to change a subset of QoS values.

If you do not configure *Analyzer*, it will use the same default configuration as *Connex*t. The default configuration file for *Connex*t is located here:

- ❑ `$NDDSHOME1/resource/qos_profiles_<version>5.0.x2/xml/NDDS_QOS_PROFILES.xml`

1. NDDSHOME represents the installation directory for *Connex*t.

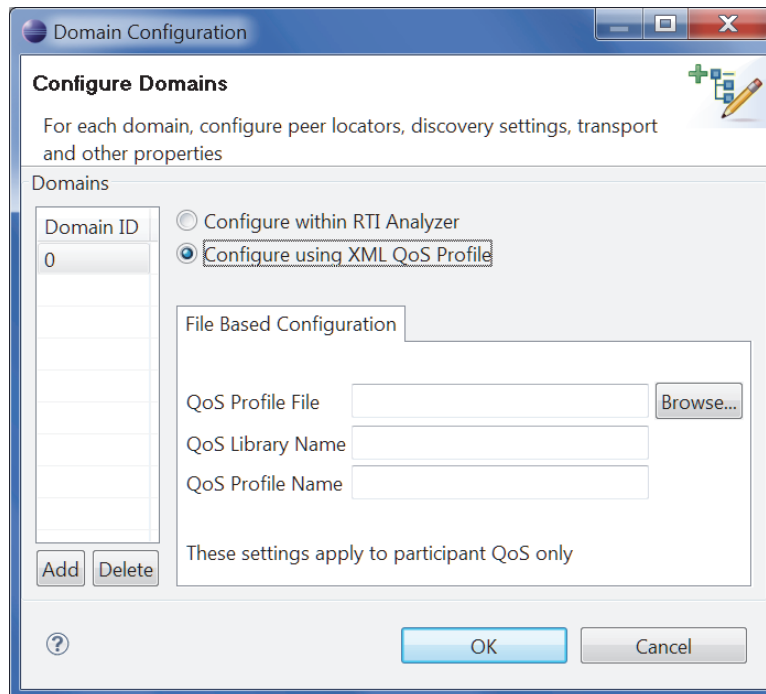
2. *x* represents a version-specific number for the current release.

4.1 Specifying a QoS Profile

The **Domain Configuration** dialog allows you to specify an XML configuration profile from a file.

To access the dialog seen in [Figure 4.1](#), select the **Configure** button  in the **Agent Control** view (lower left corner). Select the option labeled **Configure using XML QoS Profile**. Then select the profile file, library, and profile that you want to use.

Figure 4.1 Selecting an XML QoS Profile



You can create your own profile file and use it to make changes to any DomainParticipant QoS. *Analyzer* will ignore other QoS settings in the profile (because it does not create any other entities).

Note: If you make a mistake in entering the File Name, Library Name or Profile Name, *Analyzer* will print an error message the next time you click the “Start/Stop” button in the Agent Control window.

4.2 Configuring Analyzer from the GUI

You can change commonly used configuration settings from the **Configure Domains** dialog. If you need to change a parameter not exposed through the GUI, please use XML profiles (see [Section 4.1](#)).

To access this dialog, select the **Configure** button  in the **Agent Control** view (lower left corner). Then select the option labeled **Configure within RTI Analyzer**.

Then you can select from several tabs, described in the following figures.

Figure 4.2 **Peer Locators Tab**

*The **Peer Locators** tab allows you to set the peer list used by Connex to discover other entities. By default this list includes shared memory, UDP loopback and multicast. Please refer to the RTI Core Libraries and Utilities User's Manual for more information about discovery peers.*

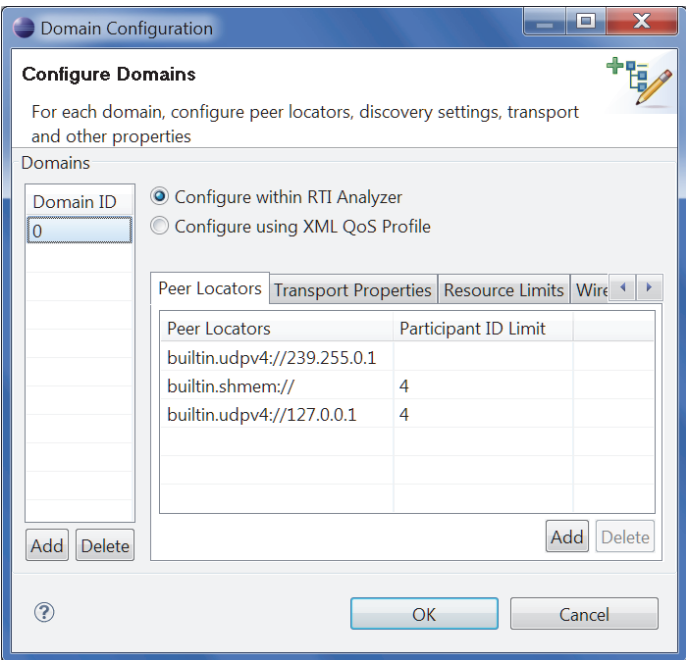


Figure 4.3 **Transport Properties**

*The **Transport Properties** tab allows you to set the properties of the transports used by Connex. By default, the UDPv4 and shared-memory transports are enabled; there are also tabs for UDPv6 and RTI Secure WAN Transport. You may also enable or disable multicast from this tab.*

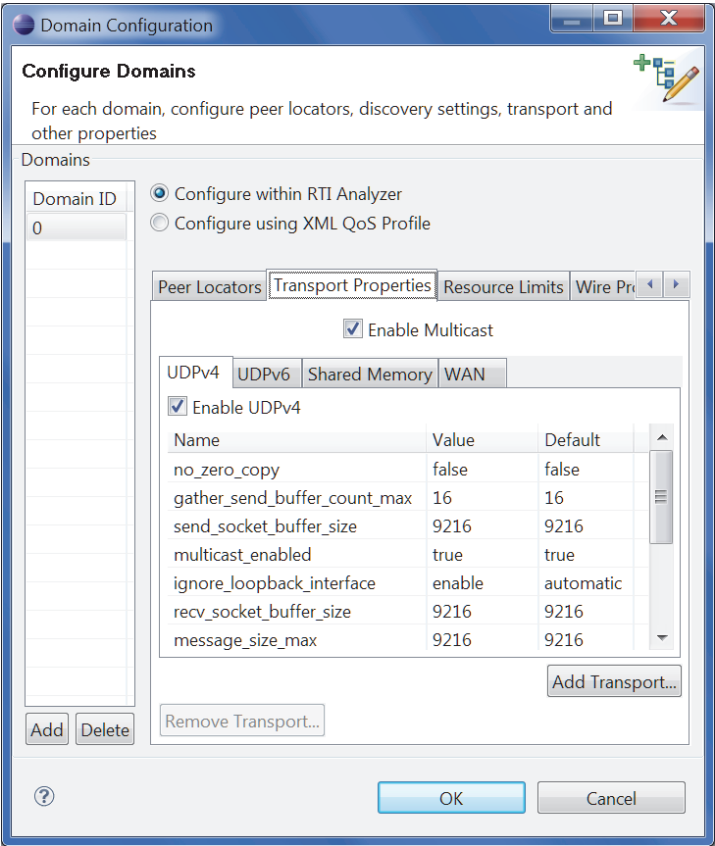


Figure 4.4 Resource Limits Tab

The Resource Limits tab allows you to change some commonly used resource limits for Connex.

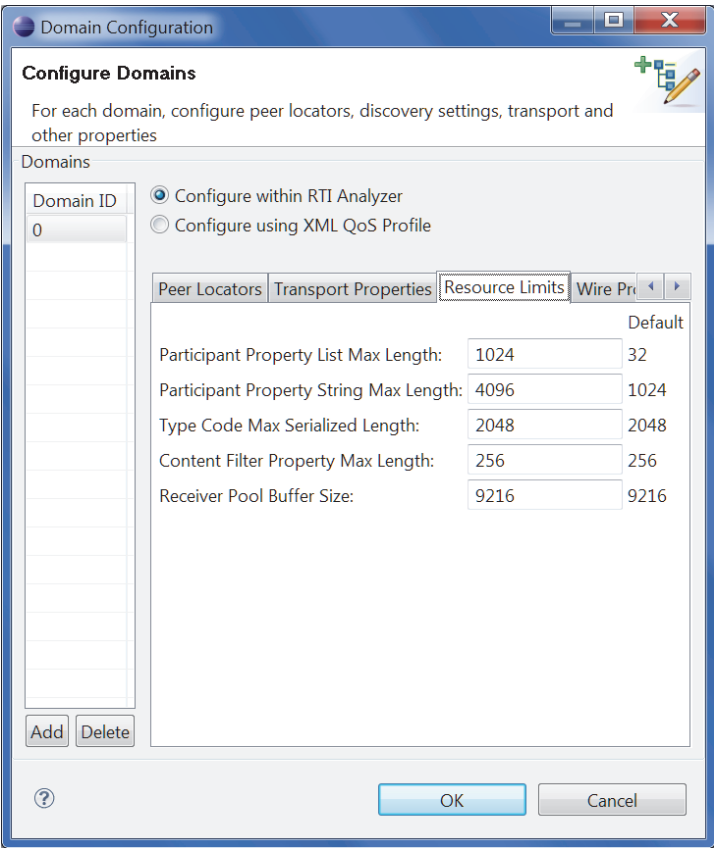


Figure 4.5 Wire Protocol Tab

The *Wire Protocol* tab allows you to change configuration parameters for the Real-Time Publish Subscribe protocol used by Connex.

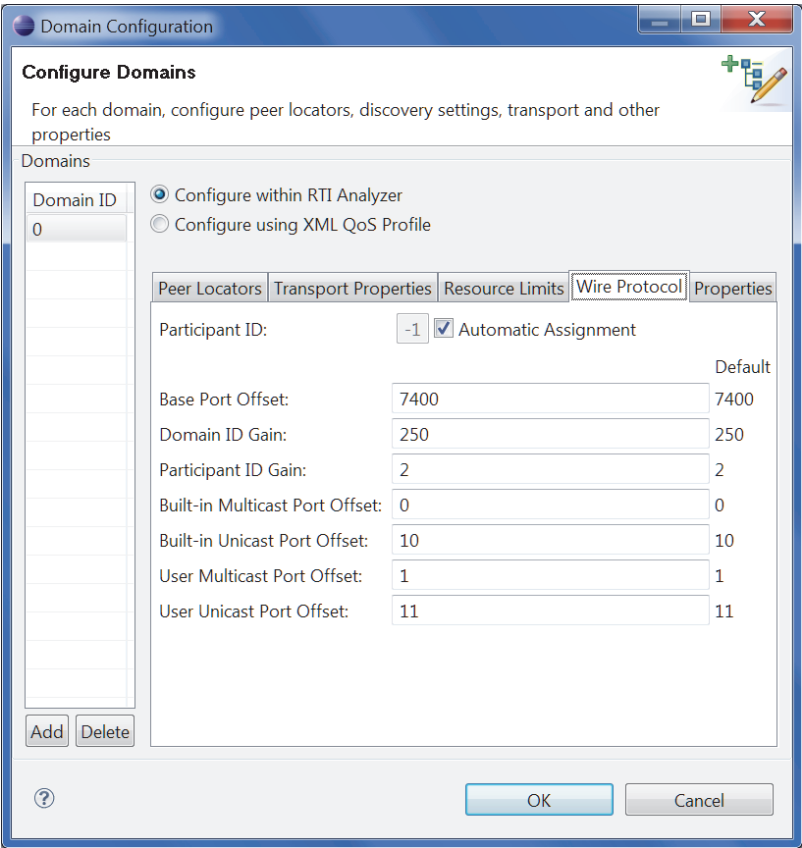
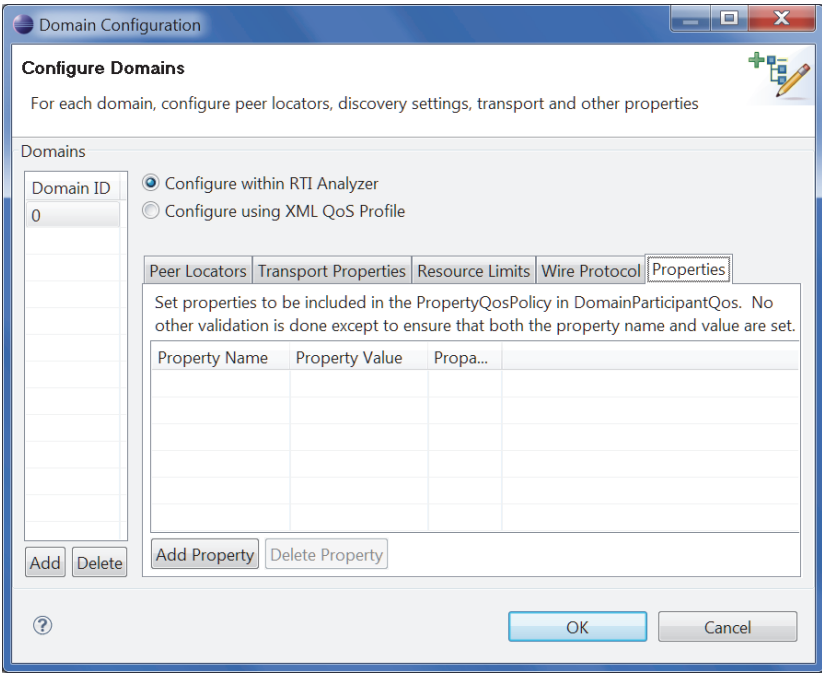


Figure 4.6 Properties Tab

The **Properties** tab allows you to add Properties to the DomainParticipant's Property QoS policy. The properties have a wide range of uses, from configuring transports to being propagated with the Connex discovery information.

Please see the RTI Core Libraries and Utilities User's Manual for more information on properties.



Chapter 5 Tutorial

The best way to learn about *Analyzer* is to start using it. The goal of this tutorial is to help you become familiar with *Analyzer*'s main features. We will use *RTI Shapes Demo* as an easy way to start a *Connex* publish-subscribe application; then we will use *Analyzer* to analyze the application's entities. We'll create a communications problem on purpose, then walk through the steps to diagnose and correct it.

Shapes Demo runs on Windows and Linux systems. If you are using a different operating system, you can still read through this tutorial to get a sense of the features, then you can experiment with the Hello World example described in the *RTI Core Libraries and Utilities Getting Started Guide*.

This tutorial assumes you have a basic understanding of *Connex* terms such as *Domain-Participants*, *Publishers*, *DataWriters*, *Topics*, and Quality of Service (QoS) policies. For an overview of *Connex* terms, please see the *RTI Core Libraries and Utilities User's Manual*.

The tutorial assumes you have the following software installed (using the same version number for both components):

- ☐ *Shapes Demo*
- ☐ *Analyzer*

See [Chapter 2](#) for installation instructions.

The tutorial walks you through the following exercises:

- ☐ [Setting Up Shapes Demo \(Section 5.1\)](#)
- ☐ [Connecting Analyzer to Shapes Demo \(Section 5.2\)](#)
- ☐ [Working with the Entity Tree \(Section 5.3\)](#)
- ☐ [Viewing Entity Details \(Section 5.4\)](#)
- ☐ [Debugging Object Communication Problems with "Match Analysis" \(Section 5.5\)](#)

- ❑ [Filtering the Match Results Tree \(Section 5.6\)](#)
- ❑ [Correcting the Circle's Liveliness QoS \(Section 5.7\)](#)
- ❑ [Saving and Comparing Snapshots \(Section 5.8\)](#)

5.1 Setting Up Shapes Demo

We'll use *Shapes Demo* to publish and subscribe to Topics which will appear in the Demo's window as colored moving shapes. Each shape is a Topic; the shape's color is used as the Topic key.

Shapes Demo showcases the capabilities of *Connex*t, including publish-subscribe communication, quality of service (QoS), fault tolerance and automatic discovery. The demo is a turnkey graphical application and does not require any programming.

- ❑ This tutorial assumes you are using a computer that has an active network interface. If this is *not* the case, see [Running without an Active Network Interface \(Section 3.3\)](#).
- ❑ *Shapes Demo* uses Domain ID 0 by default. If you want to use a different Domain ID, change the domain ID in the Configuration dialog that appears when you start *Shapes Demo*. Or you can start *Shapes Demo* with the **-domainId <#>** option (for example, to use domain ID 5, enter: `rtishapesdemo -domainId 5`).

To Begin the Tutorial:

1. Start *two* copies of *Shapes Demo*.

- To start *Shapes Demo* on Linux systems:

Enter the following in a command shell:

```
> <Shapes Demo install directory>/scripts/rtishapesdemo
```

- To start *Shapes Demo* on Windows systems:

If you have *Connex*t Messaging, you can start *Shapes Demo* from *Launcher's* Utilities tab. Or from the Windows **Start** menu, navigate to **RTI Connex**t <version>, **RTI Connex**t Messaging <version> **Components**, **RTI Shapes Demo** <version> and select **RTI Shapes Demo**.

The figure to the right shows the main *Shapes Demo* window and the Configuration dialog that appears at start-up.¹

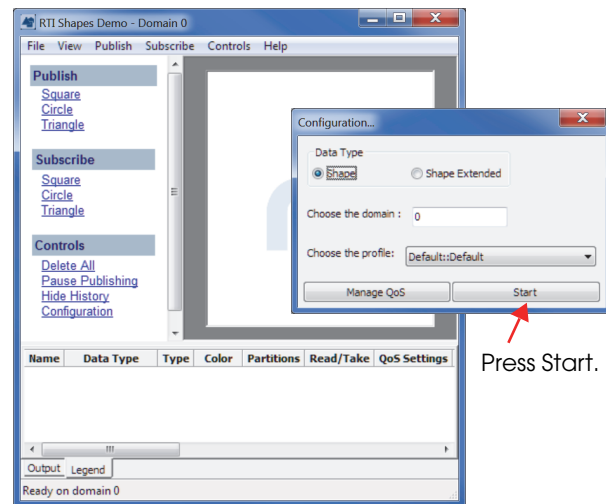
No configuration changes are necessary.

If you choose a different domain ID, make sure to use the same ID in both instances of *Shapes Demo* or they will not communicate.

Press **Start**.

Repeat, so you are running two instances of *Shapes Demo*.

We'll refer to the first Demo window as the Publisher Demo and the second one as the Subscriber Demo.



Press Start.

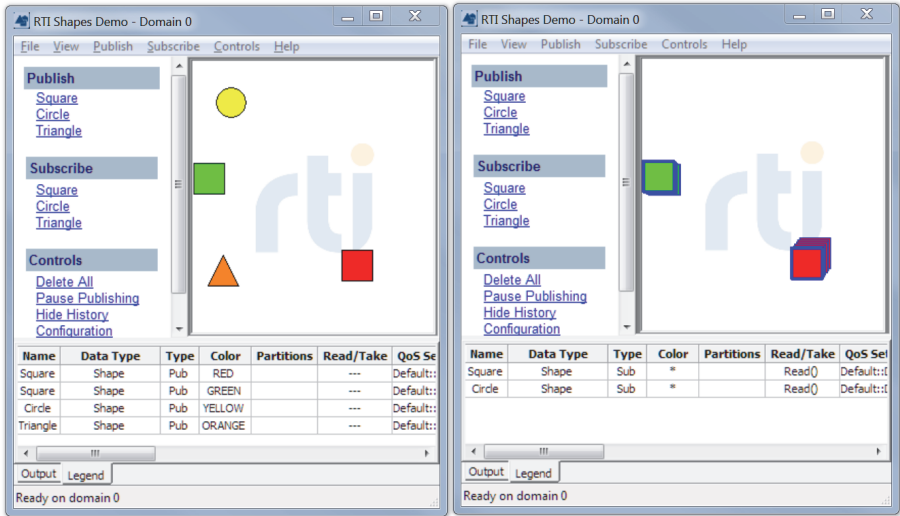
2. In the Publisher Demo, create 2 squares, 1 triangle, and 1 circle (all with default QoS settings):
 - a. Under **Publish**, click on **Square**.
 - b. In the **Create New Publisher** dialog, select **RED** and click **OK**. (Use the defaults for the other settings).
You should see a red square moving in the display area.
 - c. Use the same process to create a green square, a yellow circle, and an orange triangle (with default settings).
You've just created 4 Writers (2 Squares, a Triangle, and a Circle).
3. In the Subscriber Demo, subscribe to squares and circles (but not triangles):
 - a. Under **Subscribe**, click on **Square** and click **OK**. (Leave the default settings.)
Now you have 1 Reader that will receive data for all squares (all colors).
 - b. Use the same process to create a Reader for circles—with *Liveliness* set to *AUTOMATIC* and *Lease Duration* set to 50 ms.

1. The Configuration dialog may not appear at start up if you invoke *Shapes Demo* with the `-dataType <Shape|ShapeExtended>` command-line option.

Recall that we created the Circle Writer with default settings, so its Liveliness Lease Duration is INFINITE. The QoS compatibility rules for Liveliness require that the Writers' Lease Duration must be $\leq$ the Reader's. By setting the Reader's to 50 ms., we are intentionally causing the Circle Reader/Writer pair to be incompatible. This will demonstrate how *Analyzer* points out the error and then allows you to correct it by writing a new QoS value to *Shapes Demo*.

Do not create a Reader for triangles—this will demonstrate how *Analyzer* points out Writers without matching Readers.

Your two *Shapes Demo* windows should look like this:



Publisher

Subscriber

Notice that the Subscriber's shapes have colored borders. This is how you can distinguish between Subscribers (colored borders) and Publishers (thin black borders). By default, Subscribers show the most recent 6 samples—this is why you see extra squares behind the leading one. The most recent shape has a thick border. Older (historical) samples have a thinner border.

Also notice that you do not see a triangle or circle shape in the Subscriber's window. (We didn't create a Reader for Triangles, and the Reader for Circles has an incompatible QoS.)

- c. You may minimize these two windows, we won't need them for a while.

5.2 Connecting Analyzer to Shapes Demo

1. Start *Analyzer*.

- On Linux systems:

Enter the following command in a command shell:

```
> <Analyzer install directory>/scripts/rtianalyzer
```

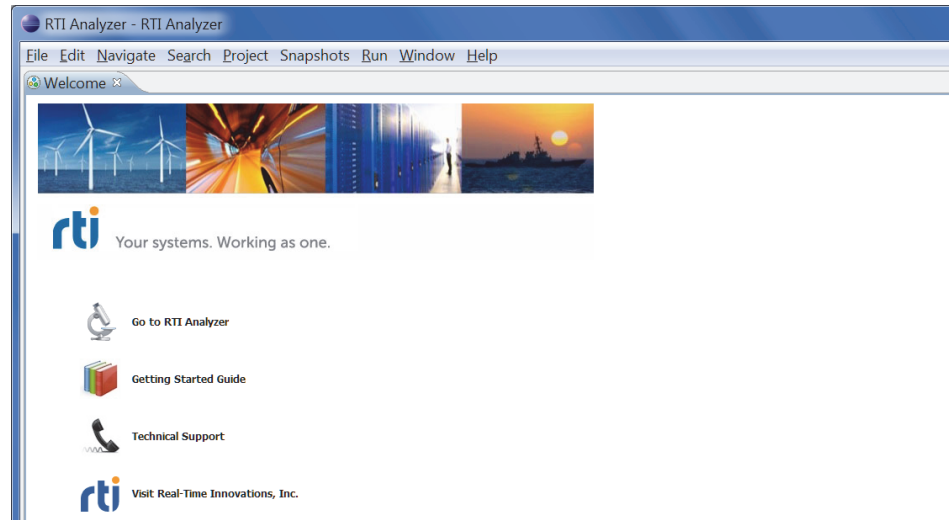
- On Windows Vista systems:

If you have *RTI Launcher*, you can start *Analyzer* from *Launcher*'s Tools tab. Or from the Windows **Start** menu, navigate to **RTI Connexx <version>, RTI Connexx Tools <version> Components, RTI Analyzer <version>**. Right-click **RTI Analyzer** and select **Run as Administrator**.

- On other Windows systems:

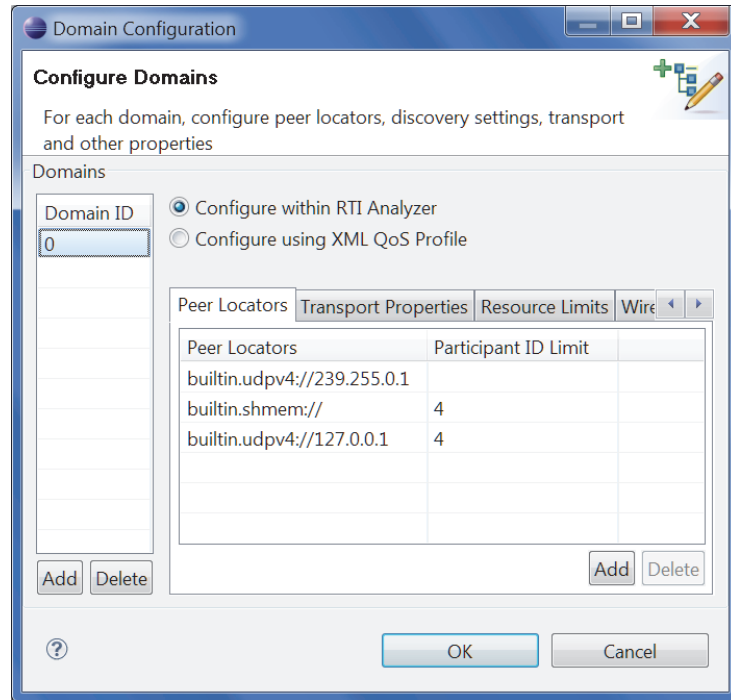
If you have *RTI Launcher*, you can start *Analyzer* from *Launcher*'s Tools tab. Or from the Windows **Start** menu, navigate to **RTI Connexx <version>, RTI Connexx Tools <version> Components, RTI Analyzer <version>** and select **RTI Analyzer**.

Next, select **Go To RTI Analyzer** when you see this Welcome screen¹:



1. The Welcome screen is only displayed the first time you use a new workspace.

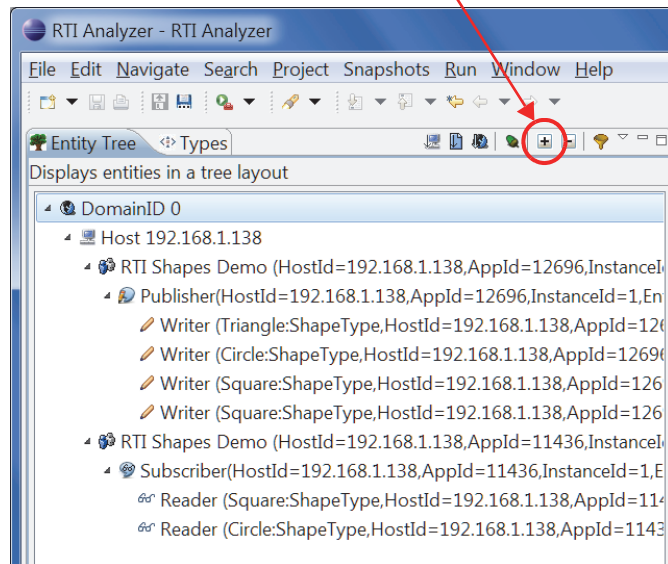
2. In the **Agent Control** view (lower left corner), select the **Configure** button . This is where you will tell *Analyzer* what applications to look for.
3. In the **Domain Configuration** dialog, click **OK**. This will configure the tool to use the default Domain ID, 0, which is the same default value used by *Shapes Demo*. (However, if you started the Demo with a different ID, change it before clicking **OK**.)



4. Back in the **Agent Control** view, select the **Start** button .
5. Within a few seconds, you should see **DomainID 0** in the Entity tree. (Or, if you changed the domain ID, you should see your modified ID.)

6. Select the **Expand Tree** button  to see all the entities:

Click here to expand the tree.



(Note: Your HostId, AppId, etc., will be different from those in the screenshots shown throughout this document.)

5.3 Working with the Entity Tree

The Entity tree shows you what Entities have been discovered. By default, the objects in the tree are sorted in this order:

- Domain ID
 - Host address
 - Application Name (Host ID, App ID, Instance ID, Entity ID)
 - Subscriber/Publisher (Host ID, App ID, Instance ID, Entity ID)
 - Reader/Writer (Topic:DataType, Host ID, App ID, Instance ID, Entity ID)

5.3.1 Changing the Tree Layout

You can change the way the tree is sorted. There are two other predefined layouts (**Sort by Topic** , **Sort by Domain ID** ). Try both buttons to see their effect.

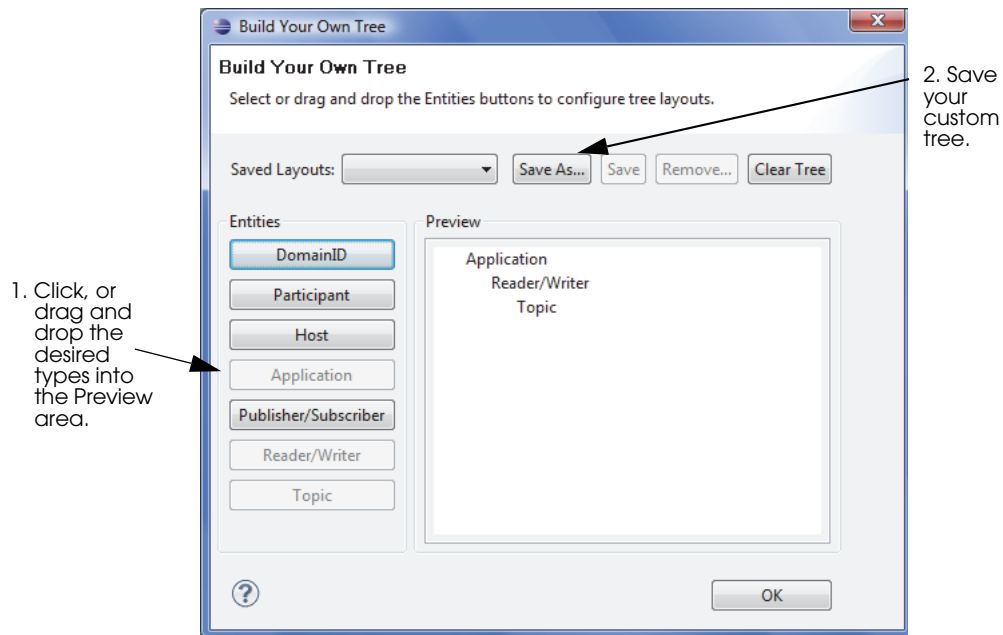
You can create your own tree views by using the **Build Your Own Tree** button .

For example, let's create a view that only shows these objects:

- Application
 - Reader/Writer
 - Topic

To create a custom tree layout:

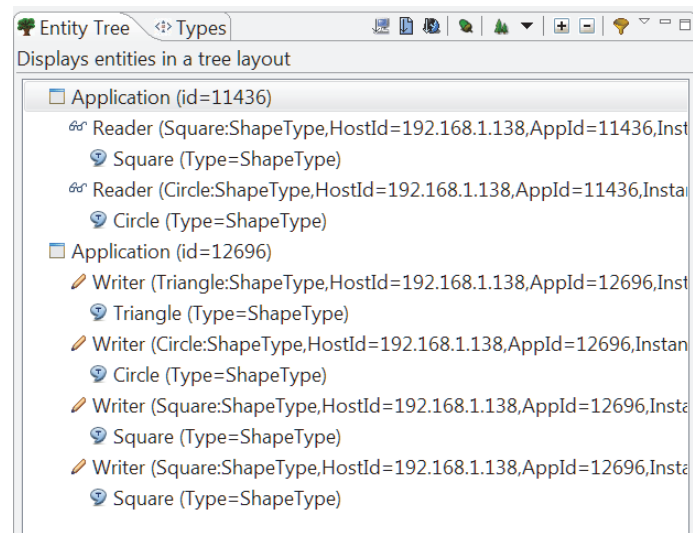
1. From the Entity Tree tab, click the **Build Your Own Tree** button .
2. Click, or drag and drop, the **Application**, **Reader/Writer**, and **Topic** buttons (in that order).



3. To save this tree layout:
 - a. Click **Save As....**
 - b. Enter a name for the view, such as **MyTreeView**.
 - c. Click **OK**.
4. Click **OK** to close the **Build Your Own Tree** dialog.

In the Entity Tree pane, you will see a new button for **Tree Layouts**  — this button only appears when there are custom tree layouts.

5. Click the triangle on the right of the  button and select your new layout from the drop-down list.
6. Select the **Expand Tree** button  to see all the entities in the tree, which should now look like this:



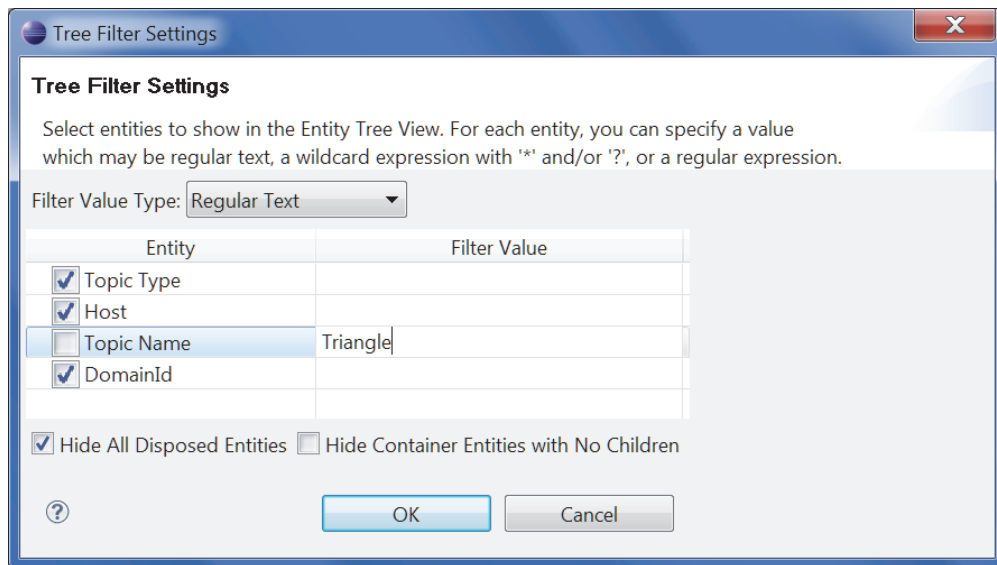
5.3.2 Filtering Out Specific Entities

The **Filter** button  allows you to control which domain IDs, hosts, and Topics appear in the tree.

In the Tree Filter Settings dialog, you can enter text, wildcard expressions, or regular expressions to refine the filter.

For example, let's hide the Topic called **Triangle** for now:

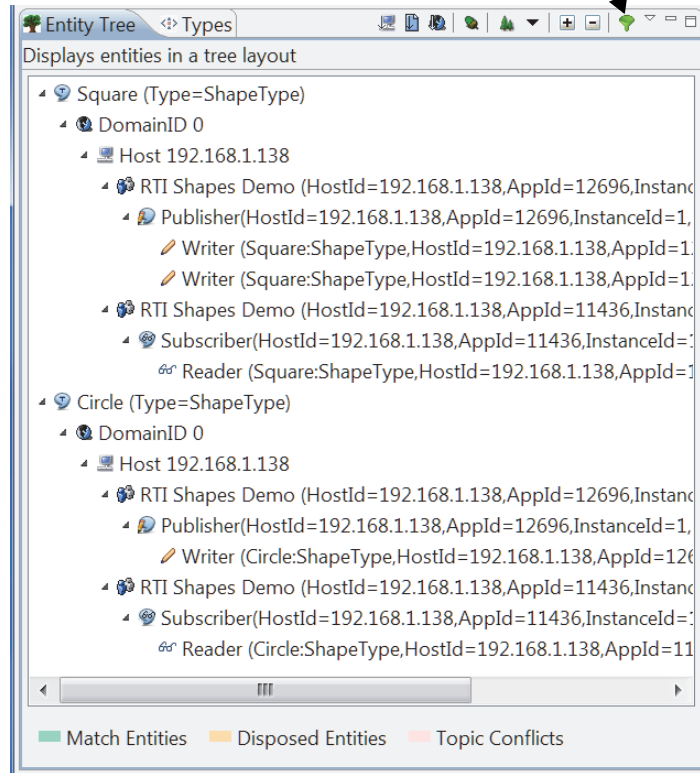
1. Select the **Sort by Topic** button .
2. Select the **Filter** button .
3. In the Tree Filter Settings dialog, clear the **TopicName** checkbox and enter 'Triangle' as shown below:



4. Click **OK** to close the dialog.

5. Select the **Expand Tree** button  to see all the (non-triangle) entities in the tree, which should now look like this:

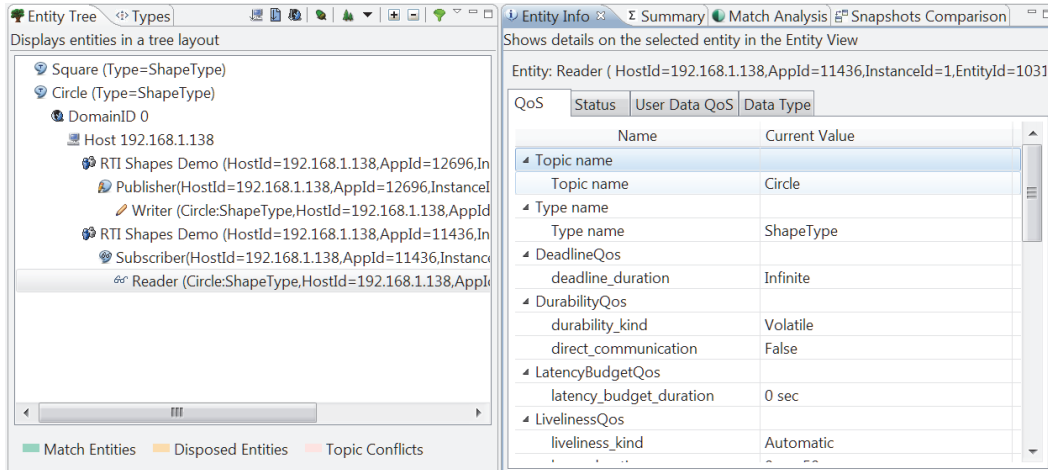
The filter button turns green when a filter is in use.



5.4 Viewing Entity Details

Now let's see what details are available about the Entities in the tree.

1. Select a Reader, such as the one for the Topic named **Circle**.



2. Select the **Entity Info** view. Within that view, select the different tabs to become familiar with their contents:
 - **QoS** Displays the current QoS settings for the selected entity.
 - **Status** Displays the entity's current state, the timestamp from the last update, and the number of state changes. (Not applicable to hosts or applications.)
 - **User Data QoS** Displays the contents of the User Data QoS policy, if any.
 - **Data Type** Displays the definition of the data type used by the selected Writer, Reader, or Topic. (Not applicable to other types of objects in the tree.)

5.5 Debugging Object Communication Problems with “Match Analysis”

Analyzer’s Match Analysis feature helps you see which Writers and Readers are ‘matched’ (and therefore able to communicate with each other) and which ones aren’t. This feature can help you spot incompatible QoS settings, as well as common errors such as misspelled Topic names.

- ❑ As you may recall, in [Setting Up Shapes Demo \(Section 5.1\)](#), we deliberately created an error in our Publisher and Subscriber Demo, so we could see how *Analyzer* helps you diagnose problems. On the publishing side, we created a Circle with default Liveliness (AUTOMATIC kind, with INFINITE lease_duration); on the subscribing side, we created a Circle with Liveliness set to 50 ms. Since the offered lease_duration (INFINITE) is not $\leq$ the requested lease_duration (50 ms), a Writer/Reader pair with incompatible QoS.
- ❑ We also created a Triangle on the publishing side, but not on the subscribing side. While this is not technically an error, it is a situation that you may want to be aware of.

The Match Analysis view uses a tree view to show *Connex*t objects, with the following hierarchy:

- Domain ID
 - Topic
 - Writer
 - Reader

The results are color-coded to make it easy to see:

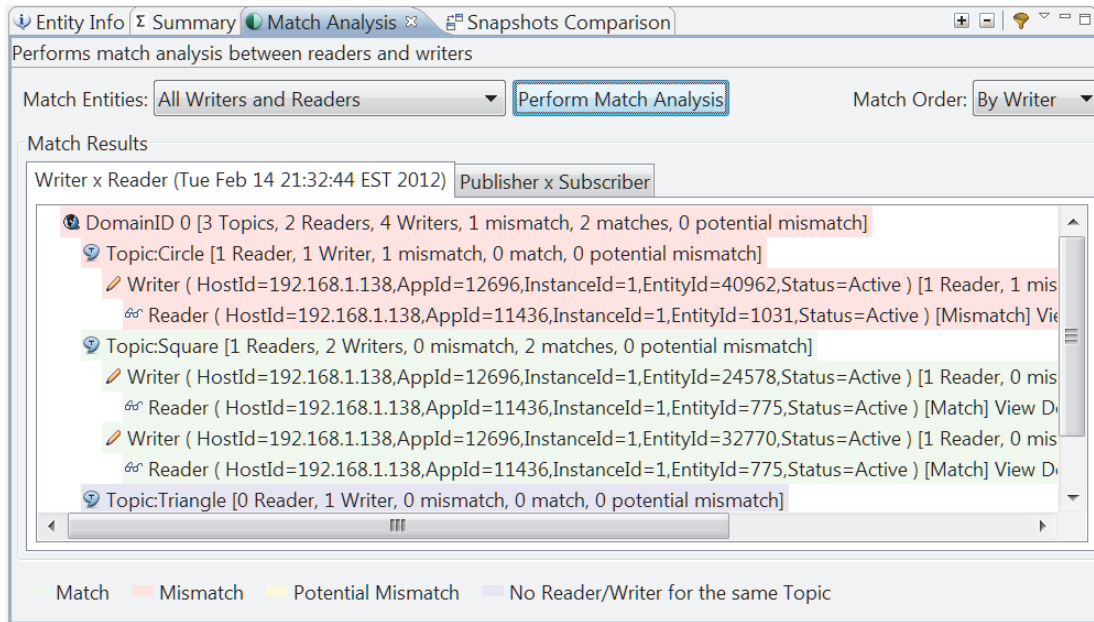
- ❑ Matches (green by default)
- ❑ Mismatches (that is, incompatible QoS) (pink by default)
- ❑ Potential Mismatches (meaning *Analyzer* does not have enough information to determine if there is a match or not) (yellow by default)
- ❑ Readers without Writers (and vice-versa) (light purple by default)

You can change the colors used for each category simply by clicking the colored boxes in the legend at the bottom of the tab.

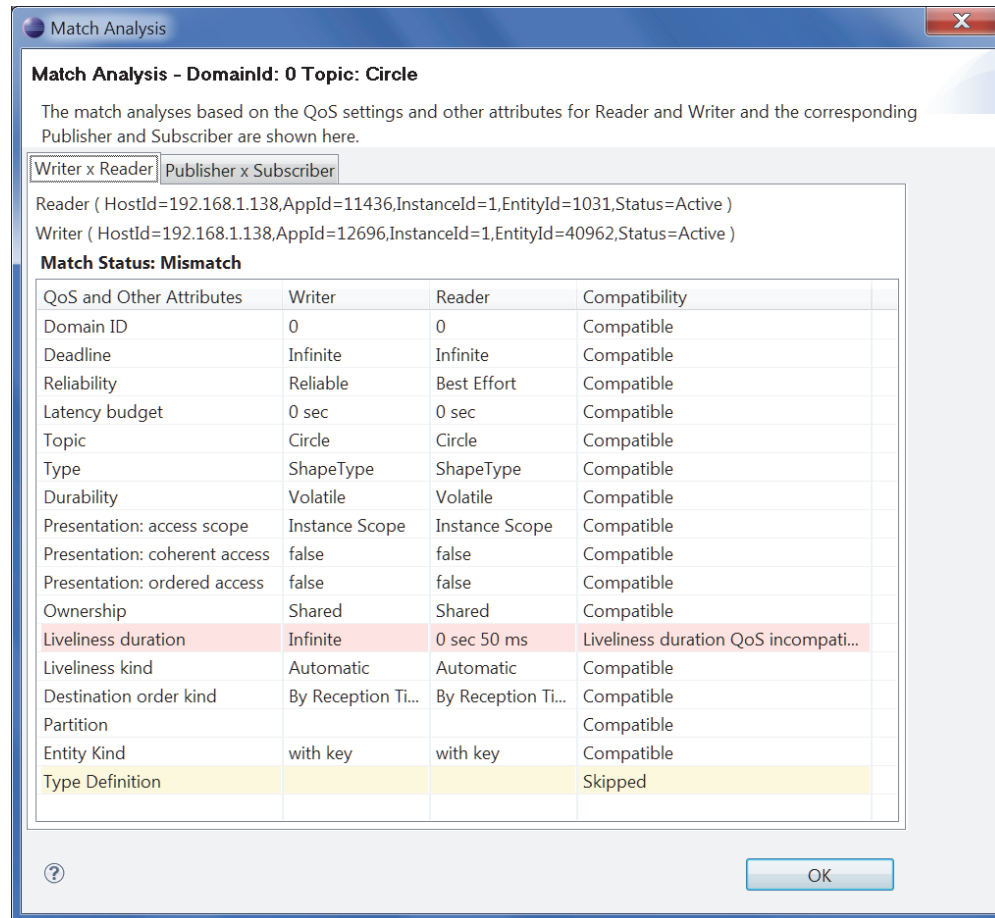
Let's see how these two issues show up in *Analyzer's* Match Analysis view.

1. Select the **Match Analysis** tab.
2. Click the **Perform Match Analysis** button. (If the button is disabled, make sure that the **Match Entities** list box has **All Writers and Readers** selected.)
3. Click the **Expand Tree** button  in the upper *right* corner. (This is *not* the same button in the Tree pane.)

The Match Results tree should look like this:



- Click the pink line for the Reader (under Topic Circle) to see the details:



The Compatibility column shows you which ‘matching’ criteria are okay and which ones are not. In this case, the Liveliness QoS settings are not compatible. Recall that we created this ‘mismatch’ on purpose to demonstrate how *Analyzer* helps you diagnose this type of problem. (In [Section 5.7](#), we will correct this error.)

- Click **OK** to close the details window.

5.5.1 Understanding Type Consistency

Analyzer includes partial support for the "Extensible and Dynamic Topic Types for DDS" specification from the Object Management Group (OMG). This section assumes that you are familiar with Extensible Types and you have read the *Core Libraries and Utilities Getting Started Guide Addendum for Extensible Types*.

Analyzer proactively checks for type consistency. This is done as new types are discovered from DataWriters and DataReaders during discovery. If there are type inconsistencies, they will be logged to the Message Log. You can also use the Match Analysis' Match Results panel to see them (provided the checkbox for "Skip Comparing Typecode in Match Analysis" is *not* selected in the Preferences).

The type consistency checking considers TypeObject information described in the "Extensible and Dynamic Topic Types for DDS" specification from the Object Management Group (OMG)¹. If you have the *Connexx Core Libraries and Utilities*, you can learn more in the *Core Libraries and Utilities Getting Started Guide Addendum for Extensible Types*².

How Analyzer determines type consistency:

1. If the TypeObject information is missing for either the DataWriter or DataReader:
 - a. If the registered type names are different, then they are inconsistent.
 - b. If the registered type names are equal:
 - If the TypeCode information is missing for either the DataWriter or DataReader, or if the TypeCodes are equal, they are consistent.
 - Otherwise, they are inconsistent.
2. If the TypeObject information is present for both the DataWriter and DataReader:
 - a. If the DataReader's TypeConsistencyEnforcementQosPolicy's **kind** is set to DISALLOW_TYPE_COERCION:
 - If the types are not structurally identical, then they are inconsistent.
 - Otherwise, they are consistent.
 - b. If the DataReader's TypeConsistencyEnforcementQosPolicy **kind** is set to ALLOW_TYPE_COERCION:

1. The OMG specification is available here: <http://www.omg.org/spec/DDS-XTypes/>.

2. <Connexx installation directory>/ndds.<version>/doc/pdf/
RTI_CoreLibrariesAndUtilities_GettingStarted_ExtensibleTypesAddendum.pdf)

- If the DataReader's type is not assignable from the DataWriter's type, then they are inconsistent. For additional information on assignability, see the *Core Libraries and Utilities Getting Started Guide Addendum for Extensible Types*².
- Otherwise, they are consistent.

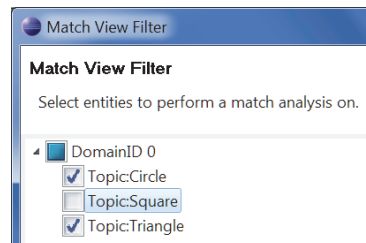
5.6 Filtering the Match Results Tree

The Match Results tree can get rather large if you have several Writers and Readers. There are two ways to scale down the number of objects included in the Match Results tree. You can filter out Topics that you are not interested in, or you can limit the analysis to selected objects. The following steps allow you to practice both ways.

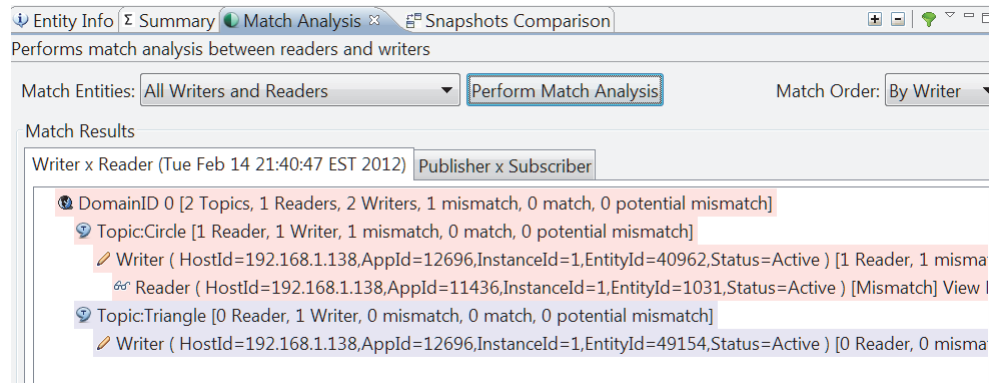
5.6.1 Filtering the Contents of the Match Analysis Tree

Since we know the Square Topic is working correctly, let's filter it out for now.

1. Select the **Filter** button  in the upper *right* corner. (This is *not* the same button in the Tree pane.)
2. Clear the checkbox for Topic Square.
3. Click **OK**.
4. Click the **Perform Match Analysis** button.
5. Select the **Expand Tree** button  on the right.



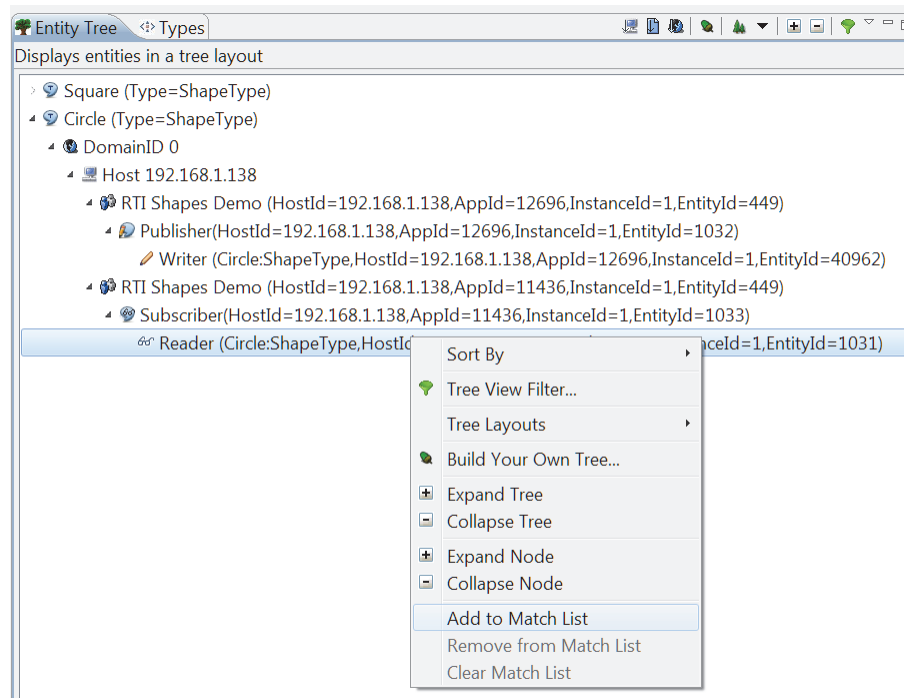
The Match Results tree should now look like this:



5.6.2 Analyzing Selected Objects Only

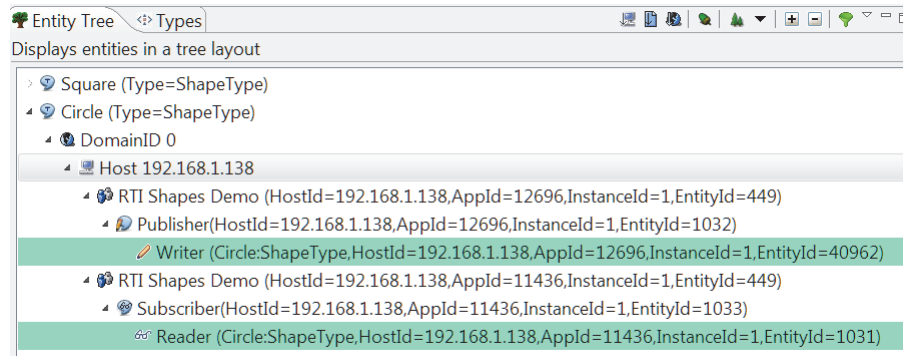
Let's perform a Match Analysis on just the Writer and Reader for the Topic Circle:

1. In the Entity Tree, right-click the Reader for Topic Circle. From the pop-up menu, select **Add to Match List**.



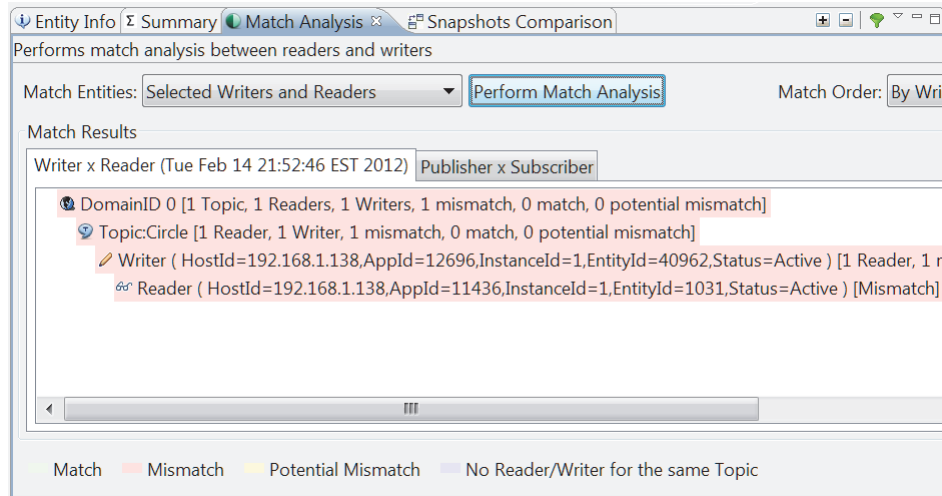
If you'd like to change the color used for highlighting entities in the match list, click the colored **Match Entities** box at the bottom.

- Do the same thing to add the Topic Circle's Writer to the match list. Now you should see that both entities are highlighted to show they are in the match list.



- In the **Match Analysis** view's **Match Entities** field, use the drop-down list to choose **Selected Writers and Readers**.
- Click the **Perform Match Analysis** button.
- Select the Expand Tree button  on the right.

The Match Results tree should now look like this:



Important Step!

- In the next section we will correct the incompatible QoS. But before we make any changes, **take a snapshot of the current system**. A snapshot saves the object

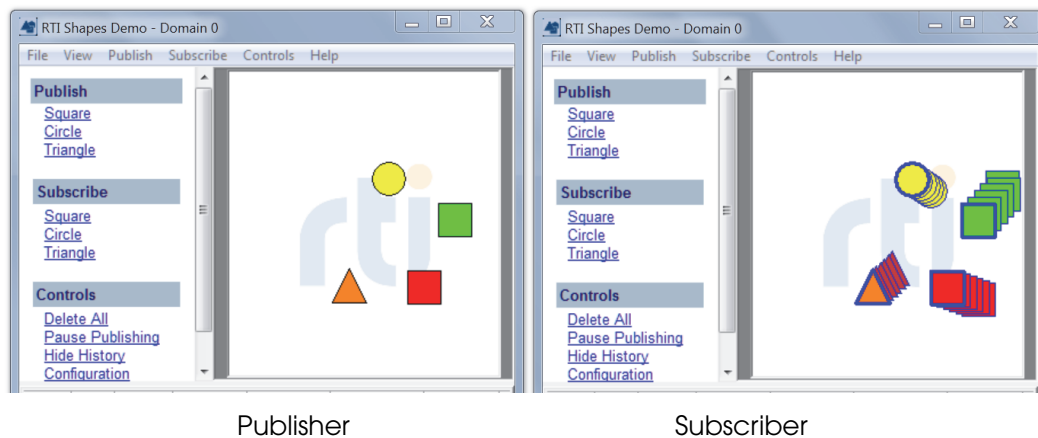
information. Later (in [Section 5.8](#)), we will take another snapshot and show you how to compare them.

Take a snapshot of our current configuration by selecting **Snapshots, Save Snapshot As...** and selecting a filename of your choice, such as **mynsnapshotA**.

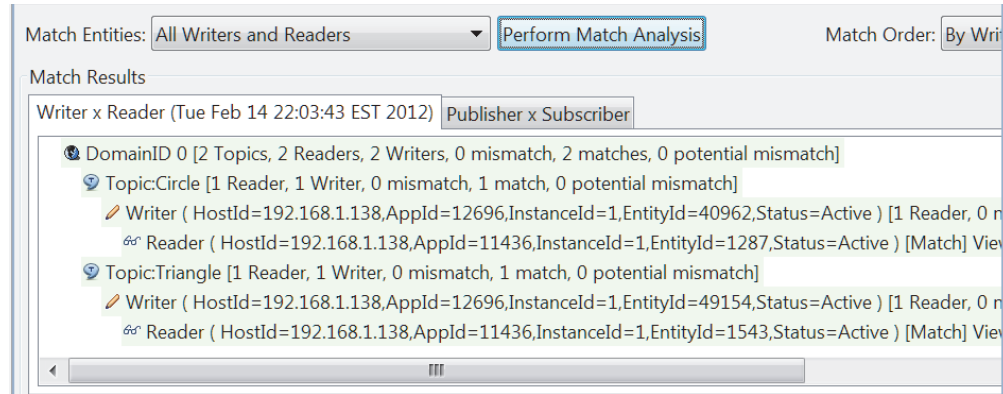
5.7 Correcting the Circle's Liveliness QoS

Now that we've seen how *Analyzer* can help you find incompatible QoS settings, let's correct the problem.

1. Open the *Shapes Demo* window for the Subscriber Demo. (This window's shapes have dark blue borders.)
2. In the Legend at the bottom, select the row for the Circle.
3. Press the **Delete** key on your keyboard.
4. Recreate the Reader for Circles: under **Subscribers**, select **Circle** and click **OK**. (So this time, we will use the default Liveliness.)
5. While we're correcting problems, create a Reader for Triangles: under **Subscribers**, select **Triangle** and click **OK**. (Although this isn't part of correcting the Circles' problem, it will make the snapshot comparison in the next section more interesting.)
6. In the Subscriber Demo window, notice that it is now receiving data for Squares, the now-compatible Circle, and Triangles, too.



7. In *Analyzer's* Match Analysis view, change the **Match Entities** list box to **All Writers and Readers**. Then press **Perform Match Analysis**. You will see the mismatch is gone. (Squares are not displayed because we filtered them out in [Section 5.6](#)).



Congratulations, you have diagnosed and corrected the communications problem!

5.8 Saving and Comparing Snapshots

In this section, we'll look at *Analyzer's* **Snapshots Comparison** feature.

A snapshot saves the object information about an attached application. You can compare one saved snapshot to the current "live" data, or compare two saved snapshots to each other. This feature can help you understand why one configuration behaves differently from another.

1. Recall that we took a snapshot of our original configuration just before we corrected the QoS of the Reader for Circles in [Step 6 on page 5-19](#).
2. Select the **Snapshots Comparison** view.
3. Under Snapshot 1, select the checkbox for Live Data.
4. For Snapshot 2, use the two **Browse...** buttons to select your snapshot file.
5. Ensure that the two checkboxes under **Options** are selected (**System being analyzed...** and **Exclude disposed entities...**).

6. Click the **Compare Snapshots** button and select the **Differences** tab. The **Entity Counts** tab will show that the current system (live data) has one more Reader than the snapshot:

Differences Snapshots			
Entity Counts Detailed Differences			
Entity	Live Data (Tue Feb 14 22:08:22...	mysnapshotA.xml (Tue Feb 14 ...	
Number of DomainIds	1	1	
Number of Hosts	1	1	
Number of Applications	2	2	
Number of Participants	2	2	
Number of Publishers	1	1	
Number of Subscribers	1	1	
Number of Writers	4	4	
Number of Readers	3	2	
Number of Topics	3	3	
Number of Types	1	1	

7. Click the **Detailed Differences** tab.

Differences Snapshots			
Entity Counts Detailed Differences			
Entity	Live Data (Tue Feb 14 22:08:22 EST 2012)	mysnapshotA.xml (Tue Feb 14 21:56	
Topic:Triangle	Topic Triangle (Type=ShapeType)	Topic Triangle (Type=ShapeType)	
Subscriber(EntityId=1033)	Subscriber (HostId=192.168.1.138,AppI...	Subscriber (HostId=192.168.1.138,A	
Reader(Circle:ShapeType, E Reader (HostId=192.168.1.138,AppId=...	Does not exist		
Reader(Triangle:ShapeType, Reader (HostId=192.168.1.138,AppId=...	Does not exist		
Reader(Circle:ShapeType, E Does not exist	Reader (HostId=192.168.1.138,AppI		

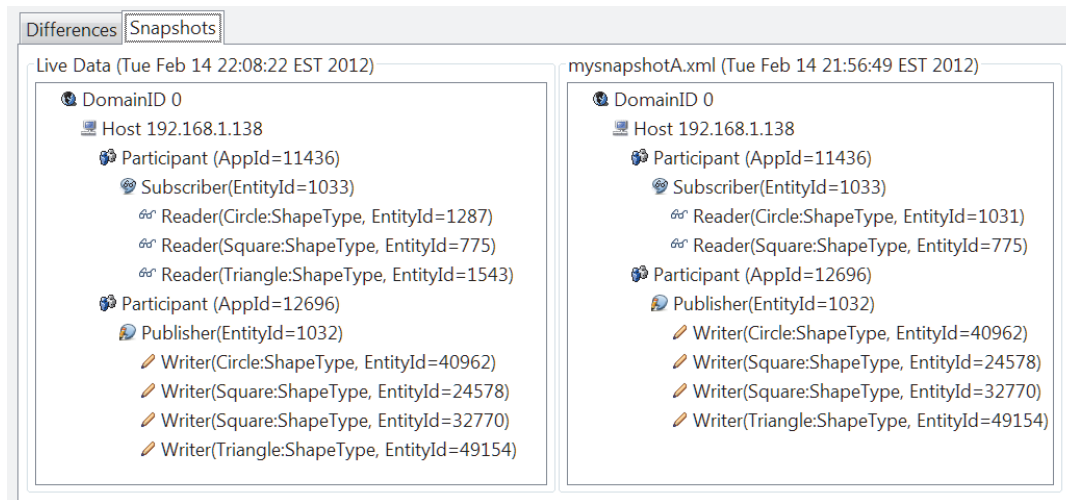
Here, *Analyzer* points out more details. Keep in mind that we deleted a Circle in the subscriber, then added a new Circle and a Triangle.

Lines 1-2 (pink): The Topic Triangle and its Subscriber exist in both the live data and the snapshot, but something about them has changed. (Recall that we deleted and added them back.)

Lines 3-4 (yellow): Two Readers were added after the snapshot was taken.

Line 5 (blue): The original Reader for Circles has been deleted.

8. Click the **Snapshots** tab to see a side-by-side comparison of the live-data and snapshot entity trees.



This completes the *Analyzer* tutorial. You can gain further experience by analyzing the applications described in the *RTI Core Libraries and Utilities Getting Started Guide* or one of your own *Connex* applications.

