RTI Federation Service

Getting Started Guide

Version 4.5

Your systems. Working as one.

Printed in U.S.A. First printing.
March 2012.

Trademarks

Copy and Use Restrictions

Technical Support

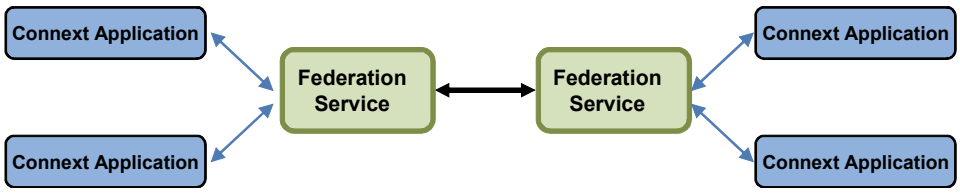
Real-Time Innovations, Inc.
232 E. Java Drive
Sunnyvale, CA 94089
Phone: (408) 990-7444
Email: support@rti.com
Website: https://support.rti.com/

Contents

Chapter 1 Welcome to RTI Federation Service

Welcome to *RTI® Federation Service*. This component of *RTI Connext™ Messaging* is an out- of-the-box solution for federating geographically dispersed systems. It scales *RTI Connext* (formerly, *RTI Data Distribution Service*) applications across domains, LANs and WANs, including firewall and NAT traversal.

Traditionally, *Connext* applications can only communicate with applications in the same domain. With *Federation Service*, you can send and receive data across domains. You can also control which data is sent by using allow and deny lists.



Simply set up *Federation Service* to pass data from one domain to another and specify any desired data filtering. No changes are required in the *Connext* applications.

Key benefits of *Federation Service*:

- **It can significantly reduce the time and effort spent integrating and scaling Connext applications across Wide Area Networks and Systems-of-Systems.**

  Many systems today already rely on *Connext* to distribute their information across a Local Area Network (LAN). However, more and more of these systems are being integrated in Wide Area Networks (WANs). With *Federation Service*, you can scale *Connext* real-time publish/subscribe data-distribution beyond the current local networks and make it available throughout a WAN.

- **With *Federation Service*, you can build modular systems out of existing systems.** Data can be contained in private domains within subsystems and you can designate that only certain "global topics" can be seen across domains. The same mechanism controls the scope of discovery. Both application-level and discovery traffic can be scoped, facilitating scalable designs.
- ***Federation Service* provides secure deployment across multiple sites.** You can partition networks and protect them with firewalls and NATS and precisely control the flow of data between the network segments.
- ***Federation Service* provides features for development, integration and testing.** Multiple sites can each locally test and integrate their core application, expose selected topics of data, and accept data from remote sites to test integration connectivity, topic compatibility and specific use-cases.
- **It connects remotely to live, deployed systems** so you can perform live data analytics, fault condition analysis, and data verification.

1.1 Available Documentation

Federation Service documentation includes:

- **Getting Started Guide** (RTI_Federation_Service_GettingStarted.pdf)—Highlights the benefits of *Federation Service*. It provides installation and startup instructions, and walks you through several examples so you can quickly see the benefits of using *Federation Service*.
- **Release Notes** (RTI_Federation_Service_ReleaseNotes.pdf)—Describes system requirements and compatibility, as well as any version-specific changes and known issues.
- **User's Manual** (RTI_Federation_Service_UsersManual.pdf)—Describes how to configure *Federation Service* and use it remotely.

Chapter 2 Installing Federation Service

This chapter describes:

- Installing on a UNIX-Based System (Section 2.1)
- Installing on a Windows System (Section 2.2)
- Installing the License File (Section 2.3)
- Uninstalling (Section 2.4)

For a list of supported system architectures, please see the *Release Notes*.

2.1 Installing on a UNIX-Based System

The distribution is packaged in a **.tar.gz** file. Unpack it as described below. You do not need to be logged in as root during installation.

1. Make sure you have GNU's version of the **tar** utility (which handles long file names). On Linux systems, this is the default **tar** executable. On Solaris systems, use **gtar**.
2. Create a directory for *Federation Service*. We will assume that you want to install under **/opt/rti/** (you may replace references to **/opt/rti/** with the directory of your choice).
3. Move the downloaded file into your newly created directory. In these instructions, we assume your distribution file is named **RTI_Federation_Service-<*version*>-i86Linux2.6gcc4.1.2.tar.gz**. Your filename will be different depending on your version and architecture.

4. Extract the distribution from the compressed files. For example:

   ```
   gunzip RTI_Federation_Service-4.5x-i86Linux2.6gcc4.1.2.tar.gz
   gtar xvf RTI_Federation_Service-4.5x-i86Linux2.6gcc4.1.2.tar
   ```

   The names of these files will differ based on the name of your version and target platform.

   Using our example path, you will end up with **/opt/rti/RTI_Federation_Service_4.5x**.

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

2.2 Installing on a Windows System

The distribution is packaged in a **.zip** file. Unpack it as described below. Depending on your version of Windows and where you want to expand these files, your user account may or may not require administrator privileges.

1. Create a directory for *Federation Service*. We will assume that you want to install under **C:\Program Files\RTI** (you may replace references to **C:\Program Files\RTI** with the directory of your choice).
2. Move the downloaded files into your newly created directory.
3. Extract the distribution from the compressed files. You will need a zip file utility such as WinZip® to help you.

   Using our example path, you will end up with **C:\Program Files\RTI\RTI_Federation_Service_4.5x**.

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

2.3 Installing the License File

If your *Federation Service* distribution requires a license file, you will receive one via email after you download the software.

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

1. The file specified with the **-licenseFile** option when you start *Federation Service* from the command-line.
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 you have any questions about license installation, please contact **support@rti.com**.

2.4 Uninstalling

To uninstall *Federation Service*, simply remove the directory where you installed the files.

Chapter 3 Running Federation Service

This chapter describes:

- ❏ Starting Federation Service (Section 3.1)
- ❏ Stopping Federation Service (Section 3.2)

3.1 Starting Federation Service

Federation Service runs as a separate application. The script to run the executable is located in **<*Federation Service installation directory*>/scripts**.

To start Federation Service, enter:

```
cd <installation directory for Federation Service>
scripts/rtifederationservice [options]
```

Example:

```
cd <installation directory for Federation Service>
scripts/rtifederationservice \
    -cfgFile example/shapes/topic_bridge.xml -cfgName example
```

Table 3.1 describes the command-line options.

3.2 Stopping Federation Service

To stop *Federation Service*, press **Ctrl-c**. *Federation Service* will perform a clean shutdown.

Table 3.1 **Federation Service Command-line Options**

Option	Description
`-appName <name>`	Assigns a name to the execution of *Federation Service*. Remote commands and status information will refer to the service using this name. See the *Federation Service User's Manual* for more information. In addition, the name of *DomainParticipants* created by *Federation Service* will be based on this name. Default: The name given with **-cfgName** if present, otherwise it is "RTI_Federation_Service".
`-cfgName`	This name is used to find the matching **<federation_service>** tag in the configuration file. It is required unless both **-remoteAdministrationDomainId** and **-noAutoStart** are used, in which case no configuration is loaded at start-up and *Federation Service* will simply wait to be configured remotely.
`-domainIdBase <ID>`	Sets the base domain ID. This value is added to the domain IDs in the configuration file. For example, if you set **-domainIdBase** to 50 and use domain IDs 0 and 1 in the configuration file, then the *Federation Service* will use domains 50 and 51. Default: 0
`-help`	Displays help information.
`-licenseFile <file>`	Specifies the license file (path and filename). Only applicable to licensed versions of *Federation Service*. If not specified, *Federation Service* looks for the license as described in Installing the License File (Section 2.3) in the Getting Started Guide.
`-identifyExecution`	Appends the host name and process ID to the service name provided with the **-appName** option. This helps ensure unique names for remote administration and monitoring. For example: MyFederationService_myhost_20024
`-maxObjectsPerThread <int>`	Parameter for the DomainParticipantFactory.
`-noAutoEnable`	Starts *Federation Service* in a disabled state. Use this option if you plan to enable *Federation Service* remotely, as described in the *Federation Service User's Manual*. This option overwrites the value of the **enable** attribute in the **<federation_service>** tag.

Table 3.1 **Federation Service Command-line Options**

Option	Description
`-remoteAdministrationDomainId <ID>`	Enables remote administration and sets the domain ID for remote communication. When remote administration is enabled, *Federation Service* will create a DomainParticipant, Publisher, Subscriber, DataWriter, and DataReader in the designated domain. The QoS values for these entities are described in the *Federation Service User's Manual*. This option overwrites the value of the tag **<domain_id>** within a **<administration>** tag. (See the *Federation Service User's Manual* for information on configuring remote access). Default: remote administration is not enabled unless it is enabled from the XML file.
`-remoteMonitoringDomainId <ID>`	Enables remote monitoring and sets the domain ID for status publication. When remote monitoring is enabled, *Federation Service* will create one *DomainParticipant*, one *Publisher*, five *DataWriters* for data publication (one for each kind of entity), and five *DataWriters* for status publication (one for each kind of entity). The QoS values for these entities are described in the *Federation Service User's Manual*. This option overwrites the value of the tag **<domain_id>** within a **<monitoring>** tag. (See the *Federation Service User's Manual* for information on configuring remote monitoring). Default: remote monitoring is not enabled unless it is enabled from the XML file.
`-sleep <seconds>`	Sleeps for x seconds before executing the next command. This is useful when issuing commands from a command file (see **-cmdName**).
`-stopAfter <sec>`	Stops the service after the specified number of seconds.
`-use42eAlignment`	Enables compatibility with *RTI Data Distribution Service* 4.2e. This option should be used when compatibility with 4.2e is required and the topic data types contain double, long long, unsigned long long, or long double members. Default: disabled

Table 3.1 **Federation Service Command-line Options**

Option	Description
`-verbosity <n>`	Controls what type of messages are logged: 0 - Silent 1 - Exceptions (Core Libraries and *Federation Service*) (default) 2 - Warnings(*Federation Service*) 3 - Information (*Federation Service*) 4 - Warnings (Core Libraries and *Federation Service*) 5 - Tracing (*Federation Service*) 6 - Tracing (Core Libraries and *Federation Service*) Each verbosity level, *n*, includes all the verbosity levels smaller than *n*.
`-version`	Prints the *Federation Service* version number.

Chapter 4 Using the Examples

This chapter describes several examples, all of which use *RTI Shapes Demo* to publish and subscribe to topics which are colored moving shapes (squares, circles, triangles):

- Example 1 - Routing All Data from One Domain to Another Using an Auto Topic Route (Section 4.1)
- Example 2 - Routing One Topic from One Domain to Another Using a Topic Route (Section 4.2)
- Example 3 - Using Remote Administration (Section 4.3)
- Example 4 - Monitoring (Section 4.4)
- Example 5 - Using TCP Transport with Federation Service (Section 4.5)

In each example, you can start all the applications on the same computer or on different computers in your network.

If you don't have *Shapes Demo* installed already, you should download and install it from RTI's Downloads page (www.rti.com/downloads) or the RTI Support Portal, accessible from https://support.rti.com/ (the portal requires an account name and password).

If you are not already familiar with how to start *Shapes Demo* and change its domain ID, please see the *Shapes Demo User's Manual* for details.

Note: If you run *Shapes Demo* and *Federation Service* on different machines and these machines do not communicate over multicast, you will have to set the environment variable NDDS_DISCOVERY_PEERS to enable communication. For example, assume that you run *Federation Service* on Host 1 and *Shapes Demo* on Host 2 and Host 3. In this case, the environment variable would be set as follows:

Host 1: `set NDDS_DISCOVERY_PEERS= <host2>, <host3>` (on Windows systems)

`setenv NDDS_DISCOVERY_PEERS <host2>, <host3>` (on UNIX-based systems)

Host 2: `set NDDS_DISCOVERY_PEERS=<host1>`

Host 3: `set NDDS_DISCOVERY_PEERS=<host1>`

4.1 Example 1 - Routing All Data from One Domain to Another Using an Auto Topic Route

This example uses the default configuration file[1] for *Federation Service*, which routes all data published on domain 0 to subscribers on domain 1.

1. Start *Shapes Demo* on domain 0 (the default domain). We'll call this the Publishing Demo.
2. Start a second copy of *Shapes Demo* on domain 1. We'll call this the Subscribing Demo.
3. In the Publishing Demo, publish some Squares, Circles, and Triangles.
4. In the Subscribing Demo, subscribe to Squares, Circles and Triangles.

   Notice that the Subscribing Demo does *not* receive any shapes. Since we haven't started *Federation Service* yet, data from domain 0 isn't routed to domain 1.
5. Start *Federation Service* by entering the following in a command shell:

   ```
   cd <Federation Service installation directory>
   scripts/rtifederationservice -cfgName default
   ```

   Now you should see all the shapes in the Subscribing Demo.
6. Stop *Federation Service* by pressing **Ctrl-c**.

   You should see that the Subscribing Demo stops receiving shapes.

Additionally, you can start *Federation Service* (Step 5) with the following parameters:

- **-verbosity 3**, to see messages from *Federation Service*, including events that have triggered the creation of routes.
- **-domainIdBase X**, to use domains **X** and **X+1** instead of 0 and 1 (in this case, you need to change the domain IDs used by *Shapes Demo* accordingly). This option adds X to the domain IDs in the configuration file.

1. ***<installation directory for RTI Federation Service>*/resource/xml/RTI_FEDERATION_SERVICE.xml**

4.2 Example 2 - Routing One Topic from One Domain to Another Using a Topic Route

In this example, the federation service receives samples of topic Square from one domain and republishes them in a different domain.

1. Start *Shapes Demo* on domain 0 (the default domain). We'll call this the Publishing Demo.
2. Start a second copy of *Shapes Demo* on domain 1. We'll call this the Subscribing Demo.
3. Start *Federation Service* by entering the following in a command shell:

```
cd <Federation Service installation directory>
scripts/rtifederationservice \
    -cfgFile example/shapes/topic_bridge.xml -cfgName example
```

4. In the Publishing Demo (domain 0), publish some Squares, Circles and Triangles.
5. In the Subscribing Demo (domain 1), subscribe to Squares, Circles and Triangles.

   You will see that all the squares (and only squares) from domain 0 are republished as on domain 1.
6. Stop *Federation Service* by pressing **Ctrl-c**.
7. To see how this example is configured, review the contents of **example/shapes/topic_bridge.xml**.
8. Try writing your own topic route that republishes triangles on domain 0 to circles on domain 1. Create some Triangle publishers and a Circle subscriber on the respective *Shapes Demo* windows.

4.3 Example 3 - Using Remote Administration

In this example, we will configure *Federation Service* remotely. We won't see data being routed until we remotely enable an auto topic route after the application is started. Then we will change a QoS value and see that it takes effect on the fly.

1. Start *Shapes Demo* on domain 0 (the default domain). We'll call this the Publishing Demo.
2. Start a second copy of *Shapes Demo* on domain 1. We'll call this the Subscribing Demo.
3. Start *Federation Service* by entering the following in a command shell:

   ```
   cd <Federation Service installation directory>
   ```

   ```
   scripts/rtifederationservice \
     -cfgFile example/shapes/administration.xml \
     -appName MyFederationService -cfgName example
   ```

4. In the Publishing Demo (domain 0), publish some Squares, Circles and Triangles.
5. In the Subscribing Demo (domain 1), subscribe to Squares, Circles and Triangles.

   Notice that no data is routed to domain 1.
6. On a different or the same machine, start the *Federation Service* shell:

   ```
   cd <Federation Service installation directory>
   ```

   ```
   scripts/rtifssh -domainId 0
   ```

   Notes:

   - We use domain 0 in the shell because *Federation Service* is configured in **administration.xml** to receive remote commands on that domain. You could have started *Federation Service* with the **-remoteAdministrationDomainId X** command-line option and then used domain **X** for the shell.
7. In the shell, enter the following command:

   ```
   > enable MyFederationService RemoteConfigExample::Session::Shapes
   ```

   Notice that the shapes are now received on domain 1. The above command consists of two parts: the name of the federation service, which you gave when you launched the application with the option **-appName**, and the name of the entity you wanted to enable. That name is formed by appending its parent entities' names starting from the domain route as defined in the configuration file **administration.xml**.

   Note that you could have run *Federation Service* without **-appName**. The name would have been the one provided with **-cfgName** (“example”). You could also have used **-identifyExecution** to generate the name based on the host and application ID. In this case, you would have used this automatic name in the shell.

8. Examine the file **example/shapes/time_filter_qos.xml** on the federation service machine. It contains an XML snippet that defines a QoS value for an auto topic route's *DataReader*. Execute the following command in the shell:

```
> update MyFederationService RemoteConfigExample::Session::Shapes
    example/shapes/time_filter_qos.xml
```

Note: The path to the XML file in this example is relative to the working directory from which you run *Federation Service*

Notice that the receiving application only gets shapes every 2 seconds. The auto topic route has been configured to read (and forward) samples with a minimum separation of 2 seconds.

Federation Service can be configured remotely using files located on the federation service machine or the shell machine. In step 9 you will edit the configuration files on both machines. Step 10 shows how to specify which of the two configuration files you want to use. If you are running the shell and *Federation Service* on the same machine, skip steps 9 and 10.

9. Edit the XML configuration files on both machines:

   a. In **example/shapes/time_filter_qos.xml** on the federation service machine, change the minimum separation to 0 seconds.

   b. In **example/shapes/time_filter_qos.xml** on the shell machine, change the minimum separation to 5 seconds.

10. Run the following commands in the shell:

    a. Enter the following command. Notice the use of **remote** at the end—this means you want to use the XML file on the federation service machine (the *remote* machine, which is the default if nothing specified).

```
> update MyFederationService RemoteConfigExample::Session::Shapes
    example/shapes/time_filter_qos.xml remote
```

Note: The path to the XML file in this example is relative to the working directory from which you run *Federation Service*.

Since no time filter applies, the shapes are received as they are published.

    b. Enter the following command. This time we use **local** at the end—this means you want to use the XML file on the shell machine (the *local* machine).

```
> update MyFederationService
      RemoteConfigExample::Session::Shapes
      example/shapes/time_filter_qos.xml local
```

Note: The path to the XML file in this example is relative to the working directory from which you run the *Federation Service* shell.

You will see that now the shapes are only received every 5 seconds.

c. Enter the following command. Once again, we use **remote** at the end to switch back to the XML file on the federation service machine.

```
> update MyFederationService RemoteConfigExample::Session::Shapes
    example/shapes/time_filter_qos.xml remote
```

Shapes are once again received as they are published.

11. Disable the auto topic route again by entering:

```
> disable MyFederationService RemoteConfigExample::Session::Shapes
```

The shapes are no longer received on Domain 1.

Note: At this point, you could still update the auto topic route's configuration. You could also change immutable QoS values, since the DataWriter and DataReader haven't been created yet. These changes would take effect the next time you called enable.

12. Run these commands in the shell and see what happens after each one:

```
> enable MyFederationService
    RemoteConfigExample::Session::SquaresToCircles
> disable MyFederationService
    RemoteConfigExample::Session::SquaresToCircles
> enable MyFederationService
    RemoteConfigExample::Session::SquaresToTriangles
```

These commands change the output topic that is published after receiving the input Square topic. As you can see, you can use the shell to switch topic routes after *Federation Service* has been started.

13. Stop the shell by running this command in the shell:

```
> exit
```

14. Stop *Federation Service* by pressing **Ctrl-c**.

4.4 Example 4 - Monitoring

With *Federation Service* you can publish status information. The monitoring configuration is quite flexible and allows selecting the entities that you want to monitor and how often they should publish their status.

1. Start *Shapes Demo* on domain 0 (the default domain). We'll call this the Publishing Demo.
2. Start a second copy of *Shapes Demo* on domain 1. We'll call this the Subscribing Demo.
3. In the Publishing Demo (domain 0), publish two Squares, two Circles, and two Triangles.
4. In the Subscribing Demo (domain 1), subscribe to Squares, Circles and Triangles.
5. Start *Federation Service* by entering the following in a command shell:

   ```
   cd <Federation Service installation directory>
   scripts/rtifederationservice \
     -cfgFile example/shapes/monitoring.xml -cfgName example \
     -appName MyFederationService
   ```

   This configuration file routes Squares and Circles using two different topic routes.
6. Now you can subscribe to the monitoring topics (see Chapter 5 in the *Federation Service User's Manual* for more information). You can do it in your own application, or using an RTI utility such as *RTI Spreadsheet Add-in for Microsoft Excel* or *rtiddsspy*. We will use *rtiddsspy*. If you have the RTI Core Libraries and Utilities installed, run this command:

   ```
   rtiddsspy -domainId 2 -printSample
   ```

   rtiddsspy, provided with the RTI Core Libraries and Utilities, monitors publications on any domain.

   If you have *Spreadsheet Add-in* for Microsoft Excel, you can visualize the monitoring information in Microsoft Excel on Windows systems. In that case, open **example/shapes/monitoring_visualization.xls**.

   Note: We use domain 2 in *rtiddsspy* because *Federation Service* is configured in **monitoring.xml** to publish status information on that domain. You could have started *Federation Service* with the **-remoteMonitoringDomainId X** command-line option and then used domain **X** for *rtiddsspy*.

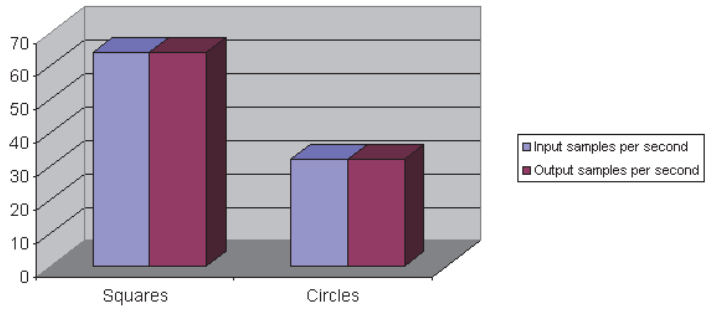
7. Depending on the publication period of the entity in the XML file we used, you will receive status samples at different rates. In *rtiddsspy*, check the statistics about the two topic routes we are using. We will focus on the input samples per second:

```
federation_service_name: "MyFederationService"
domain_route_name: "DomainRoute"
session_name: "Session"
name: "Squares"
input_samples_per_s:
   publication_period_metrics:
      period_ms: 2000
      count: 64
      mean: 31.904287
      minimum: 31.904287
      maximum: 31.904287
      std_dev: 0.000000
   historical_metrics:

federation_service_name: "MyFederationService"
domain_route_name: "DomainRoute"
session_name: "Session"
name: "Circles"
input_samples_per_s:
   publication_period_metrics:
      period_ms: 5000
      count: 158
      mean: 31.530632
      minimum: 29.940121
      maximum: 31.968033
      std_dev: 0.795508
   historical_metrics:
```

The number of samples per second in our case is 32. That value depends on the publication rate of *Shapes Demo*, configurable with the option **-pubInterval <milliseconds between writes>**. In our example we started the Publishing Demo with **-pubInterval 64**.

Using *Spreadsheet Add-in* for Microsoft Excel, select the **Topic Route** worksheet (from the tabs at the bottom of Excel); you should be able to see the following bar chart, among other data and figures:



8. Create two additional Square publishers on the publishing demo (domain 0).

9. Check *rtiddsspy* again for new status information:

```
federation_service_name: "MyFederationService"
domain_route_name: "DomainRoute"
session_name: "Session"
name: "Squares"
input_samples_per_s:
   publication_period_metrics:
      period_ms: 2000
      count: 124
      mean: 61.876247
      minimum: 59.880241
      maximum: 63.872257
      std_dev: 1.996008
   historical_metrics:

federation_service_name: "MyFederationService"
domain_route_name: "DomainRoute"
session_name: "Session"
name: "Circles"
input_samples_per_s:
   publication_period_metrics:
      period_ms: 5000
      count: 158
      mean: 31.536926
      minimum: 29.940121
      maximum: 31.936129
      std_dev: 0.798403
   historical_metrics:
```

In the topic route Squares we are receiving double amount of data.

Also look at the status of the domain route:

```
federation_service_name: "MyFederationService"
domain_route_name: "DomainRoute"
name: "Session"
input_samples_per_s:
   publication_period_metrics:
      period_ms: 5000
      count: 474
      mean: 47.305389
      minimum: 29.940121
      maximum: 63.872257
      std_dev: 1.197605
   historical_metrics:
```

It contains an aggregation of the two contained topic routes, giving us a mean of nearly 48 samples per second.

We can update the monitoring configuration at run time using the remote administration feature. In the configuration file, we enabled remote administration on domain 0.

10. On a different or the same machine, start the *Federation Service* shell:

```
> cd <Federation Service installation directory>
> scripts/rtifssh -domainId 0
```

11. We are receiving the status of the topic route Circles every five seconds. To receive it more often, use the following command:

```
> update MyFederationService DomainRoute::Session::Circles
  topic_route.entity_monitoring.status_publication_period.sec=2
```

12. In some cases, you might want to know only about one specific topic route. If you only want to know about the topic route Circles but not Squares, you can disable monitoring for Squares:

```
> update MyFederationService DomainRoute::Session::Squares
  topic_route.entity_monitoring.enabled=false
```

13. To enable it again, enter:

```
> update MyFederationService DomainRoute::Session::Squares
  topic_route.entity_monitoring.enabled=true
```

14. If you are no longer interested in monitoring this service, you can completely disable it with the following command:

```
> update MyFederationService
    federation_service.monitoring.enabled=false
```

Now you won't receive any more status samples.

15. You can enable it again any time by entering:

```
> update MyFederationService federation_service.monitoring.enabled=true
```

16. Stop *rtiddsspy* by pressing **Ctrl-c.**

17. Stop the shell:

```
> exit
```

18. Stop *Federation Service* by pressing **Ctrl-c**.

4.5 Example 5 - Using TCP Transport with Federation Service

This example shows how to use *Federation Service* to bridge data between different LANs over TCP. *Federation Service* will act as the gateway in a LAN with which other Connext applications can communicate to send or receive data. Chapter 7 of the *Federation Service User's Manual* has more information about scenarios and detailed configuration parameters.

You will run two copies of *Federation Service*. One copy will run on a machine that is behind a firewall/router with a public IP (First Peer); the other will run on a machine in another LAN (Second Peer).

On the First Peer (behind a firewall/router with a public IP):

1. In the First Peer's network, configure the firewall to forward the TCP ports used by *Federation Service*.

   In this example, we will use port 7400.

   You do not need to configure your firewall for every single *Connext* application in your LAN; doing it just once for *Federation Service* will allow other applications to communicate through the firewall.

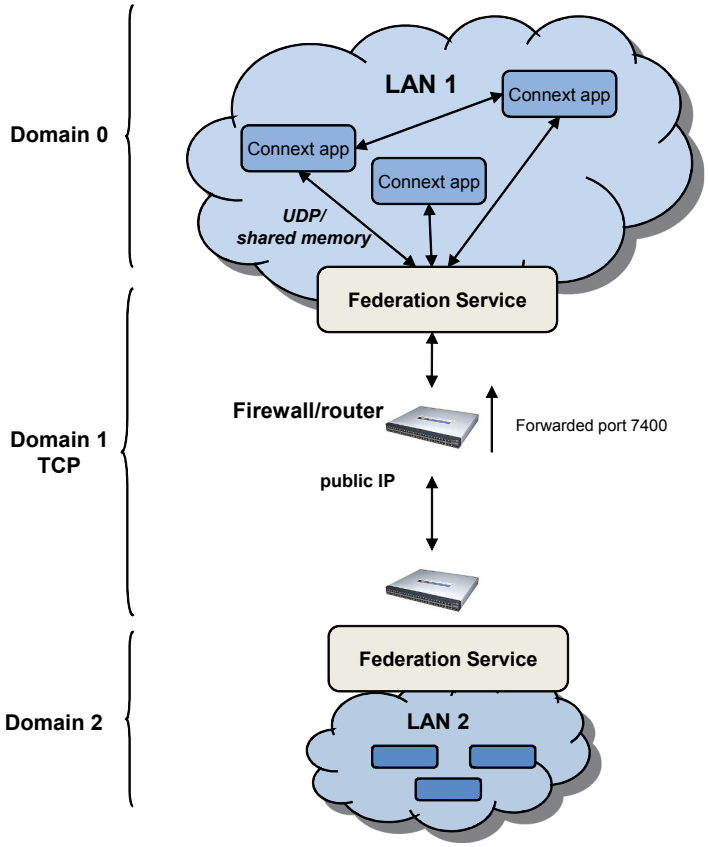
2. Include the Second Peer's public IP address and port in the **NDDS_DISCOVERY_PEERS** environment variable.

   For example, on a UNIX-based system:

   ```
   setenv NDDS_DISCOVERY_PEERS
        tcpv4_wan://<server's public IP address>:<port>
   ```

   On a Windows system:

   ```
   set NDDS_DISCOVERY_PEERS=
        tcpv4_wan://<server's public IP address>:<port>
   ```



When you configure **NDDS_DISCOVERY_PEERS**, make sure to use a transport class prefix (tcpv4_wan, udpv4, shmem) for each entry. (See Section 12.2 in the *RTI Core Libraries and Utilities User's Manual* for details on formatting addresses in **NDDS_DISCOVERY_PEERS**.)

For example:

```
setenv NDDS_DISCOVERY_PEERS tcpv4_wan://10.10.1.10:7400,\
    udpv4://192.168.0.1,udpv4://192.168.0.2,shmem://
```

3. Set the public IP address and port in the configuration file:

   a. Open the file **example/shapes/tcp_transport.xml**.

b. The file contains several federation service configurations. Find the federation service configuration, <federation_service name="TCP_1">. Then find the "public_address" property (<name>dds.transport.TCPv4.tcp1.public_address</name>) within that configuration.

c. Set the *local* public IP address and port. For example, to set the address to 10.10.1.150 and port 7400:

```
<element>
   <name>
     dds.transport.TCPv4.tcp1.public_address
   </name>
   <value>10.10.1.150:7400</value>
</element>
```

d. Save and close the file.

4. Run these commands and choose "**TCP_1**":

```
cd <Federation Service installation directory>

scripts/rtifederationservice \
    -cfgFile example/shapes/tcp_transport.xml \
    -cfgName TCP_1
```

5. On any computer in this LAN, start *Shapes Demo* and publish some shapes on domain 0.

On the Second Peer (a machine in any other LAN):

6. Include the First Peer's public IP address and port in the **NDDS_DISCOVERY_PEERS** environment variable the same way you did before.

7. Set the public IP address and port in the configuration file:

a. The file contains several federation service configurations. Find the federation service configuration, **<federation_service name="TCP_2">**. Then find the "public_address" property (**<name>dds.transport.TCPv4.tcp1.public_address</name>**) within that configuration.

b. Set the *local* public IP address and port. For example, to set the address to 10.10.1.10 and port 7400:

```
<element>
   <name>
     dds.transport.TCPv4.tcp1.public_address
   </name>
   <value>10.10.1.10:7400</value>
</element>
```

c. Save and close the file.

8. Run these commands and choose "**TCP_2**":

```
cd <Federation Service installation directory>

scripts/rtifederationservice \
    -cfgFile example/shapes/tcp_transport.xml -cfgName TCP_2
```

9. On any computer in this LAN, start *Shapes Demo* and create subscribers on domain 2. Do not use an already running instance of *Shapes Demo*—you need a new one that uses a different domain ID.

   You should receive what is being published in the server's LAN.

Notes:

❑ Running *Shapes Demo* on a Different Computer

If the computer running *Shapes Demo* is different than the computer running the client federation service, add the address of the client (IP address or host name) to the *Shapes Demo* discovery peers before starting the shapes demo. To do so, use the **–peer** command-line option or set the **NDDS_DISCOVERY_PEERS** environment variable.

❑ Using Two Computers in the Same LAN

If both machines are in the same LAN, run both federation services with the configuration file **tcp_transport_lan.xml** and use **"tcpv4_lan://"** as the peer prefix in the environment variable NDDS_DISCOVERY_PEERS. You don't need to specify an IP address in the configuration file.

❑ Running the Example on One Computer

To run the example on the same machine, open the file **example/shapes/tcp_transport_lan.xml** and change the property **dds.transport.TCPv4.tcp1.server_bind_port** within **TCP_1** to 7401. Run both federation services with the modified **tcp_transport_lan.xml** configuration file and use **"tcpv4_lan://"** as the peer prefix in the environment variable NDDS_DISCOVERY_PEERS. You will also need to specify port 7401 in the **tcpv4_lan** peer in the NDDS_DISCOVERY_PEERS environment variable of the federation service in the Second Peer to reflect this port change in the configuration file.

❑ Using a Secure Connection over WAN

To run this example, you need OpenSSL 0.9.8n (or higher) and *RTI Secure WAN Transport*. To purchase *RTI Secure WAN Transport*, contact your account representative or **sales@rti.com**. OpenSSL is available from the RTI's Downloads page

(www.rti.com/downloads), or you may obtain it from another source. Make sure the OpenSSL libraries' location is in your LD_LIBRARY_PATH (on UNIX-based systems) or Path (on Windows systems).

To run the example using a secure connection between the two router instances, use the configuration file **tcp_transport_tls.xml**. You will also need to set the peer prefix to "**tlsv4_wan://**" in the NDDS_DISCOVERY_PEERS environment variable. The **tcp_transport_tls.xml** file is based on **tcp_transport.xml** and uses a WAN configuration to establish communication.

❑ Using a Secure Connection over LAN

To run this example using a secure connection between two routers instances within the same LAN, you need OpenSSL 0.9.8n (or higher) and *RTI TLS Support*. To purchase *RTI TLS Support*, contact your account representative or **sales@rti.com**. OpenSSL is available from the RTI's Downloads page (www.rti.com/downloads), or you may obtain it from another source. Make sure the OpenSSL libraries' location is in your LD_LIBRARY_PATH (on UNIX-based systems) or Path (on Windows systems).

To use TLS encryption over a LAN configuration, you can use the file **tcp_transport_tls_lan.xml**. You will also need to set the peer prefix to "**tlsv4_lan://**" in the NDDS_DISCOVERY_PEERS environment variable. The **tcp_transport_tls_lan.xml** configuration file is based on **tcp_transport_lan.xml** and uses a LAN configuration to establish communication.