

RTI Recording Service

Getting Started Guide

Version 5.3.1



Your systems. Working as one.



© 2007-2018 Real-Time Innovations, Inc.
All rights reserved.
Printed in U.S.A. First printing.
February 2018.

Trademarks

Real-Time Innovations, RTI, NDDS, RTI Data Distribution Service, DataBus, Connex, Micro DDS, the RTI logo, IRTI and the phrase, “Your Systems. Working as one,” are registered trademarks, trademarks or service marks of Real-Time Innovations, Inc. All other trademarks belong to their respective owners.

Copy and Use Restrictions

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form (including electronic, mechanical, photocopy, and facsimile) without the prior written permission of Real-Time Innovations, Inc. The software described in this document is furnished under and subject to the RTI software license agreement. The software may be used or copied only under the terms of the license agreement.

Third-Party Copyright Notices

Portions of this product include software derived from Fmatch, (c) 1989, 1993, 1994 The Regents of the University of California. All rights reserved. The Regents and contributors provide this software "as is" without warranty.

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 Welcome to RTI Recording Service

- 1.1 Paths Mentioned in Documentation 1-2
- 1.2 Recording Service Installation Components 1-3

2 Tutorial—RTI Recording Console

- 2.1 Setting Up Shapes Demo 2-1
- 2.2 Recording Data 2-3
- 2.3 Replaying Data 2-4

3 Tutorial—Record and Replay Tools

- 3.1 Setting Up Shapes Demo 3-2
- 3.2 Configuring the Record Tool 3-3
- 3.3 Running the Record Tool 3-4
- 3.4 Viewing the Recorded Data with SQLite 3-5
- 3.5 Using Replay 3-5
 - 3.5.1 Recording Data for Replay 3-5
 - 3.5.2 Changing Replay’s Configuration File 3-6
 - 3.5.3 Replaying Data at Normal Speed 3-7
 - 3.5.4 Replaying Data at a Faster Rate and Using Remote Control 3-7
- 3.6 Managing the Record Tool Remotely 3-9
 - 3.6.1 Connecting to the Record Tool using Shared Memory 3-10

Chapter 1 Welcome to RTI Recording Service

RTI® *Recording Service* includes these components:

- **Record**, an *RTI Connex*® *DDS* application that records both *Connex DDS* discovery and topic data. All recorded data is stored in one or more SQL database files.
- **Replay**, a tool that can ‘play back’ the recorded data. You even have the option of replaying the data with different data rates or QoS settings.
- **Recording Console**, a simple graphical user interface (GUI) for using the *Record* and *Replay* tools. This interface significantly reduces *Recording Service* configuration time and complexity, and does not require any programming. The *Recording Console* makes it easy to use *Recording Service* for testing algorithms and other processing logic against pre-recorded test data, conducting regression testing from 'golden' data inputs, or recording live data from the field for post-mission analysis.
- **Convert**, a utility that enables serialized or deserialized data recorded with *Record* to be exported to CSV, HTML, SQL, or XML formats.

Recording Features:

- Records data from applications in multiple domains.
- Records entire Topics, or specific Topic fields, based on POSIX file-name matching expressions.
- Records all data types except bit-fields.
- Records to multiple files with configurable file-size limits. Optionally overwrites the oldest file when the maximum number of files has been reached.
- Records the DDS SampleInfo structure and a timestamp for both discovery data and user data.
- Records using either Best Effort or Reliable communications.
- Optionally records data from only specified partitions.
- Supports remote operation.

Replay Features

- Publishes data samples that were recorded in serialized format.
- Highly configurable—you can:
 - Choose which serialized topics to replay
 - Set the replay rate (faster or slower) or use the original rate
 - Change the QoS of the publications
 - Configure the QoS for the tool itself
 - Dynamically control the replay (start, stop, pause) and single-step through the data samples

1.1 Paths Mentioned in Documentation

The documentation refers to:

- **<NDDSHOME>**

This refers to the installation directory for *Connex DDS*.

The default installation paths are:

- Mac OS X systems:
/Applications/rti_connex_dds-version
- UNIX-based systems, non-*root* user:
/home/your user name/rti_connex_dds-version
- UNIX-based systems, *root* user:
/opt/rti_connex_dds-version
- Windows systems, user without Administrator privileges:
<your home directory>\rti_connex_dds-version
- Windows systems, user with Administrator privileges:
C:\Program Files\rti_connex_dds-version

You may also see \$NDDSHOME or %NDDSHOME%, which refers to an environment variable set to the installation path.

Wherever you see <NDDSHOME> used in a path, replace it with your installation path.

Note for Windows Users: When using a command prompt to enter a command that includes the path **C:\Program Files** (or any directory name that has a space), enclose the path in quotation marks. For example:

```
"C:\Program Files\rti_connex_dds-version\bin\rtiddsgen"
```

or if you have defined the NDDSHOME environment variable:

```
"%NDDSHOME%\bin\rtiddsgen"
```

- **<path to examples>**

Examples are copied into your home directory the first time you run *RTI Launcher* or any script in **<NDDSHOME>/bin**. This document refers to the location of these examples as **<path to examples>**.

Wherever you see <path to examples>, replace it with the appropriate path.

By default, the examples are copied here:

- Mac OS X systems:
/Users/your user name/rti_workspace/version/examples
- UNIX-based systems:
/home/your user name/rti_workspace/version/examples
- Windows systems:
<your home directory>\rti_workspace\version\examples

You can specify a different location for the **rti_workspace** directory. You can also specify that you do not want the examples copied to the workspace. See the *RTI Connex DDS Core Libraries Getting Started Guide*.

1.2 Recording Service Installation Components

Recording Service is installed automatically when you install one of *Connex DDS* packages (*Professional*, *Evaluation*, or *Basic*).

After installation, you will end up with the following directory structure, where `<NDDSHOME>` is where you installed the *Connex DDS* package (see [Paths Mentioned in Documentation \(Section 1.1\)](#)):

- `<NDDSHOME>`
 - `/bin`
Executable scripts for the applications; includes **rtireconv**, **rtirecord**, **rtirecsh**, **rtireplay**, **rtireplaysh** and **rtirecordingconsole**.
 - `/doc/manuals/recording_service`
Documentation for *Recording Service*, in PDF format.
 - `/resource/app/bin/<architecture>`
Binary executable files for the applications (**rtireconv**, **rtirecord**, **rtirecsh**, **rtireplay**, **rtireplaysh** and **RTI_recording_console**). These files are architecture-dependent; they are used by the scripts in the **bin** directory and provide the necessary environment settings. There is no need to use them directly.
 - `/resource/idl`
IDL files describing *Recorder* and *Replay*'s remote administration interfaces.
 - `/resource/schema`
Recorder and *Replay*'s XSD and DTD files for XML validation and configuration help.
 - `/resource/xml`
Recorder and *Replay* default XML configuration files (**RTI_RECORDING_SERVICE.xml** and **RTI_REPLAY_SERVICE.xml**). These files are picked up by *Recorder* and *Replay* automatically, so there is no need to specify them when using the configurations inside.
 - `/resource/app/app_support/recording_service/<architecture>`
This directory contains necessary configuration files and information for *Recording Console*.

As described in [Paths Mentioned in Documentation \(Section 1.1\)](#), examples are copied into your home directory the first time you run *RTI Launcher* or any script in `<NDDSHOME>/bin`. For *Recording Service*, the relevant files are here:

- `<path to examples>`
 - `/recording_service/record`
Example files and configurations for *Recorder*.
 - `/recording_service/replay`
Example files and configurations for *Replay*.
 - `/user_config/recording_service`

This directory contains user-modifiable files with configurations for *Recorder* and *Replay*, **USER_RECORDING_SERVICE.xml** and **USER_REPLAY_SERVICE.xml**. These files are picked up by *Recorder* and *Replay* automatically when they are present in the working directory. For example, *Launcher* always sets the working directory to this folder so that the files are detected automatically when running *Recorder* or *Replay*.

Chapter 2 Tutorial—RTI Recording Console

The best way to learn about *RTI Recording Console* is to start using it. This chapter will walk you through the steps to use the *Console* to record and replay data from *Shapes Demo*.

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

- *Recording Service*
- *Shapes Demo*

The tutorial walks you through the following exercises:

- [Setting Up Shapes Demo \(Section 2.1\)](#)
- [Recording Data \(Section 2.2\)](#)
- [Replaying Data \(Section 2.3\)](#)

Important:

- Review [Paths Mentioned in Documentation \(Section 1.1\)](#)
-

2.1 Setting Up Shapes Demo

We'll use *Shapes Demo* to publish and subscribe to Topics which will appear in the application window as colored moving shapes. Each shape is a Topic; the topic's data includes the shape's color; x and y coordinates; a fillKind (solid, empty, horizontal or vertical stripes); and an angle.

To Begin the Tutorial:

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

There are two ways to start *Shapes Demo*.

- From *RTI Launcher*, select the **Tools** tab and click the **Shapes Demo** icon.

or

- From a command prompt, enter (replacing <NDDSHOME¹> to match your system):
 - On Linux and Mac OS X systems: <NDDSHOME>/bin/rtishapesdemo
 - On Windows systems: <NDDSHOME>\bin\rtishapesdemo.bat

1. See [Paths Mentioned in Documentation \(Section 1.1\)](#).

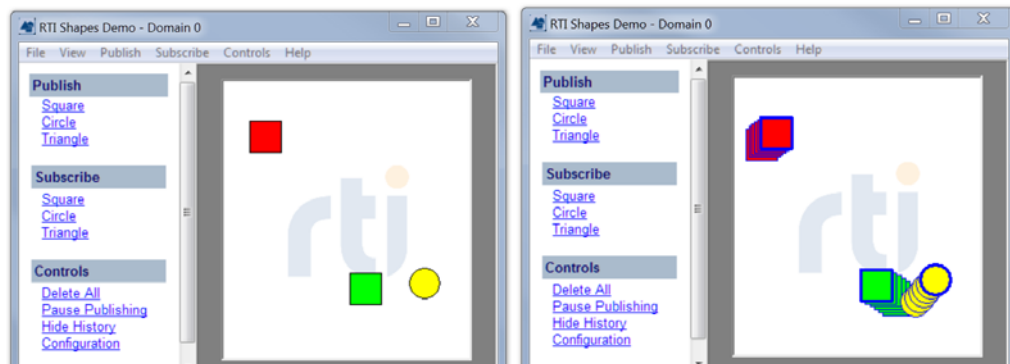
Depending on your security settings, you may see a Windows Security Alert dialog—simply click **Allow Access**.

No configuration changes are necessary. If you choose to use a different domain ID, make sure to use the same value in both instances of *Shapes Demo*, as well as in the *Console's* Configuration panels.

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 as the Subscriber Demo.

2. In the Publisher Demo, create 2 squares and a circle—all with default QoS settings:
 - a. Under **Publish**, select **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 and a yellow circle.
You've just created 3 Writers (2 Squares, 1 Circle).
3. In the Subscriber Demo, subscribe to squares and circles:
 - a. Under **Subscribe**, select **Square** and click **OK**. (Leave the default settings.)
You've just created a single Reader that will receive data for all squares (all colors).
 - b. Use the same process to create a Reader for circles.
Your two *Shapes Demo* windows should look like this:



Publisher

Subscriber

By default, Subscribers show the most recent 6 samples—this is why you see extra shapes behind the leading one. The most recent shape has a thick colored border. Older (historical) samples have a thinner border of the same color. Published shapes have a thin black border.

- c. You may minimize these two windows.

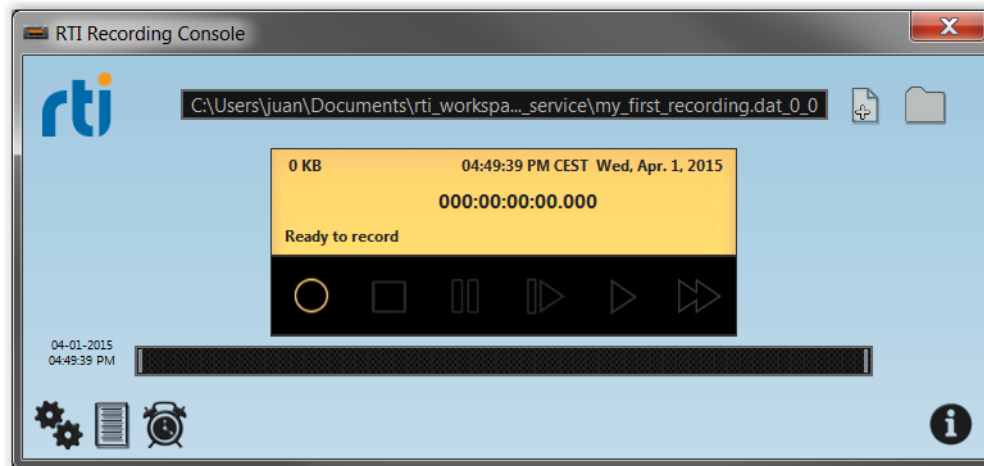
2.2 Recording Data

1. Start *Recording Console*.


There are two ways to start *Recording Console*.

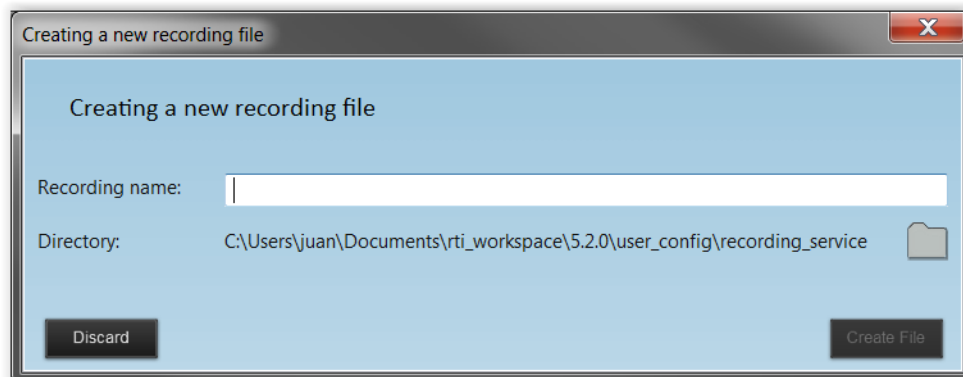
- From *RTI Launcher*, select the **Tools** tab and click the **Recording Console** icon.
- or
- From a command prompt, enter (replacing <NDDSHOME¹> to match your system):
 - On Linux and Mac OS X systems: <NDDSHOME>/bin/rtirecordingconsole
 - On Windows systems: <NDDSHOME>\bin\rtirecordingconsole.bat

2. Wait a few seconds for the *Console* to launch:



3. Create a new file in which to record data from *Shapes Demo*:

- Press the New Recording button  and specify a file name and location. Then click on **Create File**.



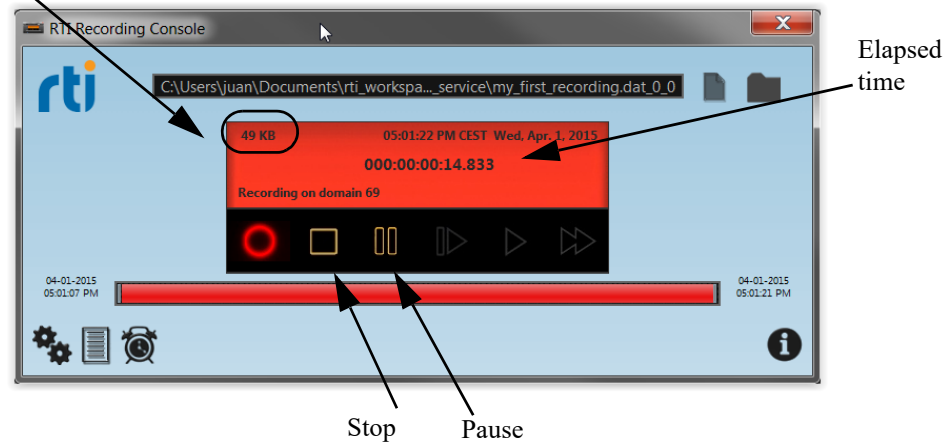
4. Press the Record button to start recording.

(On Windows systems, depending on your security settings, you may see a Windows Security Alert dialog—if so, simply click **Allow Access**.)


1. See [Paths Mentioned in Documentation \(Section 1.1\)](#).

- Wait a few seconds for the *Console* to discover and start recording data from *Shapes Demo*.

File size grows as
data is recorded



Tip: If you have trouble recording, see [Troubleshooting Recording Problems \(Section 2.4.2\)](#)


- In the Publishing Demo, click on the red square and move it around in a pattern, such as a circle or zig-zag. Notice that the red square in the Subscribing Demo is moving the same way. (We will look for this pattern of movement again later in the next section of the tutorial).
- Press the *Console's* Stop button .
- You can close the Publishing Demo. Keep the *Console* and Subscribing Demo running; you will use them in the next section.
- Notice that the shapes in the Subscribing Demo still appear but they are no longer moving. That's because this copy of *Shapes Demo* is no longer receiving data.

2.3 Replaying Data

In this section, we will replay data that was recorded from *Shapes Demo* in [Section 2.2](#).


This example will replay the recorded data with the same QoS settings as when the data was recorded.

You should still be running *Recording Console* and the Subscribing Demo from [Section 2.2](#).

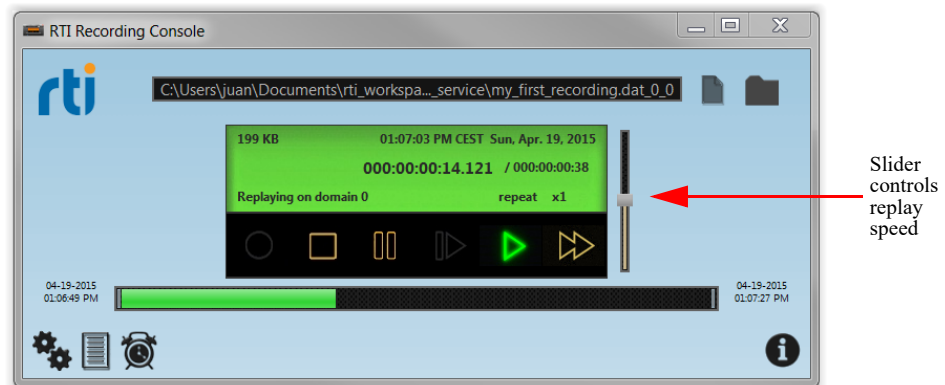
- In the *Console*, press the Open Folder button  in the upper-right corner, locate the file whose data is to be replayed, then click **Open**. This will be the same file you recorded in [Section 2.2](#). (If you skipped that part of the tutorial, you can use the example file in `<path to examples>/recording_service/replay`.)




Another way to open a recording file is simply to drag the file from your file browser and drop it into the long black rectangle at the top of the *Console*.

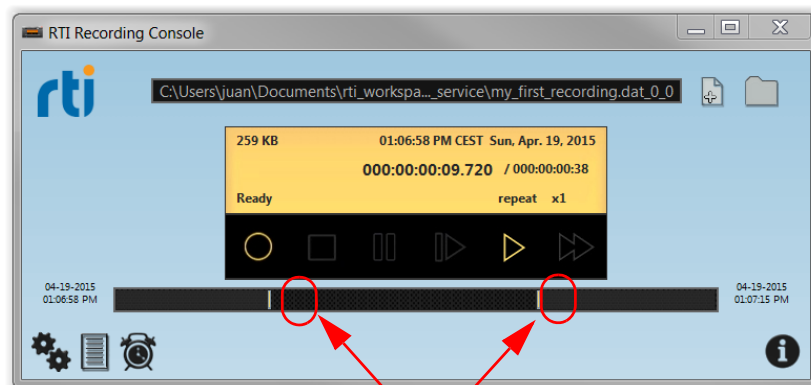
1. See [Paths Mentioned in Documentation \(Section 1.1\)](#).

2. Press Play  to begin replaying the data.


In the Subscribing Demo, you will see the objects moving exactly the same as when they were recorded, including when you moved the red square around in a pattern.

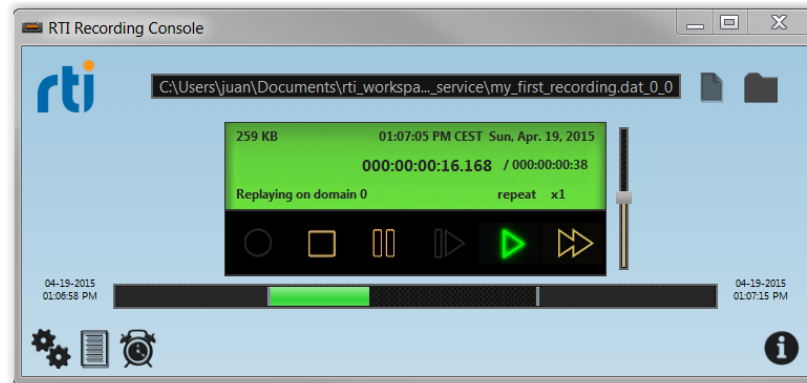




3. Press Stop .
4. Replay the data again, but this time try moving the vertical slider (on the right) which controls the replay rate. Slide it up to go faster, down to go slower. You will be able to see the rate change in the Subscribing Demo. To quickly return to the original speed, press Play .
5. Press Stop .
6. Replay just a section in the middle of the data. Drag the vertical bars on the progress bar inward to shorten the replay time:

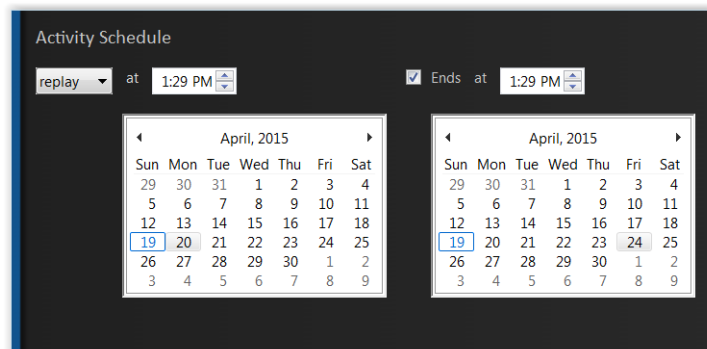



Drag these bars inward to restrict the time range for replaying data.

7. Press Play  to begin replaying the just the middle section of the data, beginning where you put the left bar, and ending where you put the right bar.




8. Press Stop .
9. Slide the bars back to their original positions (to the far left and far right).
10. Now let's see how to schedule a recording. Press the Schedule button .
11. From the drop-down list on the left, select **replay**.
12. Set a starting time for 1 minute from now and a stop time for 1 minute after that. Check the “Ends at” box.

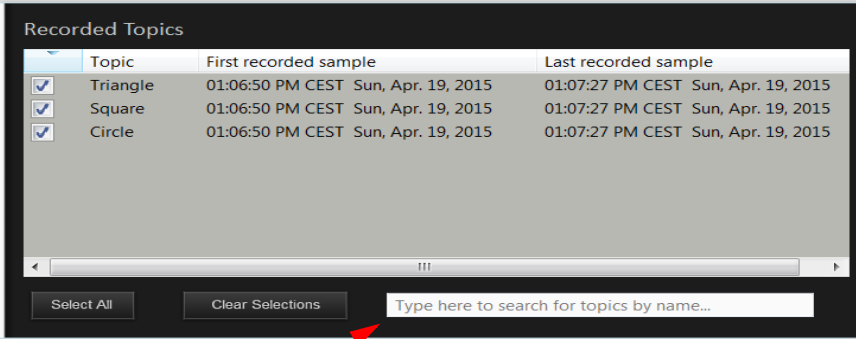


13. Watch the Subscriber Demo and/or the *Console's* display area to confirm that the data is replayed at the scheduled time. If it runs out of data before the stop time it will simply stop early. If you want it to keep repeating the replay, press  to open the Playback Configuration panels and check **Auto repeat** before scheduling the activity.




Auto repeat



14. Press the Topics Table button . This will show you the Topic names, and first and last samples for each Topic.



Topic	First recorded sample	Last recorded sample
<input checked="" type="checkbox"/> Triangle	01:06:50 PM CEST Sun, Apr. 19, 2015	01:07:27 PM CEST Sun, Apr. 19, 2015
<input checked="" type="checkbox"/> Square	01:06:50 PM CEST Sun, Apr. 19, 2015	01:07:27 PM CEST Sun, Apr. 19, 2015
<input checked="" type="checkbox"/> Circle	01:06:50 PM CEST Sun, Apr. 19, 2015	01:07:27 PM CEST Sun, Apr. 19, 2015

Select All Clear Selections Type here to search for topics by name...

Enter Search String 

15. Press “Clear Selections” and then Play . Since no topics are selected, no data will be replayed.
16. Enter “Cir” in the search field at the bottom. Now only Circle topics appear in the table. Check the box next to the Circle.
17. Press Play  again. Notice in the Subscriber Demo that only the circle object is being replayed.

This concludes the tutorial for *Recording Console*.

To learn how to use the individual *Replay* and *Record* tools from the command-line (not through the *Console*), see [Chapter 3: Tutorial—Record and Replay Tools](#). Or for more information, see the *Recording Service User’s Manual*.

Chapter 3 Tutorial—Record and Replay Tools

While the previous chapter’s tutorial focused on recording and replaying data from the *Recording Console*, this tutorial will show you how to use the individual *Replay* and *Record* tools. These are actually the same services that the *Console* invokes.

First, we will generate recordable data via *Shapes Demo*, a *Connex DDS* application. Then we will record data sent between two instances of *Shapes Demo*. Next, we will replay the data. Finally, we will learn how to manage the *Record* tool remotely.

Important:

- Review [Paths Mentioned in Documentation \(Section 1.1\)](#)
- To see the database tables, use a SQLite® viewer tool of your choice, such as SQLite Studio or SQLiteMan.
- To understand the contents of each database record, you will need to consult the *RTI Connex DDS Core Libraries C API Reference HTML* documentation.

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

- *Recording Service*
- *Shapes Demo*

Exercises in the tutorial include:

- [Setting Up Shapes Demo \(Section 3.1\)](#)
- [Configuring the Record Tool \(Section 3.2\)](#)
- [Running the Record Tool \(Section 3.3\)](#)
- [Viewing the Recorded Data with SQLite \(Section 3.4\)](#)
- [Using Replay \(Section 3.5\)](#)
- [Managing the Record Tool Remotely \(Section 3.6\)](#)

3.1 Setting Up Shapes Demo

We'll use *Shapes Demo* to publish and subscribe to Topics which will appear in the application window as colored moving shapes. Each shape is a Topic; the topic's data includes the shape's color; x and y coordinates; a fillKind (solid, empty, horizontal or vertical stripes); and an angle.

To Begin the Tutorial:

1. Start *two* copies of *Shapes Demo*.
 - From *RTI Launcher*, select the **Tools** tab and click the **Shapes Demo** icon.
 - or
 - From a command prompt, enter (replacing <NDDSHOME¹> to match your system):
 - On Linux and Mac OS X systems: <NDDSHOME>/bin/rtishapesdemo
 - On Windows systems: <NDDSHOME>\bin\rtishapesdemo.bat

Depending on your security settings, you may see a Windows Security Alert dialog—simply click **Allow Access**.

No configuration changes are necessary. If you choose to use a different domain ID, make sure to use the same value in both instances of *Shapes Demo*, as well as in the *Console's* Configuration panels.

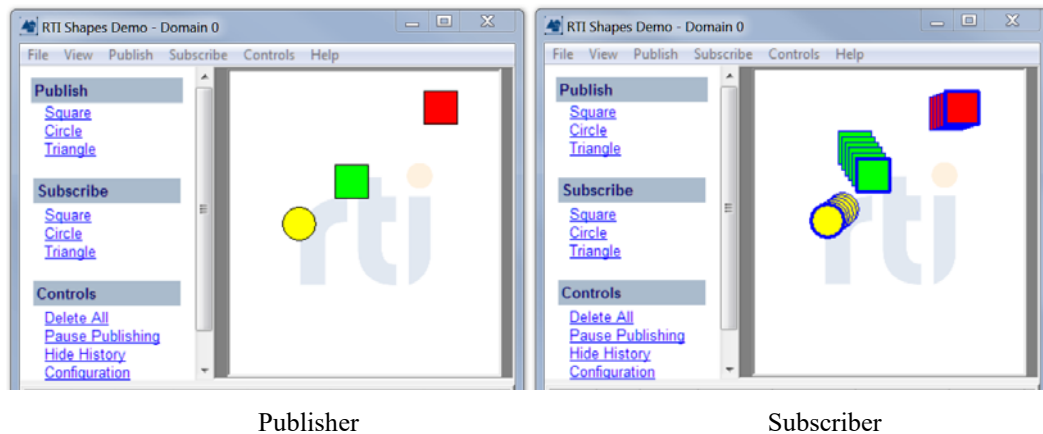
To start a second instance: In *Shapes Demo*, select **File, New Shapes Demo**.

Now you should be running two instances of *Shapes Demo*. We'll refer to the first Demo window as the Publisher Demo and the second as the Subscriber Demo.

2. In the Publisher Demo, create 2 squares and a circle—all with default QoS settings:
 - a. Under **Publish**, select **Square**.
 - b. In the **Create New Publisher** dialog, select **RED** and click **OK**. (We'll 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 and a yellow circle.
You've just created 3 Writers (2 Squares, 1 Circle).
3. In the Subscriber Demo, subscribe to squares and circles:
 - a. Under **Subscribe**, select **Square** and click **OK**. (Leave the default settings.)
You've just created a single Reader that will receive data for all squares (all colors).
 - b. Use the same process to create a Reader for circles.

1. See [Paths Mentioned in Documentation \(Section 1.1\)](#).

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



By default, Subscribers show the most recent six samples—this is why you see extra shapes behind the leading one. The most recent shape has a thick colored border. Older (historical) samples have a thinner border of the same color. Published shapes have a thin black border.

- c. You may minimize these two windows.

3.2 Configuring the Record Tool

You may specify a configuration file when you start the *Record* tool. The configuration file is used to set properties that control what to record, how to record, and where to save the recorded data.

The `<path to examples>/recording_service/record` directory contains these example files:

- `simple_config.xml`
- `advanced_config.xml`
- `remote_shell.xml`

Let's start learning about the *Record* tool with the simple configuration.

1. Copy `<path to examples>/recording_service/record/simple_config.xml` to `<path to examples>/recording_service/record/mysimple_rerecord_config.xml`.
2. If your computer has an active network interface *and* you are using the default domain ID (0) for *Shapes Demo*: proceed to [Running the Record Tool \(Section 3.3\)](#)—you can use the copied configuration file as is.

Otherwise, open `mysimple_record_config.xml` with an editor of your choice, make the following changes, then save and close the file:

- a. If you changed the domain ID used by *Shapes Demo*, change the `<domain_id>` line to use the same value. For example, if you set the domain ID to 3 in *Shapes Demo*, then change it here too:

1. See [Paths Mentioned in Documentation \(Section 1.1\)](#).

Change the bold
line to use the
same domain
ID as Shapes
Demo

```
<domain name="domain0">
<domain_id> 3 </domain_id>
<deserialize_mode>
    RTIDDS_DESERIALIZEMODE_ALWAYS
</deserialize_mode>
</domain>
```

- b. If your computer does not have an active network interface, use the following lines in the domain properties section.

Turn off
multicast—>

Only use shared
memory—>

```
<domain name="domain0">
  <domain_id> 0 </domain_id>
  <participant_qos>
    <discovery>
      <multicast_receive_addresses/>
    </discovery>
    <transport_builtin>
      <mask>SHMEM</mask>
    </transport_builtin>
  </participant_qos>
</domain>
```

3.3 Running the Record Tool

1. Open a command prompt¹ and change to the <NDDSHOME²>/bin directory.
2. Start the *Record* tool with your new configuration file from Section 3.2. For example (enter this all on one line and replace <path to examples> as described in [Paths Mentioned in Documentation \(Section 1.1\)](#)):

```
rtirecord -cfgFile
<path to examples>/recording_service/record/mysimple_record_config.xml
-cfgName example
```

On Windows systems when using an evaluation version of *Recording Service*: If you are not logged in as Administrator, the license manager will display a message:

```
Error in saving configuration settings: You are not a license adminis-
trator.
```

This message is benign and the *Record* tool should run normally.

On Windows systems, depending on your security settings, you may see a Windows Security Alert dialog—if so, simply click **Unblock**.

1. On Windows systems: from the Start menu, select **Accessories, Command Prompt**.
2. See [Paths Mentioned in Documentation \(Section 1.1\)](#).

3. Wait about one minute to give the *Record* tool time to record data from *Shapes Demo*.
4. Stop the *Record* tool by pressing <Control-C>. Leave the command prompt window open—you will use it in [Section 3.4](#).

3.4 Viewing the Recorded Data with SQLite

1. After running and stopping the *Record* tool as described in [Section 3.3](#), list the contents of the current directory. You will see that the *Record* tool created a file called **simple_config.dat_0_0**. (The first 0 is a fileset number, the second 0 is a segment number.)
2. With a SQLite viewer of your choice, open **simple_config.dat_0_0**.
3. Look at the list of the recorded tables.

The *Record* tool creates this table for discovery data:

- DCPSPublication (data type DDS_PublicationBuiltinTopicData)

This table stores the recorded topic data:

- Circle\$RecordAll\$domain0
- Square\$RecordAll\$domain0

Topic table names follow this format: *TopicName\$RecordGroupName\$DomainName*. A “RecordGroup” is a way to specify what topics to record from what domains. This concept is explained in the *Recording Service User’s Manual*. In this example, the RecordGroup is called **RecordAll**.

The *Record* tool also creates two tables for internal use, **RTILog** and **RTIVersion**.

4. Open one of the recorded tables that contains user data, such as Circle\$RecordAll\$domain0. The *Record* tool records each field of the data type in its own column, alongside some metadata fields.
5. Close the SQLite viewer when you are done.

3.5 Using Replay

In this section, you will learn about:

- [Recording Data for Replay \(Section 3.5.1\)](#)
- [Changing Replay’s Configuration File \(Section 3.5.2\)](#)
- [Replaying Data at Normal Speed \(Section 3.5.3\)](#)
- [Replaying Data at a Faster Rate and Using Remote Control \(Section 3.5.4\)](#)

3.5.1 Recording Data for Replay

To get started with *Replay*, you can either (a) use the example database provided in <path to examples>/**recording_service/replay/replay_database.dat_0_0** or (b) configure and run *Record* using the steps in [Section 3.2](#) and [Section 3.3](#) with the **replay_example** configuration instead of **example**. Enter this all on one line and replace <path to examples> as described in [Paths Mentioned in Documentation \(Section 1.1\)](#):

```
rtirecord -cfgFile
<path to examples>/recording_service/record/mysimple_record_config.xml
-cfgName replay_example
```

Remember the name of the data file that gets created; you will need it in the next section.

Replay can replay data that has been recorded in either serialized or deserialized form. For files recorded in serialized mode, SQLite is unable to look at the individual fields in the sample data. For files recorded in deserialized mode, it is important for the *Record tool* to record *all* of the fields of the sample data, as *Replay* is unable to replay partial data.

3.5.2 Changing Replay's Configuration File

Replay requires an XML configuration file. We will use the example file, **simple_config.xml**, provided in `<path to examples>/recording_service/replay`. (Note that this is not the same file that you changed in [Section 3.5.1](#)—that was *Record's* **simple_config.xml**; now we are using *Replay's* **simple_config.xml**.)

1. Copy `<path to examples>/recording_service/replay/simple_config.xml` to `<path to examples>/recording_service/replay/mysimple_replay_config.xml`.
2. The **simple_config.xml** configuration file assumes that the database containing serialized samples is named **replay_database.dat_0_0** (this file is in `<path to examples>/recording_service/replay`); if you want to replay this example database, skip the rest of this step.

However, if you want to replay a file that has a different name, you will need to edit **mysimple_replay_config.xml**. To change the name of the database file in **mysimple_replay_config.xml**, change the value of the `<filename>` parameter (just below the tag for `<replay_database>`).

The lines to look for are:

```
<!-- Source Database, Required -->
<replay_database name="simple_config">
  <filename>replay_database.dat_0_0</filename>
```

Note: There are three configurations in **simple_config.xml**, so you will need to **change the filename in three places** if you want to use a different filename.

3. The configuration file assumes you want to use domain ID 0 for the participant that will replay the data and domain ID 1 for remote administration. If that's alright, you can skip this step.

Otherwise, you will need to change the domain ID(s) in the configuration file, or specify the replay domain ID and/or the remote administration domain ID on the command line with **-domainIdBase** and **-remoteAdministrationDomainId** options, respectively.

In the configuration file, the domain ID assignment for replaying the data is right after the filename parameter and looks like this:

```
<participant>
  <domain_id>0</domain_id>
</participant_qos>
```

The replay participant domain ID assignment occurs **in three places**, so be sure to change all of them.

The domain ID assignment for remote administration is near the top of the service configuration) and looks like this:

```
<administration>
  <name>example</name>
```

1. See [Paths Mentioned in Documentation \(Section 1.1\)](#).

```
<domain_id>1</domain_id>
</administration>
```

The remote administration domain ID is only set in one of the examples.

3.5.3 Replaying Data at Normal Speed

This example will replay the recorded data once, at the same speed and with the same QoS as when the data was recorded. The data will be replayed on the domain ID specified in the configuration (default 0), or on the domain ID specified on the command line.

1. Start one copy of *Shapes Demo*, configured as in [Section 3.2](#), and create subscribers for Squares and Circles.
2. Open a command prompt¹ and change to the `<NDDSHOME>/bin` directory.
3. Start *Replay* with your new configuration file from [Section 3.5.2](#). Enter this all on one line and replace `<path to examples>` as described in [Paths Mentioned in Documentation \(Section 1.1\)](#):

```
rtireplay -cfgFile
<path to examples>/recording_service/replay/mysimple_replay_config.xml
-cfgName simple_example
```

Expected Results:

In *Shapes Demo*, you should see the objects moving exactly the same as when they were recorded.

4. When the example has finished replaying, press `<Control-C>` to exit.
5. In the XML configuration file, look at the configuration named **simple_example** (the first example).

This example defines a single database (**simple_config**), with a single session (**A_Session**), and a single topic (**All_Topic**), yet *Replay* replayed the data from both Squares and Circles. This happened because the topic description uses wildcards ("*") to indicate that *Replay* should replay any topics in the database with any values of topic name, type name, record group name, and domain name. *Replay* created a separate *DataWriter* for each topic that it found matching the selection parameters; in this example, there were two: Squares, and Circles. In the next example we will be more discriminating.

3.5.4 Replaying Data at a Faster Rate and Using Remote Control

This example demonstrates a higher degree of control over the parameters used during replay. We will replay the data faster than its original speed by using a different configuration named **fast_replay**. And while the data is replaying, you will see how to remotely control the replay by pausing, resuming, and single-stepping.

1. Open a command prompt³ and change to the `<NDDSHOME>/bin` directory.
2. Start *Replay* in the same manner described in [Section 3.5.3](#), with these minor changes to the command line:
 - Replace `-cfgName simple_example` with `-cfgName fast_replay`

1. On Windows systems: from the Start menu, select **Accessories, Command Prompt**.

2. See [Paths Mentioned in Documentation \(Section 1.1\)](#).

3. On Windows systems, from the Start menu, select **Accessories, Command Prompt**.

- Add the parameter **-remoteAdministrationDomainId** followed by the domain ID that you want to use for remote operation. (You can omit this option if you already changed the remote administration domain ID assignment in the configuration file as discussed in [Section 3.5.2](#)).

For example, enter this all on one line and replace <path to examples> as described in [Paths Mentioned in Documentation \(Section 1.1\)](#):

```
rtireplay -cfgFile
<path to examples>/recording_service/replay/mysimple_replay_config.xml
-cgfName fast_replay -remoteAdministrationDomainId <other_domain_id>
```

Expected Results:

You should see the objects moving in *Shapes Demo* at a higher rate of speed than in the previous example.

In the XML configuration file, look at the configuration named **fast_replay** (the second example). You will see that the higher rate of replay was configured using the <rate> parameter of the <replay_service> <time_control> section of this configuration.

By this time you should also notice that this replay does not stop, but repeats the movement of the objects once the original data has been replayed. You can see in the configuration that this was selected by setting the mode **LOOP** for the <start_mode> parameters of the <replay_topic> <topic_time_control> sections of **A_Topic** and **B_Topic**. (Repeating the data allows us to experiment with remote control, without having the data stream come to an end.)

You can also see that both **A_Topic** and **B_Topic** are more discriminating in their choice of topics to replay, explicitly naming the topics and types of interest. Separating the two topics allows them to be controlled individually.

3. Now we'll see how to control *Replay* remotely using a program called **rtireplaysh** (the *Replay* shell).
 - a. Open a command prompt and change to the <NDDSHOME>/bin directory.
 - b. Start *Replay Shell* using the same domain ID that you used for the **-remoteAdministrationDomainId** argument in the previous step:

```
> rtireplaysh -domainId <other_domain_id>
```

You will see this prompt:

```
"Replay Shell> "
```

- c. Enter "**help**" to see what commands are supported.

Replay Shell commands use this format:

```
<command> <replay_service> <entity> [value]
```

where:

- <command> is one of the supported commands.
- <replay_service> is the name given to the *Replay* service by one of the following, in descending order of precedence:
 - Value specified with the **-appName** command-line option used when starting *Replay* (*highest precedence*)
 - Value for the <replay service><administration><name> element
 - Value for the <replay_service> name attribute (*lowest precedence*)

- **<entity>** is any one of the service entities expressed in this hierarchical form: **<database-name>[:<session-name>[:<topic-name>]]**.
- **value** depends on the command. Not all commands require a value.

Note: *In this release, not all commands are supported for all entity levels.* Please see the *Recording Service Release Notes* for details on which modes are currently supported.

The database-name must match a name from a `<replay_database>` tag in the configuration file that you specified when starting *Replay*, such as:

```
<replay_database name="simple_config">
```

Similarly, if you specify a session-name, it must match a name from a `<session>` tag within the specified database, such as:

```
<session name="A_Session">
```

If you specify a topic-name, it must match a name from a `<replay_topic>` tag within the specified session, such as:

```
<replay_topic name="All_Topic">
```

4. While topics are being replayed, you can pause or stop the replay. For example, to pause the replay, use this command:

```
Replay Shell> pause example simple_config::A_Session
```

Note: The *Replay Shell* has no line-editing capabilities, so take extra care when entering commands.

5. Once paused, you can resume or step like this:

```
Replay Shell> step example simple_config::A_Session
Replay Shell> resume example simple_config::A_Session
```

6. It is also possible to control individual topics:

```
Replay Shell> pause example simple_config::A_Session::A_Topic
Replay Shell> step example simple_config::A_Session::A_Topic
Replay Shell> resume example simple_config::A_Session::A_Topic
Replay Shell> rate example simple_config::A_Session::A_Topic 1
Replay Shell> rate example simple_config::A_Session::B_Topic 1
```

3.6 Managing the Record Tool Remotely

The *Record* tool can be managed remotely using one of two ways:

- *RTI Record Shell*
- Another *Connext DDS* application using special message types

See Chapter 7 in the *Recording Service User's Manual* for details.

This section contains examples of how to use the *Record Shell* commands.

The configuration files used in the examples are in `<path to examples>/recording_service`.

Note: If you are using different domain IDs for running the *Record* tool, you must edit the configuration files to use your domain IDs. By default, the example configuration files use domain IDs 0 and 1.

3.6.1 Connecting to the Record Tool using Shared Memory

1. Change to this directory: `<NDDSHOME>/bin`
2. Start the *Record* tool. Enter this all on one line and replace `<path to examples>` as described in [Paths Mentioned in Documentation \(Section 1.1\)](#):

```
rtirecord -cfgFile  
<path to examples>/recording_service/record/advanced_config.xml  
-cfgName remote_example
```

3. In another shell, change to `<NDDSHOME>/bin` and start the *Record Shell*:

```
cd <NDDSHOME>/bin  
rtirecsh -domainId 0 -noUdpv4
```

4. Enter **info** (in the shell that is running **rtirecsh**) to see which *Record* session it is connected to.
5. Enter **status** to check *Record*'s status. Note that State is RECORDING.
6. Enter **stop** to tell *Record* to stop recording.
7. Enter **status**. The State value should be STOPPED, indicating that recording has been stopped.
8. Reconfigure the *Record* tool with a file that is local to the *Record* tool (enter this on one line and replace `<path to examples>` as described in [Paths Mentioned in Documentation \(Section 1.1\)](#)):

```
configure advanced_example -remotefile  
<path to examples>/recording_service/record/advanced_config.xml
```

9. Wait a few seconds and enter **status** in the shell. The *Record* tool should now be configured based on **advanced_config.xml**.
10. Wait a few seconds and enter **stop** in the shell. The *Record* tool should now be stopped.
11. Reconfigure the *Record* tool with a file that is local to the Record Shell tool (enter this on one line and replace `<path to examples>` as described in [Paths Mentioned in Documentation \(Section 1.1\)](#)):

```
configure advanced_example -localfile  
<path to examples>/recording_service/record/advanced_config.xml
```

12. Wait a few seconds and enter **status** in the shell. The Record tool should now be configured based on **advanced_config.xml**.
13. Enter the **stop** command to stop recording.
14. Enter the **shutdown** command to shut down the *Record* tool .
15. Enter the **exit** command to exit the shell.