# *RTI Recording Service*

## Release Notes

Version 5.0.0

Your systems. Working as one.

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

## Trademarks

Real-Time Innovations, RTI, and Connext 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



## 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

# Release Notes

## 1 System Requirements

*RTI® Recording Service* is supported on these platforms:

- ❑ Linux Platforms:
  - CentOS 5.4, 5.5 (2.6 kernel)
  - Fedora 12 (2.6 kernel)
  - Red Hat Enterprise Linux 5.0-5.2, 5.4, 5.5, 6.0, 6.1 (2.6 kernel)
  - Ubuntu Server 10.04 (2.6 kernel)
- ❑ All Windows platforms listed in the *Core Libraries and Utilities Release Notes* for *RTI Connext™* (formerly *RTI Data Distribution Service*) with the same version number.

## 2 Additional Libraries Needed on Select 64-Bit Platforms

If you are installing *Recording Service* on an Ubuntu 64-bit platform or a Red Hat Enterprise Linux 5 or 6 64-bit platform, you will need to install a set of 32-bit libraries:

- ❑ On Red Hat Enterprise Linux 5 and 6 64-bit platforms, run this command (you will need *root* permission):

```
yum install glibc.i686 libX11.i686 gtk2.i686 libXtst.i686
```

- ❑ On Ubuntu 64-bit platforms, run this command (you will need *root* permission):

```
apt-get install libc6-i386 ia32-libs
```

# 3 Compatibility with Other RTI Products

*Recording Service* is compatible with *RTI Connext*, as well as *RTI Data Distribution Service* 4.5[b-e], 4.4d, 4.4b, 4.3e and 4.2e[1].

The *Record* tool supports the standard *Connext* transports (UDPv4, UDPv6, and shared memory), as well as the *RTI Secure WAN Transport* plugins.

**Note:** *Recording Service* is not compatible with applications built with *RTI Data Distribution Service* 4.5e and earlier releases when communicating over shared memory. For more information, please see the Transport Compatibility section in the *RTI Core Libraries and Utilities Release Notes*.

The *Replay* tool works with database files created by any version of *RTI Recording Service.*

The *Recording Console* is a graphical user-interface that internally uses the *Record* and *Replay* tools. It can replay data that was recorded with previous versions of *Recording Service* (as long as the QoS settings used to replay the data match those used when it was originally recorded.)

# 4 What's New in 5.0.0

## 4.1 Integrated Support for Distributed Logger

The *RTI Distributed Logger* library is now included with *RTI Recording Service*.

When you enable the *Distributed Logger* library for the *Record* and/or *Replay* tools, they will publish their log messages to *Connext*. Then you can visualize the log message data with *RTI Monitor*, a separate GUI application that can run on the same host as your application or on a different host. Since the data is provided in a *Connext* topic, you can also use *rtiddsspy* or even write your own visualization tool.

For details on how to enable the *Distributed Logger* library, see the chapters on *Configuring the Record Tool* and *Configuring the Replay Tool* in the *RTI Recording Service User's Manual*, as well as the *RTI Distributed Logger Getting Started Guide*. These documents will show you how to use the new XML configuration tag, <distributed_logger>.

1. To support compatibility with 4.2e, please see the *RTI Core Libraries and Utilities Release Notes*.

## 4.2 Support for Extensible Types

*Recording Service* includes partial support for the "*Extensible and Dynamic Topic Types for DDS*" specification from the Object Management Group (OMG). See Section 4.10 in the *Recording Service User's Manual* for details.

## 4.3 Version Number Now Propagated as a DomainParticipant Property

The *Recording Service* version number is now propagated as a DomainParticipant property called "rti.service.version". The format of the value is as follows:

<major>.<minor>.<release>.rev<revision>

The version property is set in all the DomainParticipants created by the service.

## 4.4 Changes to Default Configuration Files for Better Integration with RTI Administration Console

The default XML configuration files for the *Record* and *Replay* tools have been modified. They now contain the following configurations:

**Recorder (RTI_RECORDING_SERVICE.xml):**

- **default**: This configuration records all topics and all fields in these topics, for any topic discovered in domain 0. Data is always recorded in deserialized mode. Remote administration with *Distributed Logger* is enabled on domain 0, with the filter set to warning level.
- **defaultNoAutoStart**: This configuration records all topics and all fields in these topics, for any topic discovered in domain 0. Data is always recorded in deserialized mode. Remote administration with *Distributed Logger* is enabled in domain 0, with the filter set to warning level. Recording does not start automatically, it has to be started manually using remote administration.
- **defaultNoAdmin**: This configuration records all topics and all fields in these topics, for any topic discovered in domain 0. Data is always recorded in deserialized mode. Remote administration and distributed logging are not enabled.

**Replay (RTI_REPLAY_SERVICE.xml):**

- **default**: This configuration replays every table in the database (any topic or type, any record group and any domain name) in domain 0, at the original recording speed. Remote administration of the service is enabled in domain 0. Distributed logging is enabled, with the filter set to warning level.
- **defaultNoAdmin**: This configuration replays every table in the database (any topic or type, any record group and any domain name) on domain 0, at the speed the data was recorded. Remote administration of the service is disabled.

## 4.5 Support for Environment Variables in XML Configuration Files

This new feature allows you to refer to an environment variable within an XML tag. When the *Connext* XML parser parses the configuration file, it will expand the environment variable. To refer to an environment variable, use the format $(MY_VARIABLE).

For example:

```
<element>
    <name>The name is $(MY_NAME)</name>
    <value>The value is $(MY_VALUE)</value>
</element>
```

# 5 What's Fixed in 5.0.0

## 5.1 Recording Console Wrongly Reported a Failure to Replay

When creating a very short recording, or replaying a short recording at a very high rate, *Recording Console* reported that the replay failed, when in fact the replay operation was completed in under a second.

*Recording Console* now prevents playback if it would take less than 1 second to replay. In this case, you will see a notice in the main display that "Recording is too short for the current playback speed."

If you see this message, adjust the playback speed to a slower rate in order to enable playback.

[RTI Issue ID RECORD-164]

## 5.2 Unable to Replay after Cancelling Recording to an Existing File

When initiating a recording in a non-empty file, *Recording Console* prompts you to confirm this action. Due to an internal error, if you esponded to the prompt with "Cancel" *Recording Console* would not allow you to replay the file until it was reloaded by the console.

This problem has been resolved. Now *Recording Console* behaves as expected—it will cancel the recording operation, return to a normal state, and allow you to replay the contents of the file.

[RTI Issue ID RECORD-168]

## 5.3 Unable to Replay Recorded Serialized Data on Platform with Different Endianness

When replaying data that was recorded on a machine with a different endianness, re-serialization of the samples was not working properly and it generated corrupted samples. This problem has been resolved.

[RTI Issue ID RECORD-195]

## 5.4 Some Configuration Settings Overwritten by Recording Console

When using a file to configure *Recording Service*, some of the attributes used by *Recording Console* to configure remote administration would override the original configuration settings in the file, namely:

- **dds.transport.UDPv4.builtin.recv_socket_buffer_size**
- **dds.transport.UDPv4.builtin.parent.message_size_max**
- **dds.transport.UDPv4.builtin.send_socket_buffer_size**
- **dds.transport.shmem.builtin.parent.message_size_max**
- **dds.transport.shmem.builtin.receive_buffer_size**

This was due to the fact that the administration domain used by *Recording Console* to manage and monitor the record and replay services was the same domain ID used for recording and replaying data. To correct this problem, *Recording Console* now uses domain 99 to administer the record and replay services.

Therefore, when using a file to configure the *Record* or *Replay* tools, if you are using domain 99 as your recording or replay domain, we recommend that you change the administration domain ID from 99 to a different domain ID. This can be done by either specifying the domain ID in the configuration file's administration section or by editing the file **settings.ini**, which should be located in your home directory:

- **On Windows Systems:**
  **My Documents\RTI\RTI Recording Service 5.0.0\console\settings.ini**
- **On Linux systems:**
  **~/RTI/RTI Recording Service 5.0.0/console/settings.ini**

Again, this recommendation is only needed if you are using domain ID 99 in your configuration file for recording or replaying data.

[RTI Issue ID RECORD-205]

### 5.5 Long Delay when Reconfiguring Record Tool

When using the *Record* tool's **configure** command to change its configuration, the first reconfiguration could take a very long time to start recording and sending status updates again. This problem has been resolved.

[RTI Issue ID RECORD-208]

### 5.6 Data not Recorded when using Recording Console and Partitions

*Recording Console* did not record topics that used partitions. This problem has been resolved.

[RTI Issue ID RECORD-215]

## 6 Known Issues

- When using the *Convert* utility on a recording created with *Recording Console*, you may see a warning related to internal topics used by the *Console*:

  ```
  exception:[RTIConverterModelPublisherCallback@2293]:Failed to
  create type com_rti_tools_remotectx
  ```

  You can safely ignore the warning—the conversion results *are* valid.

- To record a data type that has more than 1,950 primitive types, you must set the **deserialize_mode** property to RTIDDS_DESERIALIZEMODE_NEVER. Otherwise, you will see the following error message and recording will fail:

  ```
  "exception:[RTIDRTUserDataTable_update@610]:too many SQL variables"
  ```

  [RTI Issue ID RECORD-38; Bug # 12794]

- The DynamicData API does not support out-of-order assignment of members with a length greater than 65,535 bytes. In this situation, the following error is reported:

  ```
  sparsely stored member exceeds 65535 bytes
  ```

  For example:

  ```
  struct MyStruct {
      string<131072> m1;
      string<131072> m2;
  };
  ```

  With the above type, the following sequence of operations will fail because **m2** is assigned before **m1** and has a length greater than 65535 characters.

```
str = DDS_String_alloc(131072);
memset(str, 'x', 131072);
str[131071]= 0;
DDS_DynamicData_set_string(
    data,"m2", DDS_DYNAMIC_DATA_MEMBER_ID_UNSPECIFIED, str);
DDS_DynamicData_set_string(
    data,"m1", DDS_DYNAMIC_DATA_MEMBER_ID_UNSPECIFIED, str);
```

If the member **m1** is assigned before **m2,** the sequence of operations will succeed.

[RTI Issue ID CORE-3791; Bug # 13745]

- RTI does not recommend using files that are mounted over NFS to store recorded data. *Recording Service* uses file-locking, which has known issues working over NFS. If file-locking is not working, *Recording Service* will hang. In particular, this problem may appear on Yellow Dog Linux systems.
- Leading and trailing spaces in a Topic Name are ignored. However, spaces within the string are allowed. For example, " My Topic " will be treated as "My Topic".
- Fully qualified field names in struct's cannot be longer than 1,024 characters.
- Sequence and array indices cannot be used in Topic or Field expressions.
- Topics of DynamicData types that contain bitfields are not supported.
  [RTI Issue ID CORE-3949; Bug # 13949]
- *Recording Service* and *Converter* (**rtirecconv**) cannot deserialize bitfields. If this type is used, the deserialize mode must be RTIDDS_DESERIALIZEMODE_NEVER.
- In files recorded on Windows systems, the recorded timestamp is the number of microseconds since the device was booted, not since January 1, 1970. Therefore the **-time gmt** option to *Converter* (**rtirecconv**) will not show the correct time.
- *Converter* (**rtirecconv**) cannot convert tables with only a subset of the data. In general, if you record in deserialized mode, use the **sqlite3** command to convert to HTML and CSV; if you record in serialized mode, use *Converter*.
- If the *Connext* application being recorded has a keyed data-type and **DataWriter-ProtocolQosPolicy.disable_inline_keyhash** is set to TRUE (not the default), then *Recording Service* may misinterpret samples as being from the wrong instance.

- If you start an instance of the *Record* tool using command-line options (not a configuration file), then sending a new configuration to that instance of the *Record* tool using the remote shell will not work.
- The *Replay* tool currently does not support the following XML configuration modes:
  - <replay_service> <auto_exit> (has no effect)
  - <replay_topic> <output> <keyed> (has no effect)
  - <time_control> <start_mode> MATCHED or LOOP modes
  - <time_control> <rate> AS_FAST_AS_POSSIBLE (except for session level)
  - <topic_time_control> <start_mode> MATCHED mode
- Limitations with the *Replay* tool's shell commands:
  - The **step** command is functional for session and topic entities only (not service or database)
  - The **rate** command is functional for topic entities only
- Performance and indexing with the *Replay* tool:

  The *Replay* tool replays stored samples in the same order in which they were received, using SQLite indexes to retrieve the samples in sorted order. SQLite automatically builds indexes when opening an SQLite table for sorted access, and for large tables the process of building the index may take some time. To improve initialization performance, the *Replay* tool attempts to create and store indexes, rather than depend upon automatic indexing, for the tables which it will be replaying, saving initialization time on subsequent replays.

  The *Replay* tool's ability to store indices is controlled by the <readonly> parameter of the <replay_database>. The default value of true (readonly) will prevent the *Replay* tool from storing an index for a table; in this mode. the *Replay* tool will display a message during initialization for each table opened stating that it was unable to store the table index. Changing the value of <readonly> to false will allow the *Replay* tool to write the table indices to the database.

  In summary, the *Replay* tool's performance is not affected by this option; it will use the fastest means of retrieving samples in either case. But setting the <readonly> option to false may help improve initialization performance.
- When changing playback speed, or skipping to another playback location, occasionally playback will appear stuck (it is actually paused). The workaround is to click the Pause button twice.

- When loading a large file for playback, please be aware that this operation may not be instantaneous and could take some time.
- If you load the configuration file, **examples/replay_simple_config.xml**, and select the **fast_replay** configuration profile while using your own recorded data file (instead of the example recording from RTI), the *Replay* service will exit and log a message regarding 'no match in the recording for A_Topic.'
- The *Record* and *Replay* Shells are not completely compatible with standard input piping of commands.
- When <time_mode> is set to TOPIC_RELATIVE, the first sample in a recording is not sent right away when replay starts. [RTI Issue ID RECORD-133; Bug # 14300].
- *Recording Console* may fail to shut down gracefully after stepping through to the end of a recording. If a recording is paused and then stepped through to the end, the *Replay* service may not shut down properly. In this case, *Recording Console* displays an error that the service stopped unexpectedly. [RTI Issue ID RECORD-135; Bug # 14340]
- For *RTI Admininstration Console* to work properly with the *Replay* tool, do not use the XML <name> tag under <administration>. *Admin Console* will not recognize the replay service and will not be able to administer it. This will be addressed in a future release. [RTI Issue ID BIGPINE-429]
- Interaction between *Recording Console* and *RTI Administration Console*

  This issue only applies if you are using *Recording Console* and *RTI Admin Console* at the same time, and you have configured *Admin Console* to join domain ID 99. In this scenario, do not use *Admin Console* to pause or disable any *Recording Console* services (their names begin with "RTI-Recorder-" or "RTI-Replay-"). Doing so may cause an error in *Recording Console*. [RTI Issue ID BIGPINE-795]