

# *RTI Recording Service*

for

**RTI Data Distribution Service**

## **Release Notes**

Version 4.5e Rev. 01



The Global Leader in DDS



© 2011 Real-Time Innovations, Inc.  
All rights reserved.  
Printed in U.S.A. First printing.  
December 2011.

### **Trademarks**

Real-Time Innovations and RTI are registered trademarks of Real-Time Innovations, Inc. All other trademarks used in this document are the property of their respective owners.

### **Copy and Use Restrictions**

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

### **Technical Support**

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

# Release Notes

---

## 1 System Requirements

*RTI Recording Service* 4.5e is supported on these platforms:

Linux Platforms:

- CentOS 5.4, 5.5 (2.6 kernel)
- Fedora 12 (2.6 kernel)
- Red Hat Enterprise Linux 4.0, 5.0-5.2, 5.4, 5.5, 6.0, 6.1 (2.6 kernel)

**Note:** *RTI Recording Console* is not supported on Red Hat Enterprise Linux 4.0. All other components of *RTI Recording Service* (*Replay*, *Record*, etc.) are supported on this platform.

- Ubuntu Server 10.04 (2.6 kernel)

All Windows platforms listed in the *RTI Data Distribution Service Release Notes* for 4.5e.

---

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

If you are installing *RTI 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

*RTI Recording Service* 4.5e is compatible with *RTI Data Distribution Service* 4.5[b-e], 4.4d, 4.4b, 4.3e and 4.2e<sup>1</sup>.

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

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

*RTI 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 *RTI 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 4.5e Rev. 01

This release includes the following improvements to *RTI Recording Console*:

- Improved Topic Table:
  - More efficient handling of tables with a large number of topics (tested 1,000)
  - Better scrolling and sorting
  - Clicking on a timestamp in the table during playback will skip playback to that point in time
- Improved accuracy in the progress bar and playback limiters
- Easier way to open recording files: you can drag and drop recording files into the recording file bar at the top of the console.
- Improved error handling:

If the *Replay* or *Record* Service reports an error, you will see a notification on the console: "\* Errors found, please check log." Click on the message to open the log file. Look for lines that start with "WARN" or "ERR" for issues reported by the services.

---

1. To support compatibility with 4.2e, please see the *RTI Data Distribution Service Release Notes* for 4.5b.

## 5 What's Fixed in 4.5e Rev. 01

### 5.1 Issues with Recording

#### 5.1.1 File Rollover Feature Did Not Work Properly

The 'roll-over' feature, which prevents data files from growing in size, did not work properly after a few iterations of overwriting the file. This problem has been resolved.

[RTI Bug # 14150]

#### 5.1.2 Console Crashed when Recording Monitoring Topics

*RTI Recording Console* may have crashed while recording DDS traffic that included Monitoring Topics, such as traffic from the dynamic version of *RTI Shapes Demo*. This problem has been resolved.

[RTI Bug # 14216]

### 5.2 Issues with Replay

#### 5.2.1 Problems with Topic Names with Spaces or Other Special Characters

*RTI Replay Service* would not replay recordings in which the Topic name contained spaces and other special characters.

[RTI Bug # 14171]

#### 5.2.2 Incorrect Repositioning During Replay

*RTI Replay Service* was repositioning to the time of the next sample. This caused problems in some cases with topics becoming unsynchronized. You may have seen *RTI Recording Console's* progress bar jump from side to side. This problem has been resolved; now repositioning is done to exactly the time-point specified by the user.

Repositioning is a feature that was added in 4.5e.

#### 5.2.3 Replay Failed if Database Contained Topics with No Recorded Data

*RTI Replay Service* may have ended unexpectedly if a recorded database contained topics with no recorded data. No data would have been replayed in this situation. This problem has been resolved.

[RTI Bug # 14159]

## 6 Previous Release

### 6.1 What's New in 4.5e

- ❑ This release introduces *Recording Console*, a point-and-click interface for *RTI Recording Service*. This interface significantly reduces *RTI Recording Service* configuration time and complexity, and does not require any programming. The *Recording Console* makes it easy to use *RTI 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.
- ❑ There is a new feature to replicate the database tables in each database file so that they can be converted separately. See the `<self_contained>` database property in the updated *User's Manual* (Section 2.4).
- ❑ The *Convert* tool can now process a single, self-contained database file instead of the full fileset.
- ❑ The *Replay Shell* has a new command, **goto**, which repositions an entity to a specific point in the playback range (relative to the entity's start and end times).
- ❑ The *Replay Shell's* **step** and **rate** commands now work with service and database entities.
- ❑ There is a new optional TopicGroup property, `<include_meta_column>`. When set to `FALSE`, less columns are created in the SQLite database. So this option can save space and execution time when these requirements are critical.
- ❑ The *Replay* tool has a new `<administration>` property, `<status_period>`. This property determines how often the Replay service will send a status message to the Replay shell. The default is 0, which means no status is sent.
- ❑ The *Record* tool now allows you to specify multiple Topics in a `topic_group` expression.
- ❑ Error messages regarding failed remote commands have been improved.

## 6.2 What's Fixed in 4.5e

### 6.2.1 Problems with Remote Add Command

The *Record Shell's* **add** command did not work properly. This problem has been resolved.

### 6.2.2 Remote Modify Command Removed

In the previous release, the remote **modify** command was documented in the *User's Manual* but the functionality was missing. In this release, the **modify** command has been removed.

### 6.2.3 Problems Caused by Topic Names that Contained a Period

There was a problem when trying to use the *Record* and *Replay* tools with topics that had a period (".") in their names. This problem has been resolved.

[RTI Bug # 14127]

### 6.2.4 Replay Tool used Maximum Domain ID of 99

The *Replay* tool only worked with applications that used a domain ID of 99 or less. In this release, domain IDs up to 232 are supported.

[RTI Bug # 14041]

### 6.2.5 Problems with Replay Tool's Step Command

When using the **step** command, the current playback position would go back and forth, instead of always going forward. This problem has been resolved.

### 6.2.6 Replay Shell's Pause and Resume Commands Failed

The **pause** and **resume** commands failed when invoked from *RTI Recording Console*. This problem has been resolved. (This problem only affected customers who received the Early Access Release of *RTI Recording Console*.)

## 7 Known Issues

- ❑ When using the *Convert* utility on a recording created with *RTI 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 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 Bug # 13745]



- ❑ RTI does not recommend using files that are mounted over NFS to store recorded data. *RTI Recorder* uses file-locking, which has known issues working over NFS. If file-locking is not working, *RTI Recorder* 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 Bug # 13949]
- ❑ *RTI Recorder* and *RTI Converter* (**rtireconv**) 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 *RTI Converter* (**rtireconv**) will not show the correct time.
- ❑ *RTI Converter* (**rtireconv**) 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 *RTI Converter*.
- ❑ If the *RTI Data Distribution Service* application being recorded has a keyed datatype and **DataWriterProtocolQosPolicy.disable\_inline\_keyhash** is set to TRUE (not the default), then *RTI Recorder* 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 Shell's **delete** command is currently not supported.