RTI Queuing Service

Release Notes

Version 6.0.1

Trademarks

Copy and Use Restrictions

This is an independent publication and is neither affiliated with, nor authorized, sponsored, or approved by, Microsoft Corporation.

The security features of this product include software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).

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 Supported Platforms

RTI® Queuing Service is supported on the platforms in Table 1.1 Supported Platforms. No custom platforms are supported.

Table 1.1 Supported Platforms

Platform	Operating System
Linux®	All platforms on x86/x64 CPUs listed in the *RTI Connext® DDS Core Libraries Release Notes* for the same version number, except SUSE® Linux Enterprise Server and Wind River® Linux 7.
macOS®	All platforms listed in the *RTI Connext DDS Core Libraries Release Notes* for the same version number.
Windows®	

For details on these platforms, see the *RTI Connext DDS Core Libraries Platform Notes*.

2 Compatibility

Queuing Service is built on top of, and intended for use with, *RTI Connext® DDS* with the same version number.

For backward compatibility information, if any, between 6.0.1 and previous releases, see the *Migration Guide* on the RTI Community Portal (https://community.rti.com/documentation).

3 What's New in 6.0.1

3.1 New platforms

This release adds support for the following platforms:

- macOS 10.14 (x64)
- Red Hat® Enterprise Linux 8 (x64)
- Windows 10 (x86, x64) with Visual Studio® 2019
- Windows Server 2016 (x86, x64) with Visual Studio 2019

3.2 Removed platforms

These platforms are no longer supported:

- macOS 10.11
- Windows 7
- Windows Server 2008 R2

3.3 nssm no longer shipped

RTI no longer ships the utility **nssm**, and the **-installService** and **-uninstallService** command-line options will no longer work. If you want to run *Queuing Service* as a Windows Service, you will need to download **nssm** from https://nssm.cc/download.

Chapter 4 What's Fixed in 6.0.1

4.1 Queueing Service crashed if configuration exchange between replicas timed out on startup

The startup configuration algorithm for *Queueing Service* had a bug causing a crash if the configuration exchange from a replica timed out. This problem was related to cleanup and has now been resolved.

[RTI Issue ID QUEUEING-704]

5 Previous Releases

5.1 What's New in 6.0.0

5.1.1 Support for heap memory allocation monitoring in persistent queues

Queuing Service now supports the monitoring of heap memory allocations when done in the context of persistent queues.

5.2 What’s Fixed in 6.0.0

5.2.1 Queuing Service in debug mode did not link with debug version of Distributed Logger library

The debug version of *Queuing Service* linked with the release version of *RTI Distributed Logger* rather than the debug version. This may have led to unexpected behavior, including potential segfaults. This problem has been resolved.

[RTI Issue ID QUEUEING-667]

5.2.2 Sample lifespan not working when app_ack_sample_to_producer disabled on SharedReaderQueue

Queuing Service did not expire samples when their lifespan expired, when app_ack_sample_to_producer was disabled on a SharedReaderQueue. Now, even when app_ack_sample_to_producer is disabled, *Queuing Service* expires samples (using whichever method you use to control lifespan: either the queue_qos or the Lifespan QoS for the QueueProducer).

[RTI Issue ID QUEUEING-670]

5.2.3 Repeated error messages logged when using BY_SOURCE_TIMESTAMP_DESTINATIONORDER_QOS

If you set BY_SOURCE_TIMESTAMP_DESTINATIONORDER_QOS as the **destination_order** of the *Queuing Service DataWriter*, you may have seen the following error message logged repeatedly:

```
WriterHistoryMemoryPlugin_addSample:out of order
PRESWriterHistoryDriver_addWrite:!timestamp order
PRESPsWriter_writeInternal:!timestamp order
QUEUEDequeueProcessor_write: Write returned error 3
QUEUEDequeueProcessor_processMessages: Write returned error 1.
```

This problem has been resolved; however, if the *Queuing Service DataWriter* is set to use BY_SOURCE_TIMESTAMP_DESTINATIONORDER_QOS, there is still no guarantee all messages will be delivered. Messages with timestamps older than previously sent messages will be sent to the dead letter queue and will not be delivered to queue consumers. The above message may be logged just once.

Using BY_SOURCE_TIMESTAMP_DESTINATIONORDER_QOS is not recommended when using *Queuing Service*.

[RTI Issue ID QUEUEING-673]

5.2.4 Improper queue shutdown when using WAIT_WITHOUT_REPLACEMENT policy

When attempting to delete a queue with the WAIT_WITHOUT_REPLACEMENT sample replacement policy set, *Queuing Service* may have hung or crashed. This problem may have occurred during the graceful shutdown of a queue triggered by a remote administration DELETE command. This problem has been resolved.

[RTI Issue ID QUEUEING-677]

5.2.5 Queuing Service could potentially crash during shutdown

Queuing Service may have crashed if a full queue became available during shutdown. This problem has been resolved.

[RTI Issue ID QUEUEING-681]

5.2.6 Queuing Service kept resending undeliverable messages

Queuing Service kept trying to send messages that couldn't be delivered due to unrecoverable errors, resulting in an endless stream of error messages such as:

```
QUEUEDequeueProcessor_write: Write returned error 3
QUEUEDequeueProcessor_processMessages: Write returned error 1
WriterHistoryMemoryPlugin_addSample:out of order
PRESWriterHistoryDriver_addWrite:!timestamp order
```

```
PRESPsWriter_writeInternal:!timestamp order.
```

Unrecoverable write error messages can be caused by a variety of reasons. For example, an unrecoverable error occurs if a message is written with a timestamp earlier than that of a previously sent message. In this case, the write operation fails and the message cannot possibly be delivered.

This problem has been resolved. Now *Queuing Service* moves the undeliverable messages to the dead letter queue. If there is no dead letter queue available, *Queuing Service* deletes the messages.

A new UndeliveredReasonKind UNRECOVERABLE_WRITE_ERROR_UNDELIVERED_REASON is now used to mark messages sent to a dead letter queue due to unrecoverable *DataWriter* write errors.

[RTI Issue ID QUEUEING-682]

5.2.7 Unexpected "Producer not found" exception when Routing Service acted as QueueProducer

When the QueueProducer was a *DataWriter* relaying samples from an original *DataWriter*, *Queuing Service* printed the following exception for every enqueued message:

```
"Producer not found"
```

One example in which this issue occurred was when a Routing Service *DataWriter* was the QueueProducer.

This issue has been fixed by increasing the verbosity level of the message.

[RTI Issue ID QUEUEING-686]

5.2.8 Substitution of XML variable did not preserve text outside of variable

If an XML element's text contained a mixed of text and a variable, the expansion of the variable removed any existing text. For example:

```
<element> This is $(MY_VAR)</element>
```

If MY_VAR is defined as 'my var value', it expanded to:

```
<element>my var value</element>
```

This problem has been resolved; any surrounding text is now also preserved.

[RTI Issue ID QUEUEING-687]

5.2.9 Queueing API: possible race condition led to failure destroying QueueReplier or QueueRequester

A race condition may have caused a failure during the destruction of a QueueReplier/Requester. The following log message (among others) may have been printed:

```
REDAWorker_enterExclusiveArea:worker rEvt083eaf7fa79 deadlock risk: cannot enter 893ffa0 of
level 20 from level 30
```

This problem occurred when an internal QueueReplier/Requester thread finished after all other application threads using the same QueueReplier/Requester finished. The problem would never happen if the QueueReplier/Requester had a listener attached.

This problem has been resolved.

[RTI Issue ID QUEUEING-688]

6 Current Limitations

The QueueProducer and QueueConsumer wrapper APIs are only supported for the Modern C++ and .NET APIs.

7 Available Documentation

Queuing Service documentation also includes:

- **Getting Started Guide** (RTI_Queuing_Service_GettingStarted.pdf)—Provides installation and startup instructions.
- **User's Manual** (RTI_Queuing_Service_UsersManual.pdf)—Describes how to configure and use *Queuing Service*.