

RTI Queuing Service

Release Notes

Version 6.0.0

© 2019 Real-Time Innovations, Inc.
All rights reserved.
Printed in U.S.A. First printing.
February 2019.

Trademarks

Real-Time Innovations, RTI, NDDS, RTI Data Distribution Service, 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.

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	1
2 Compatibility	2
3 What's Fixed in 6.0.0	
3.1 Queuing Service in debug mode did not link with debug version of Distributed Logger library ...	3
3.2 Sample lifespan not working when app_ack_sample_to_producer disabled on SharedReaderQueue	3
3.3 Repeated error messages logged when using BY_SOURCE_TIMESTAMP_ DESTINATIONORDER_QOS	3
3.4 Improper queue shutdown when using WAIT_WITHOUT_REPLACEMENT policy	4
3.5 Queuing Service could potentially crash during shutdown	4
3.6 Queuing Service kept resending undeliverable messages	4
3.7 Unexpected "Producer not found" exception when Routing Service acted as QueueProducer	5
3.8 Substitution of XML variable did not preserve text outside of variable	5
3.9 Queuing API: possible race condition led to failure destroying QueueReplier or QueueRequester	5
4 Current Limitations	6
5 Available Documentation	7

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 <i>RTI Connexxt® DDS Core Libraries Release Notes</i> for the same version number, except not supported on SUSE® Enterprise Server or Wind River® Linux 7. Not supported on ARM® CPUs.
OS X®	All platforms listed in the <i>RTI Connexxt DDS Core Libraries Release Notes</i> for the same version number.
Windows®	

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 6.0.0 and previous releases, see the *Migration Guide* on the RTI Community Portal (<https://community.rti.com/documentation>).

3 What's Fixed in 6.0.0

3.1 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 seg-faults. This problem has been resolved.

[RTI Issue ID QUEUEING-667]

3.2 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]

3.3 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]

3.4 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]

3.5 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]

3.6 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]

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

When the QueueProducer was a *DataWriter* relaying samples from an original *DataWriter*, *Queueing 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]

3.8 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]

3.9 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:

```
REDASWorker_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]

4 Current Limitations

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

5 Available Documentation

Queuing Service documentation also includes:

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