

RTI Persistence Service

Release Notes

Version 6.1.1



© 2022 Real-Time Innovations, Inc.
All rights reserved.
Printed in U.S.A. First printing.
March 2022.

Trademarks

RTI, Real-Time Innovations, Connex, NDDS, the RTI logo, 1RTI 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/>). This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Notice

Any deprecations noted in this document serve as notice under the Real-Time Innovations, Inc. Maintenance Policy #4220.

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 New in 6.1.1	
3.1 New Platform	3
3.2 Support for External Databases is Deprecated	3
3.3 Third-Party Software Upgrades	3
4 What's Fixed in 6.1.1	
4.1 Persistence Service database corrupted if system had multiple versions of a type with the same name	4
4.2 Default value of -restore option not described clearly in help menu	4
4.3 RTICdrTypeCodeUtils_type_has_external_members:!get member error when running Persistence Service	4
4.4 Unexpected "DDS_Topic_createI:!type registered" error when two Topics with same type	5
4.5 Memory leak when persisting Topics with multiple versions of a type	5
4.6 Validity check for <pool_sample_buffer_max_size> was wrong	5
4.7 TRANSIENT DataReader may not have received samples from Persistence Service configured with non-finite persistent_sample_buffer_max_size	6
4.8 Fixes Related to Vulnerabilities	6
4.8.1 Potential crash in Persistence Service due to multiple vulnerabilities in SQLite	6
4.8.2 Potential crash, leak of sensitive information, or database corruption in Persistence Service due to multiple vulnerabilities in SQLite	7
5 Previous Release	
5.1 What's New in 6.1.0	8
5.1.1 New platforms	8
5.1.2 Removed platforms	8
5.2 What's Fixed in 6.1.0	8
5.2.1 Error starting Persistence Service when restoring from existing MySQL database	8

5.2.2 Internal failure in Persistence Service did not fully shut down Persistence Service	9
5.2.3 Checkpoint period and ACK period configuration parameters may not have been applied correctly .	9
5.2.4 Potential crash, leak of sensitive information, or database corruption in Persistence Service due to multiple vulnerabilities in SQLite	9
6 Optional Database Components	10
7 Known Issues	
7.1 Coherent Changes not Propagated as Coherent Set	11
7.2 BLOBs not Supported by ODBC Storage	11
7.3 TopicQueries not Supported in PERSISTENT Mode	11
7.4 <comm_ports> not Supported when Using Real-Time WAN Transport	11
7.5 'Incorrect arguments to mysql_stmt_execute' Errors when using MySQL ODBC Driver	12
8 Available Documentation	13

1 Supported Platforms

RTI® Persistence Service is included with *RTI Connexxt® DDS*. If you choose to use it, it must be installed on top of *Connexxt DDS* with the same version number.

Persistence Service is supported on the platforms listed in [Table 1.1 Supported Platforms](#). For details on these platforms, see the *RTI Connexxt DDS Core Libraries Platform Notes*.

Note: POSIX®-compliant architectures that end with "FACE_GP" are not supported. Custom target platforms are not supported.

Table 1.1 Supported Platforms

Platforms	Description
INTEGRITY®	INTEGRITY 10.0.2 on Intel® x86 CPU with multi 5.0.6 (architecture pentiumInty10.0.2.pcx86, available on demand). Supports Transient Durability Mode only. Available as a static C library, not an executable.
Linux®	All Linux platforms on Intel x64 CPUs listed in the <i>RTI Connexxt DDS Core Libraries Release Notes</i> for the same version number. Ubuntu® 18.04 LTS on Arm v7 (architecture armv7Linux4gcc7.5.0). The following platforms were tested in PERSISTENT mode with a filesystem and MySQL 5.7: Red Hat Enterprise Linux 6.x, CentOS 6.x. All other platforms were tested in PERSISTENT mode with a filesystem only.
macOS®	All macOS platforms listed in the <i>RTI Connexxt DDS Core Libraries Release Notes</i> for the same version number. Tested in PERSISTENT mode with a filesystem only. No external database support.
Windows®	All Windows platforms listed in the <i>RTI Connexxt DDS Core Libraries Release Notes</i> for the same version number. The following platforms were tested in PERSISTENT mode with a filesystem and MySQL 5.7: Windows 8 with Visual Studio 2012 and Windows Server 2012 R2 with Visual Studio 2012. All other platforms were tested in PERSISTENT mode with a filesystem only.

2 Compatibility

When *Persistence Service* is configured in PERSISTENT mode, you may choose between storing the topic data in files or in an external relational database.

The only supported external database is MySQL. For information on the specific version supported, see [6 Optional Database Components on page 10](#).

For backward compatibility information between 6.1.1 and previous releases, see the *Migration Guide* on the RTI Community Portal (<https://community.rti.com/documentation>).

3 What's New in 6.1.1

3.1 New Platform

This release adds support for macOS 11 systems on Arm v8 CPUs (arm64Darwin20clang12.0).

3.2 Support for External Databases is Deprecated

The `<external_database>` tag has been deprecated starting in release 6.1.1. Support for external databases will be removed in a future release.

This deprecation serves as notice under the Real-Time Innovations, Inc. Maintenance Policy #4220.

3.3 Third-Party Software Upgrades

The following third-party software used by *Persistence Service* has been upgraded:

Third-Party Software	Previous Version	Current Version
SQLite@ODBC	0.9996	0.9998
SQLite	3.29.0	3.37.2

Some of these upgrades may fix potential vulnerabilities. See [4.8 Fixes Related to Vulnerabilities on page 6](#).

For information on third-party software used by *Connex DDS* products, see the "3rdPartySoftware" documents in your installation: `<NDDSHOME>/doc/manuals/connex_dds_professional/release_notes_3rdparty`.

4 What's Fixed in 6.1.1

4.1 Persistence Service database corrupted if system had multiple versions of a type with the same name

Systems with multiple versions of a type for the same *Topic* were not correctly supported if the types used the same name as the registered type name (set with `TypeSupport::register_type()`). In this case, the *Persistence Service* database became corrupted upon restart. This problem has been resolved.

[RTI Issue ID PERSISTENCE-109]

4.2 Default value of `-restore` option not described clearly in help menu

The `-restore` option described its default value in the output of `rtipersistenceservice -help` as "Use XML value." This description has been changed to: "If this option is not specified, the corresponding XML value in the `<persistent_storage>` tag is used."

[RTI Issue ID PERSISTENCE-216]

4.3 `RTICdrTypeCodeUtils_type_has_external_members!` get member error when running Persistence Service

The following error may have occurred when using *Persistence Service*:

```
RTICdrTypeCodeUtils_type_has_external_members!get member
```

The error was harmless unless you were using `ContentFilteredTopics` in some of the *DataReaders* communicating with *Persistence Service*. In that case, the evaluation of the filter in *Persistence Service* may have been wrong. Samples that should have passed the filter(s) may not have passed it and vice-versa.

To work around the problem, you could disable TypeCode as the wire representation by setting the following QoS in the applications communicating with *Persistence Service* before the *Persistence Service* database was built.

```
participant_qos.resource_limits.type_code_max_serialized_length = 0
```

This problem has been fixed.

[RTI Issue ID PERSISTENCE-219]

4.4 Unexpected "DDS_Topic_createI:!type registered" error when two Topics with same type

Persistence Service may not have been able to persist (in memory or disk) a *Topic A* with type 'T1' if the system contained other *Topics* using the same type 'T1'.

The problem only occurred when the other *Topics* were created with different versions of a type in addition to 'T1'.

When this problem occurred, you may have seen error messages like this:

```
DL Error: : [D0500|CREATE Topic|T=Topic A] DDS_Topic_createI:!type registered
```

This problem has been resolved. Now *Persistence Service* persists all *Topics* even if they use the same type.

[RTI Issue ID PERSISTENCE-225]

4.5 Memory leak when persisting Topics with multiple versions of a type

A memory leak occurred when a *Persistence Service* execution contained a `<persistent_group>` persisting one *Topic 'A'*, with multiple type versions, when the same `<persistent_group>` also persisted a different *Topic 'B'*.

The memory leak was small and could not lead to unbounded memory growth, since the leak was bound by the number of versions of a type, which cannot be UNLIMITED.

This issue has been fixed.

[RTI Issue ID PERSISTENCE-226]

4.6 Validity check for `<pool_sample_buffer_max_size>` was wrong

The validity check for `<pool_sample_buffer_max_size>` was wrong. As a result, you may not have been warned if you used invalid values (such as -2). Also, a valid value may have been flagged as invalid if the unrelated `<persistent_sample_buffer_max_size>` was invalid. This problem has been fixed.

[RTI Issue ID PERSISTENCE-233]

4.7 TRANSIENT DataReader may not have received samples from Persistence Service configured with non-finite persistent_sample_buffer_max_size

A TRANSIENT *DataReader* may not have received samples from *Persistence Service* unless <persistent_sample_buffer_max_size> was set to a finite value (default is unlimited). This issue only occurred for types containing unbounded members (sequences or strings). This problem has been resolved.

[RTI Issue ID PERSISTENCE-235]

4.8 Fixes Related to Vulnerabilities

This release fixes some potential vulnerabilities, described below.

4.8.1 Potential crash in Persistence Service due to multiple vulnerabilities in SQLite

Persistence Service had a third-party dependency on SQLite version 3.29.0. That version of SQLite is known to be affected by a number of publicly disclosed vulnerabilities.

These vulnerabilities have been fixed by upgrading to the latest stable version of SQLite, 3.37.2. See [3.3 Third-Party Software Upgrades on page 3](#).

The impact on *Persistence Service* of using the previous version varied depending on your *Persistence Service* configuration:

- With *Connex Secure* (enabling RTPS protection):
 - Exploitable through a compromised local file system containing malicious SQLite database files.
 - *Persistence Service* could crash.
 - CVSS v3.1 Score: 6.2 MEDIUM
 - CVSS v3.1 Vector: [AV:L/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H](#)
- Without *Connex Secure*:
 - Exploitable through a compromised local file system containing malicious SQLite database files.
 - Remotely exploitable through malicious RTPS messages.
 - *Persistence Service* could crash.

- CVSS v3.1 Score: 7.5 HIGH
- CVSS v3.1 Vector: [AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H](#)

[RTI Issue ID PERSISTENCE-253]

4.8.2 Potential crash, leak of sensitive information, or database corruption in Persistence Service due to multiple vulnerabilities in SQLite

This issue was fixed in 6.1.0, but not documented at that time.

In releases prior to 6.1.0, *Persistence Service* had a third-party dependency on SQLite version 3.7.2. That version of SQLite is known to be affected by a number of publicly disclosed vulnerabilities.

These vulnerabilities were fixed by upgrading to the latest stable version of SQLite, 3.29.0.

The impact on *Persistence Service* of using the previous version varied depending on your *Persistence Service* configuration:

- With *Connex Secure* (enabling RTPS protection):
 - Exploitable through a compromised local file system containing malicious SQLite database files.
 - *Persistence Service* could crash or leak sensitive information. An attacker could compromise *Persistence Service* integrity.
 - CVSS v3.1 Score: 5.9 MEDIUM
 - CVSS v3.1 Vector: [AV:L/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:L](#)
- Without *Connex Secure*:
 - Exploitable through a compromised local file system containing malicious SQLite database files.
 - Remotely exploitable through malicious RTPS messages.
 - *Persistence Service* could crash or leak sensitive information. An attacker could compromise *Persistence Service* integrity.
 - CVSS v3.1 Score: 7.3 HIGH
 - CVSS v3.1 Vector: [AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:L](#)

[RTI Issue ID PERSISTENCE-254]

5 Previous Release

5.1 What's New in 6.1.0

5.1.1 New platforms

This release adds support for the following platforms:

- macOS 10.15 (x64)
- Red Hat® Enterprise Linux 7.6 (x64)
- Ubuntu 18.04 LTS (Arm v7)
- Ubuntu 20.04 LTS (x64)

5.1.2 Removed platforms

These platforms are no longer supported:

- AIX®
- macOS 10.12
- Solaris™
- SUSE Linux Enterprise Server 11
- Ubuntu 12.04 LTS

5.2 What's Fixed in 6.1.0

5.2.1 Error starting Persistence Service when restoring from existing MySQL database

Persistence Service may not have started successfully when the command-line option **-restore** was set to 1 and the state was stored in a MySQL database.

This issue only occurred when the *DataReader* in a persistence group was configured with a **destination_order** of `DDS_BY_SOURCE_TIMESTAMP_DESTINATIONORDER_QOS`.

You would have seen an error like this:

```
PRESCstReaderCollator_new:!restore queue state
```

This problem has been resolved.

[RTI Issue ID PERSISTENCE-203]

5.2.2 Internal failure in Persistence Service did not fully shut down Persistence Service

If *Persistence Service* failed due to an internal problem, *Persistence Service* incorrectly did not fully shut down. Specifically, *Persistence Service* stopped processing newly discovered *DataWriters*, while the *Persistence Service* application remained running. This problem has been fixed. Now, after experiencing an internal failure, *Persistence Service* will fully shut down, and the *Persistence Service* application will stop running.

[RTI Issue ID PERSISTENCE-208]

5.2.3 Checkpoint period and ACK period configuration parameters may not have been applied correctly

In previous releases, it is possible that a value different than 0 for the `<writer_checkpoint_period>` and `<writer_ack_period>` in a `<persistence_group>` may not have been applied correctly.

In addition, when these values were different than 0, *Persistence Service* may have produced a segmentation fault during shutdown.

This problem has been resolved.

[RTI Issue ID PERSISTENCE-214]

5.2.4 Potential crash, leak of sensitive information, or database corruption in Persistence Service due to multiple vulnerabilities in SQLite

See [4.8.2 Potential crash, leak of sensitive information, or database corruption in Persistence Service due to multiple vulnerabilities in SQLite on page 7](#) for details.

[RTI Issue ID PERSISTENCE-254]

6 Optional Database Components

When *Persistence Service* is used in PERSISTENT mode, you can configure it to store DDS samples into a relational database, such as MySQL.

In principle, you can use any database that provides an ODBC driver, since ODBC is a standard. However, not all ODBC databases support the same feature set. Therefore, there is no guarantee that the persistent durability features will work with an arbitrary ODBC driver.

The use of MySQL requires the separate installation of the MySQL ODBC 5.1.6 (or higher) driver. For non-Windows platforms, the installation of UnixODBC 2.2.12 (or higher) is also required.

RTI has tested *Persistence Service* with MySQL 5.7 with MySQL ODBC 8.0.22 driver.

- To use MYSQL, you will need:
 - MySQL 5.7 (download from <http://www.mysql.com>)
 - MySQL ODBC 5.1.6 driver or higher (download from <http://dev.mysql.com/downloads/connector/odbc>)
 - UnixODBC 2.2.12 or higher (download from <http://www.unixodbc.org>.)

The Durable Writer History and Durable Reader State features in *Connex DDS* also use a relational database. Therefore, the installation instructions for MySQL are provided in *RTI Connex DDS Core Libraries Database Setup*.

If you need help with the download or installation process, contact **support@rti.com**.

7 Known Issues

7.1 Coherent Changes not Propagated as Coherent Set

Persistence Service will propagate the samples inside a coherent change. However, it will propagate these samples individually, not as a coherent set.

7.2 BLOBs not Supported by ODBC Storage

The ODBC storage does not support BLOBs. The maximum size for a serialized sample is 65535 bytes in MySQL.

7.3 TopicQueries not Supported in PERSISTENT Mode

Getting TopicQuery data from a *Persistence Service* instance configured to store data on disk is not currently supported.

Note: Getting TopicQuery data from a *Persistence Service* instance running in TRANSIENT (storing data in memory) mode is supported.

[RTI Issue ID PERSISTENCE-143]

7.4 <comm_ports> not Supported when Using Real-Time WAN Transport

Persistence Service can use the *RTI Real-Time WAN Transport*. However, the port configuration using <comm_ports> or the property `dds.transport.UDIPv4_WAN.builtin.comm_ports` is not currently supported by *Persistence Service*.

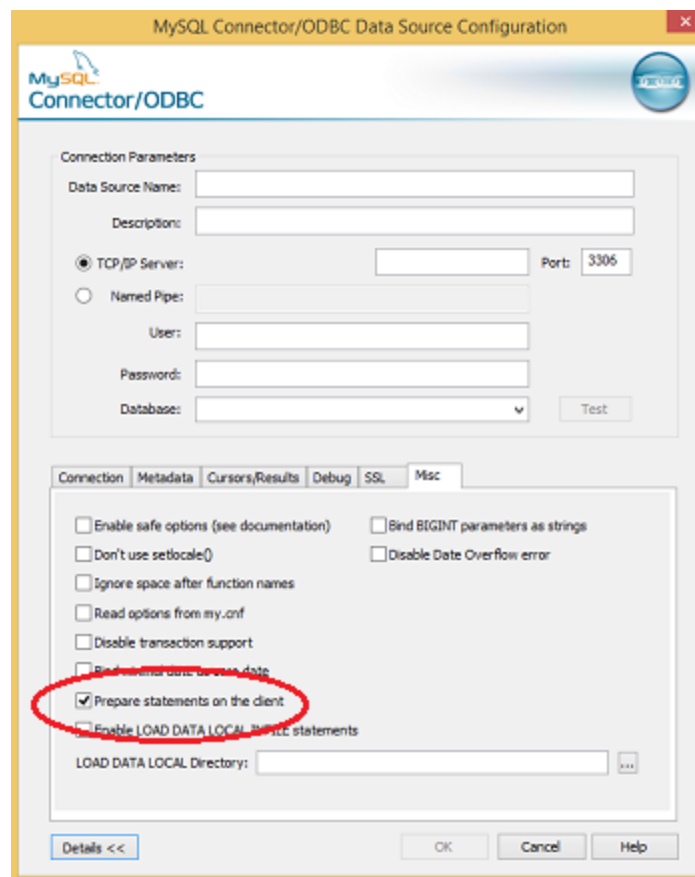
[RTI Issue ID PERSISTENCE-206]

7.5 'Incorrect arguments to mysql_stmt_execute' Errors when using MySQL ODBC Driver

Some versions of the MySQL ODBC driver may not work out-of-the-box and produce ODBC errors that include the following message:

```
Incorrect arguments to mysql_stmt_execute.
```

In this case, you will need to enable the "Prepare statements on the client" option in the DSN configuration. You will find that option under **Details, Misc, Prepare statements on the client** when adding or configuring a DSN. This behavior has been observed with MySQL ODBC driver version 8.0.23, but other versions may also be affected.



8 Available Documentation

The following documentation is provided with the *Persistence Service* distribution. (The paths show where the files are located after *Persistence Service* has been installed in <NDDSHOME>):

- General information, configuration, use cases, and execution of *Persistence Service*:
RTI Connex DDS Core Libraries User's Manual
(<NDDSHOME>/doc/manuals/connex_dds_professional/users_manual/RTI_ConnexDDS_CoreLibraries_UsersManual.pdf)

- Example code

By default, the *Persistence Service* examples are copied here:

- macOS systems:

```
/Users/your user name/rti_workspace/version/examples/persistence_service/  
<language>/hello_world_persistence
```

- Linux systems:

```
/home/your user name/rti_workspace/version/examples/persistence_service/  
<language>/hello_world_persistence
```

- Windows systems:

```
<your home directory>\rti_workspace\version\examples\persistence_service\  
<language>/hello_world_persistence
```

- Overview of persistence and durability features:
Open <NDDSHOME>/ReadMe.html, choose your desired API (C, C++, or Java), then select **Modules, RTI Connex DDS API Reference, Durability and Persistence**.