

RTI Persistence 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
2 What's New in 6.0.0	
2.1 Support for heap memory allocation monitoring in filesystem operations	3
3 What's Fixed in 6.0.0	
3.1 Persistence Service created as many Subscribers/Publishers as topics	4
3.2 Persistence Service remained locked after ungraceful termination	4
3.3 Unbounded memory growth when <allow_durable_subscriptions> set to TRUE and DataReaders with role name set were created/destroyed continuously	5
3.4 instance_replacement not applied correctly for PERSISTENT topic groups	5
4 Optional Database Components	6
5 Known Issues	
5.1 Coherent Changes not Propagated as Coherent Set	7
5.2 BLOBs not Supported by ODBC Storage	7
5.3 TopicQueries not Supported in PERSISTENT Mode	7
6 Available Documentation	8

1 Supported Platforms

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

Persistence Service is supported on the platforms listed in [Table 1.1 Supported Platforms](#). No custom platforms are supported.

Table 1.1 Supported Platforms

Platforms	Description
AIX®	AIX 7.1 platform on POWER7® CPU with XLC/C++ 12.1 (architecture 64p7AIX7.1xlc12.1). Tested with filesystem only in PERSISTENT mode. No external database support.
INTEGRITY®	INTEGRITY 10.0.2 on x86 CPU with multi 5.0.6 (architecture pentiumInty10.0.2.pcx86). Supports Transient Durability Mode only. Available as a static C library, not an executable.
Linux®	All Linux platforms on x86/x64 CPUs listed in the <i>RTI Connexx DDS Core Libraries Release Notes</i> for the same version number, except not supported on Wind River® Linux, or Linux on ARM® CPUs. On Ubuntu® 12.04 LTS, tested in PERSISTENT mode with filesystem and MySQL 5.5. On other platforms, tested with filesystem only in PERSISTENT mode.
OS® X	All OS X platforms listed in the <i>RTI Connexx DDS Core Libraries Release Notes</i> for the same version number. Tested with filesystem only in PERSISTENT mode. No external database support.
Solaris™	All Solaris platforms listed in the <i>RTI Connexx DDS Core Libraries Release Notes</i> for the same version number. Tested with filesystem only in PERSISTENT mode. Note: Solaris platforms are only available by request.
Windows®	All Windows platforms listed in the <i>RTI Connexx DDS Core Libraries Release Notes</i> for the same version number. Tested in PERSISTENT mode with filesystem and MySQL 5.5 on Windows 7 and Windows Server 2008 platforms. Tested in PERSISTENT mode with a filesystem on all other supported Windows platforms.

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 the *RTI Connex DDS Core Libraries Getting Started Guide Addendum for Database Setup*.

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

2 What's New in 6.0.0

2.1 Support for heap memory allocation monitoring in filesystem operations

Persistence Service now supports monitoring of heap memory usage in all functionality related to <filesystem> storage.

3 What's Fixed in 6.0.0

3.1 Persistence Service created as many Subscribers/Publishers as topics

Persistence Service was always creating as many Subscribers/Publishers as topics. The `<single_subscriber>` and `<single_publisher>` tags were ignored. This problem has been resolved:

- If `<single_subscriber>` or `<single_publisher>` is set to `FALSE`, *Persistence Service* will create as many Subscribers/Publishers as topics.
- If `<single_subscriber>` or `<single_publisher>` is set to `TRUE` or is absent, *Persistence Service* will share the same Publisher/Subscriber for all the topics within the same persistence group (default behavior).

For more information, refer to section "30.8.3 Sharing a Publisher/Subscriber" in the *RTI Connext DDS Core Libraries User's Manual*.

[RTI Issue ID PERSISTENCE-166]

3.2 Persistence Service remained locked after ungraceful termination

In 5.3.0, a feature was introduced that prevents multiple *Persistence Service* instances from using the same database at the same time. This feature consists of a database locking mechanism, which prevents a *Persistence Service* instance from connecting to a database if the database is already in use by another instance.

In 5.3.0, if *Persistence Service* finished ungracefully, the database remained locked and required manual intervention to unlock the database.

Starting in 6.0.0, the following changes are introduced to make this feature more friendly in the ungraceful termination scenario:

- The database locking is now disabled by default. If you want to prevent multiple instances of *Persistence Service* from accessing the same database, you now need to use the command line parameter **-enableDatabaseLocking**.
- The database locking is renewed periodically. If the locking has not been renewed within a certain period of time, the database will be considered unlocked. This way, if an ungraceful termination of the service happens, another *Persistence Service* instance can start and use the same database after the previous lock expires. The locking refresh period can be controlled with the new command-line argument (**-databaseLockRefreshPeriod**).
- The "disableDatabaseLocking" command-line argument has been deprecated.

[RTI Issue ID PERSISTENCE-168]

3.3 Unbounded memory growth when <allow_durable_subscriptions> set to TRUE and DataReaders with role name set were created/destroyed continuously

There may have been an unbounded memory growth in *Persistence Service* when:

- <allow_durable_subscriptions> was set to true in a Persistence Group. true is the default value.
- You created/deleted PERSISTENT or TRANSIENT *DataReaders* where **reader_qos.subscription_name.role_name** was set to a value other than NULL.

This problem has been fixed.

[RTI Issue ID PERSISTENCE-172]

3.4 instance_replacement not applied correctly for PERSISTENT topic groups

If you set <max_instances> to a finite value in a PERSISTENT *DataWriter*, <instance_replacement> may not have been applied correctly. That is, when <max_instances> was exceeded, the *DataWriter* may have replaced an instance that did not meet the replacement criteria defined in <instance_replacement>.

For example, if you set instance replacement to be DDS_DISPOSED_INSTANCE_REPLACEMENT, when <max_instances> was exceeded the *DataWriter* may have chosen for replacement an instance or multiple instances that were not in the DISPOSED state.

This problem has been resolved.

[RTI Issue ID PERSISTENCE-178]

4 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.

RTI has tested *Persistence Service* with MySQL 5.5 with MySQL ODBC 5.1.6.

The usage 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.

- To use MYSQL, you will need:
 - MySQL 5.5 or higher (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 the *RTI Core Libraries Getting Started Guide Addendum for Database Setup*.

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

5 Known Issues

5.1 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.

5.2 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.

5.3 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]

6 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 Connexst DDS Core Libraries User's Manual
(<NDDSHOME>/doc/manuals/connexst_dds/RTI_ConnexstDDS_CoreLibraries_User-
sManual.pdf)

- Example code

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

- Mac OS X systems:

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

- UNIX-based 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 Connexst DDS API Reference, Durability and Persistence**.