RTI Persistence Service

Release Notes

Version 5.2.0

Your systems. Working as one.

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/

Release Notes

1 Supported Platforms

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

Persistence Service is supported on the architectures listed in Table 1.1.

Table 1.1 **Supported Architectures**

Platforms	Operating System	Architecture
AIX®	AIX 5.3 *(No external database support)*	p5AIX5.3xlc9.0 64p5AIX5.3xlc9.0
	AIX 7.1 *(No external database support)*	64p7AIX7.1xlc12.1
INTEGRITY®	INTEGRITY 10.0.2 *(Supports Transient Durability Mode only. Available as a library, not an executable)*	pentiumInty10.0.2.pcx86
Linux®	CentOS 5.4, 5.5 (2.6 kernel)	i86Linux2.6gcc4.1.2 x64Linux2.6gcc4.1.2
	CentOS 6.0, 6.2 - 6.4 (2.6 kernel) *(No external database support)*	i86Linux2.6gcc4.4.5 x64Linux2.6gcc4.4.5
	Red Hat Enterprise Linux 5.0 (2.6 kernel)	i86Linux2.6gcc4.1.1 x64Linux2.6gcc4.1.1
	Red Hat Enterprise Linux 5.1, 5.2, 5.4, 5.5 (2.6 kernel)	i86Linux2.6gcc4.1.2 x64Linux2.6gcc4.1.2
	Red Hat Enterprise Linux 6.0 - 6.5 (2.6 kernel) *(No external database support)*	i86Linux2.6gcc4.4.5 x64Linux2.6gcc4.4.5
	Red Hat Enterprise Linux 7.0 (2.6 kernel) *(No external database support)*	i86Linux3gcc4.8.2 x64Linux3gcc4.8.2
	SUSE® Linux Enterprise Server 11 SP2, SP3 (2.6 kernel)	x64Linux2.6gcc4.3.4
	SUSE Linux Enterprise Server 11 SP2 (3.x kernel)	i86Linux3gcc4.3.4
	Ubuntu® Server 12.04 LTS (3.x kernel) *(No external database support)*	i86Linux3.xgcc4.6.3 x64Linux3.xgcc4.6.3
	Ubuntu 14	i86Linux3gcc4.8.2 x64Linux3gcc4.8.2

Table 1.1 **Supported Architectures**

Platforms	Operating System	Architecture
Mac OS X	OS X 10.8	x64Darwin12clang4.1
	OS X 10.10	x64Darwin12clang6.0
Solaris™	Solaris 2.10 *(No external database support)*	sparcSol2.10gcc3.4.2 sparc64Sol2.10gcc3.4.2
Windows®	All Windows platforms listed in the *RTI Connext DDS Core Libraries Release Notes* except those using Visual Studio 2008 on x64 platforms. *(No external database support on Windows 8, Windows 8.1, and Windows Server 2012 R2)*	

2 Compatibility

Persistence Service is compatible with *Connext DDS*, as well as *RTI Data Distribution Service* 4.5[b-e], 4.4d, 4.3e and 4.2e except as noted below.

❑ Prior to 5.2.0, **service_cleanup_delay** was not supported and *Persistence Service* did not purge information regarding an instance after receiving a dispose for the instance.

Starting in 5.2.0, **service_cleanup_delay** is supported. This provides a way to cause disposed instances to be immediately removed from *Persistence Service*.

- If you want disposed instances to be purged:

  Set **service_cleanup_delay** = 0 (the default) and **use_durability_service** (in the Persistence Service configuration) = 1

- If you want to keep the old behavior, so that disposed instances are not purged, there are two options:

  Set **use_durability_service** = 0 (the default)

  or

  Set **use_durability_service** = 1 and **service_cleanup_delay** = INFINITE

❑ *Persistence Service* is not compatible with applications built with *RTI Data Distribution Service* 4.5e and earlier releases when communicating over shared memory. For more information, please see the Transport Compatibility section in the *RTI Connext DDS Core Libraries Release Notes*.

❑ In *Connext DDS* 5.1.0, the default **message_size_max** for the UDPv4, UDPv6, TCP, Secure WAN, and shared-memory transports changed to provide better out-of-the-box performance. *Persistence Service* 5.1.0 also uses the new value for **message_size_max**. Consequently, *Persistence Service* 5.1.0 and higher is not out-of-the-box compatible with applications running older versions of *Connext DDS* or *RTI Data Distribution Service*. Please see the *RTI Connext DDS Core Libraries Release Notes* for instructions on how to resolve this compatibility issue with older *Connext DDS* and *RTI Data Distribution Service* applications.

❑ The types of the remote administration topics in 5.1.0 and higher are not compatible with 5.0.0, therefore:

- The 5.0.0 *Record* and *Replay* shells, *Admin Console* 5.0.0 and *Connext DDS* 5.0.0 user applications performing administration are not compatible with *Recording Service* 5.1.0 and higher.

- The 5.1.0 and higher *Record* and *Replay* shells, *Admin Console* 5.1.0 and higher, and *Connext DDS* 5.1.0 and higher user-applications performing administration are not compatible with *Recording Service* 5.0.0.

2.1 Command-Line Options Compatibility

Starting with version 4.5b, the command-line parameter **-srvName** has been replaced with **-cfgName**, which is a required parameter.

2.2 Library API Compatibility

The following fields in the RTI_PersistenceServiceProperty structure have new names (starting in 4.5d Rev. 12):

❏ **app_name** has been replaced with **application_name**

❏ **stack_size** has been replaced with **thread_stack_size**

2.3 Persistent Storage

2.3.1 ODBC 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.

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.

Persistence Service has been tested with the MySQL 5.1.44 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.

2.3.2 Storage Schema Compatibility

In *Connext DDS* 5.2.0, the schema of the information persisted into files or into an external relational database changed. Consequently, you will not be able to open *Connext DDS* 5.1.0 and earlier files and databases with *Connext DDS* 5.2.0.

2.4 Persistence Service Synchronization

Starting with version 5.0.0, the format of the **<synchronization>** tag value under **<persistence_service>** tag has changed.

Before 5.0.0, the value of the tag was a boolean indicating whether or not sample synchronization was enabled.

Starting with version 5.0.0, there are two different kinds of information that can be synchronized independently: data samples and durable subscription state. The **<synchronization>** tag value is no longer a boolean; now it is a complex value that may contain up to three new tags:

❏ **<synchronize_data>**

❏ **<synchronize_durable_subscriptions>**

❏ **<durable_subscription_synchronization_period>**

Any existing XML configuration files that use the old **<synchronization>** tag as follows:

```
<dds>
  <persistence_service>
    ...
```

```
      <synchronization>true</synchronization>
    </persistence_service>
```

must be changed to:

```
<dds>
  <persistence_service>
    ...
    <synchronization>
      <synchronize_data>true</synchronize_data>
    </synchronization>
  </persistence_service>
```

For more information on *Persistence Service* synchronization, see the *RTI Persistence Service* chapters in the R*TI Connext DDS Core Libraries User's Manual*.

3 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 5.1.44 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 5.1.44 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 *RTI Connext DDS*™ (formerly *RTI Data Distribution Service*) also use a relational database. Therefore, the installation instructions for MySQL are provided in the *RTI Core Libraries and Utilities Getting Started Guide Addendum for Database Setup*.

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

4 What's New in 5.2.0

4.1 Ability to Immediately Purge Disposed Instances from Persistence Service

This release includes the ability to purge instances from *Persistence Service*. The **service_cleanup_delay** field of the DurabilityServiceQosPolicy controls when *Persistence Service* is able to remove all information regarding a data instance. The currently supported values for **service_cleanup_delay** are zero or INFINITE. The default **service_cleanup_delay** value is 0, meaning that when an instance is disposed, it will be purged from the persistence service immediately. This will only happen if *Persistence Service* has been configured with **use_durability_service=true**. A value of INFINITE disables the purging of disposed instances.

4.2 Limited Support for Unbounded Sequences and Strings

This releases introduces limited support for unbounded sequences and strings.

Out-of-the-box, *Persistence Service* will not persist *Topics* for which the underlying type has one or more unbounded members. In order to do that, you need to change the default value of <persistence_group>/<memory_management>/<persistent_sample_buffer_max_size> from UNLIMITED to a finite value that is big enough to hold the largest sample received from the matching *DataWriters*.

4.3 New Default Value for <memory_management>/<pool_sample_ buffer_max_size> in Persistence Group

The default value for <memory_management>/<pool_sample_ buffer_max_size> in a persistence group has changed from UNLIMITED to 4096.

This change reduces the out-of-the-box memory footprint when persisting *Topics* whose types have a large maximum serialized size.

Notice that the change will not affect backward compatibility from a functional point of view, but it may affect performance by increasing the time required to persist large samples with a size greater than 4096 bytes.

5 What's Fixed in 5.2.0

5.1 Samples not Sent to DataReader for which Liveliness was Previously Lost

Persistence Service would not send samples to a *DataReader* with which it had previously lost liveliness. This may have occurred, for example, if the network connection between a *DataReader* and *Persistence Service* was lost for a duration greater than **participant_liveliness_lease_duration** (set in the *DataReader's* DiscoveryConfigQosPolicy).

This problem has been resolved.

[RTI Issue ID PERSISTENCE-87]

5.2 Long Delay Receiving Data from Persistence Service

A late-joiner *DataReader* may have taken a long time to receive all historical data from *Persistence Service*, even if there were few historical samples. For example, assume *Persistence Service* is configured to keep the last sample for each instance (last-value cache). Consider the following sequence of samples coming from the original *DataWriter*:

S1 (Instance 1), S2 (Instance 2), ..., S1000000 (Instance 2)

In this case, *Persistence Service* will keep only two samples: S1 (Instance 1) and S1000000 (Instance 2).

The problem was that when a late-joiner started up, it received S1 from *Persistence Service* immediately, but it took a while to receive S1000000. *Persistence Service* did not manage sample GAP messages efficiently. The service generated significant RTPS GAP traffic to declare that it did not have samples S2 to S999999. This problem has been resolved.

[RTI Issue ID PERSISTENCE-89]

5.3 Unexpected Timeout from DataReader's wait_for_historical_data() when using Delegated Reliability

When a *DataWriter* and matching *DataReader* were configured for delegated reliability with *Persistence Service*, the *DataReader's* **wait_for_historical_data()** operation always returned a TIMEOUT error, even if the *DataReader* had not received all historical data from *Persistence Service*. This problem has been resolved.

[RTI Issue ID PERSISTENCE-95]

5.4 Potential Segmentation Fault when Running 64-bit Persistence Service in PERSISTENT Mode

Persistence Service may have issued a segmentatiol fault when running in PERSISTENT mode on 64-bit architectures. This problem has been resolved.

[RTI Issue ID PERSISTENCE-105]

5.5 Setting <participant_id> in Persistence Service <participant> was not Supported

Setting an explicit <participant_id> for a Persistence Service <participant> was not supported. For example:

```
<persistence_service name="HelloWorldFile">
    <participant name="HelloWorldParticipant">
        <domain_id>0</domain_id>
        <participant_qos>
            <wire_protocol>
                <participant_id>0</participant_id>
            </wire_protocol>
        </participant_qos>
    </participant>
    <participant name="HelloWorldParticipant2">
        <domain_id>0</domain_id>
        <participant_qos>
            <wire_protocol>
                <participant_id>2</participant_id>
            </wire_protocol>
        </participant_qos>
    </participant>
</persistence_service>
```

If you tried to use the above example, the second <participant> creation would have failed with the following errors:

```
[D0000|ENABLE]DDS_DomainParticipantPresentation_reserve_participant_index_e
ntryports:!enable reserve participant index
[D0000|ENABLE]DDS_DomainParticipant_enableI:Participant index 0 is in use.
PLEASE SPECIFY A DIFFERENT PARTICIPANT INDEX.
PERSISTENCEServiceParticipant_initialize:!enable dds participant
PERSISTENCEServiceParticipant_new:!init PERSISTENCEServiceParticipant
object
```

This problem has been resolved.

[RTI Issue ID PERSISTENCE-108]

5.6 Potential Memory Leaks when Creation of Persistence Group Failed

When the creation of a persistence group failed, *Persistence Service* may have exited with memory leaks. This problem has been resolved.

[RTI Issue ID PERSISTENCE-111]

5.7 Potential Valgrind Memory Error When Restoring Persisted Data

When *Persistence Service* restored persisted data, valgrind may have reported this error:

```
Source and destination overlap in memcpy
```

This error, which is benign, will no longer appear.

[RTI Issue ID PERSISTENCE-112]

5.8 Unexpected Memory Growth when Persisting Keyed Topics to Disk

There was potential for unbounded memory growth when running *Persistence Service* in PERSISTENT mode and persisting keyed topics. This problem has been resolved.

[RTI Issue ID PERSISTENCE-116]

5.9 Error Restoring Multiple Persistence Groups on the Same Topic

Persistence Service only restored the first persistence group on a topic. If there were other persistence groups on the same topic, they were not restored. For example:

```
<persistence_service name="MyPersistence">
    <participant name="MyParticipant">
        <persistence_group name="MyGroup1">
            <filter>MyTopic</filter>
            <publisher_qos>
                <partition>
                    <name>
                        <element>A</element>
                    </name>
                </partition>
            </publisher_qos>
            <subscriber_qos>
                <partition>
                    <name>
                        <element>A</element>
                    </name>
                </partition>
            </subscriber_qos>
        </persistence_group>
        <persistence_group name="MyGroup2">
            <filter>MyTopic</filter>
            <publisher_qos>
                <partition>
                    <name>
                        <element>B</element>
                    </name>
                </partition>
            </publisher_qos>
            <subscriber_qos>
                <partition>
                    <name>
                        <element>B</element>
                    </name>
                </partition>
            </subscriber_qos>
        </persistence_group>
    </participant>
</persistence_service>
```

In the above example, the second persistence group on partition B was not restored when *Persistence Service* was restarted. This problem has been resolved.

[RTI Issue ID PERSISTENCE-118]

6 Known Issues

6.1 TCP Transport not Supported

Persistence Service does not support the TCP transport.

6.2 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.

6.3 BLOBs not Supported by OBDC Storage

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

7 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 on *RTI Persistence Service*

Open **<NDDSHOME>/ReadMe.html**, then select **RTI Persistence Service**.

❑ 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

Additional documentation is provided with *Connext DDS*:

❑ Configuration, use cases, and execution of *Persistence Service*:
RTI Connext DDS Core Libraries User's Manual
(**<NDDSHOME>/doc/manuals/connext_dds/ RTI_ConnextDDS_CoreLibraries_UsersManual.pdf**)

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