RTI Persistence Service

Release Notes

Version 5.1.0

Your systems. Working as one.

Printed in U.S.A. First printing.
December 2013.

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 Professional*. If you choose to use it, it must be installed on top of *RTI Connext* (formerly *RTI Data Distribution Service*) 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	Format
AIX®	AIX 5.3 *(No external database support)*	p5AIX5.3xlc9.0 64p5AIX5.3xlc9.0	Executable
	AIX 7.1 *(No external database support)*	64p7AIX7.1xlc12.1[jdk]	Executable
INTEGRITY®	INTEGRITY 10.0.2 *(Supports Transient Durability Mode only)*	pentiumInty10.0.2.pcx86	Library
Linux®	CentOS 5.4, 5.5 (2.6 kernel)	i86Linux2.6gcc4.1.2 [jdk] x64Linux2.6gcc4.1.2[jdk]	Executable
	CentOS 6.0, 6.2 - 6.4 (2.6 kernel) *(No external database support)*	i86Linux2.6gcc4.4.5[jdk] x64Linux2.6gcc4.4.5[jdk]	Executable
	Red Hat Enterprise Linux 5.0 (2.6 kernel)	i86Linux2.6gcc4.1.1[jdk] x64Linux2.6gcc4.1.1[jdk]	Executable
	Red Hat Enterprise Linux 5.1, 5.2, 5.4, 5.5 (2.6 kernel)	i86Linux2.6gcc4.1.2[jdk] x64Linux2.6gcc4.1.2[jdk]	Executable
	Red Hat Enterprise Linux 6.0 - 6.4 (2.6 kernel) *(No external database support)*	i86Linux2.6gcc4.4.5[jdk] x64Linux2.6gcc4.4.5[jdk]	Executable
	SUSE® Linux Enterprise Server 11 SP2 (2.6 kernel)	x64Linux2.6gcc4.3.4[jdk]	Executable
	SUSE Linux Enterprise Server 11 SP2 (3.x kernel)	i86Linux3gcc4.3.4[jdk]	Executable
	Ubuntu® Server 12.04 LTS (2.6 kernel) *(No external database support)*	i86Linux2.6gcc4.6.3[jdk] x64Linux2.6gcc4.6.3[jdk]	Executable
Solaris™	Solaris 2.10 *(No external database support)*	sparcSol2.10gcc3.4.2[jdk] sparc64Sol2.10gcc3.4.2[jdk]	Executable
Windows®	All Windows platforms listed in the *RTI Core Libraries and Utilities Release Notes* except those using Visual Studio 2005/2008 on x64 platforms. *(No external database support on Windows 8 and Windows Server 2012 R2)*		Executable

2 Compatibility

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

- *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* 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 is not out-of-the-box compatible with applications running older versions of *Connext* or *RTI Data Distribution Service*. Please see the *RTI Core Libraries and Utiltities Release Notes* for instructions on how to resolve this compatibility issue with older *Connext* and *RTI Data Distribution Service* applications.
- The types of the remote administration topics in 5.1.0 are not compatible with 5.0.0. Therefore:
  - The 5.0.0 *Record* and *Replay* shells, *Admin Console* 5.0.0 and *Connext* 5.0.0 user applications performing administration are not compatible with *Recording Service* 5.1.0.
  - The 5.1.0 *Record* and *Replay* shells, *Admin Console* 5.1.0 and *Connext* 5.1.0 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

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.

2.4 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 additional information on *Persistence Service* synchronization, see the *RTI Persistence Service* chapters in the *RTI Core Libraries and Utilities User's Manual*.

3 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>**):

- Installation instructions: *RTI Persistence Service Installation Guide* (**<NDDSHOME>/doc/pdf/RTI_Persistence_Service_InstallationGuide.pdf**, also available for download from the RTI Customer Portal, https://support.rti.com.)
- General information on *RTI Persistence Service:* Open **<NDDSHOME>/ReadMe.html**, then select **RTI Persistence Service**.
- Example code: **<NDDSHOME>/example/<*language*>/helloWorldPersistence**.

Additional documentation is provided with *Connext:*

- Configuration, use cases, and execution of *Persistence Service*: *RTI Connext Core Libraries and Utilities User's Manual* (**<NDDSHOME>/doc/pdf/RTI_CoreLibrariesAndUtilities_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 API Reference**, **Durability and Persistence**.

4 What's New in 5.1.0

4.1 New Platforms

This release adds support for the following platforms:

- ❏ AIX 7.1
- ❏ CentOS and Red Hat Enterprise Linux 6.2 - 6.4 (2.6 kernel)
- ❏ SUSE Linux Enterprise Server 11 SP2 (3.x kernel)
- ❏ Ubuntu® Server 12.04 LTS (2.6 kernel)
- ❏ Windows 8 (32-bit and 64-bit Editions)
- ❏ Windows Server 2012 R2 (64-bit Edition)

4.2 Support for Mutable Types

This release adds support for mutable types, as defined in the "Extensible and Dynamic Topic Types for DDS" (DDS-XTypes) specification from the Object Management Group (OMG). Starting with this release, *Persistence Service* is able to persist topics where the underlying types are mutable.

5 What's Fixed in 5.1.0

5.1 Samples Unexpectedly Removed from Durable Writer History after Restore Operation

If a *DataWriter* using Durable Writer History was restarted after a graceful or ungraceful shutdown, the samples in the *DataWriter's* history were removed unexpectedly if the following conditions were met:

- ❏ The *DataWriter* was configured with a finite lifespan
- ❏ The *DataWriter's* property **dds.data_writer.history.odbc_plugin.in_memory_state** was set to zero

This issue affected both *DataWriters* using Durable Writer History and *RTI Persistence Service*.

[RTI Issue ID CORE-5752]

5.2 Misleading Error Message if Storage Directory not Found

Persistence Service requires that the directory specified with the <directory> tag under <persistence_storage> must exist. If the directory does not exist, *Persistence Service* prints an error message. In previous releases, the error messages were misleading:

```
PERSISTENCEDatabaseManager_new:!init database manager
PERSISTENCEService_initialize:!init database manager
PERSISTENCEService_new:!init persistence service
main:!create persistence service
```

This problem has been resolved; this release prints the following messages:

```
PERSISTENCEDatabaseManager_initialize:storage directory '/tmp/ps' does not
exist
PERSISTENCEDatabaseManager_new:!init database manager
PERSISTENCEService_initialize:!init database manager
PERSISTENCEService_new:!init persistence service
```

[RTI Issue ID PERSISTENCE-62]

5.3 Error Parsing XML Configuration File with <use_wait_set> Tag

If the XML configuration file included the tag <use_wait_set>, *Persistence Service* failed with error messages such as these:

```
RTIXMLParser_onStartTag:Parse error at line 658: Unexpected tag
'use_wait_set'
RTIXMLParser_parseFromFile:Parse error in file './lib/x64Darwin10gcc4.2.1/
../../resource/xml/PsTest.xml'
DDS_XMLParser_parse_from_file:Error parsing file
PERSISTENCECfgFileManager_loadCfgFile:!error parsing XML file
PERSISTENCECfgFileManager_initialize:!load configuration file
PERSISTENCECfgFileManager_new:!init configuration file manager
PERSISTENCEService_initialize:!init configuration file manager
PERSISTENCEService_new:!init persistence service
```

This problem has been resolved.

[RTI Issue ID PERSISTENCE-63]

5.4 wait_for_historical_data() Timed Out when used in Persistent DataReader with direct_communication = FALSE

If an application called **wait_for_historical_data()** on a *DataReader* configured with PERSISTENT or TRANSIENT Durability and **durability.direct_communication** set to FALSE, the operation may have returned the error code DDS_RETCODE_TIMEOUT even if all the historical data had been received from *Persistence Service* before the operation timeout expired.

[RTI Issue ID PERSISTENCE-66]

5.5 Persistent DataReader with direct_communication = FALSE Received Samples from Original DataWriter

A *DataReader* configured with TRANSIENT or PERSISTENT Durability and **durability.direct_communication** set to FALSE received samples from a TRANSIENT or PERSISTENT *DataWriter* instead of receiving the samples only from the *Persistence Service* DataWriters.

Notice that this issue affected the bandwidth utilization but not application correctness because the *DataReader* discarded duplicate samples before providing them to the application.

[RTI Issue ID PERSISTENCE-67]

5.6 Parser Did not Fail when <administration> was Missing <domain_id>

The tag <domain_id> within <administration> is required, but failing to specify it did not generate a parsing error. This issue did not affect the execution of *Persistence Service*. If <domain_id> was not specified, the domain ID was set to the default value of 0. This problem has been resolved; now if there is no <domain_id> within <administration>, the parser will fail and show an error message.

[RTI Issue ID PERSISTENCE-68]

5.7 Persistence Service Failed to Enable Monitoring with a Custom QoS Profile

When you enable monitoring in a participant, you can choose to create it with a different QoS. For example:

```
<participant_qos>
    <property>
        <value>
         <element>
```

```
                <name>rti.monitor.library</name>
                <value>rtimonitoring</value>
            </element>
            <element>
                <name>
            rti.monitor.config.new_participant_domain_id</name>
                <value>54</value>
            </element>
            <element>
                <name>rti.monitor.config.qos_library</name>
                <value>MyMonitoringQosLibrary</value>
            </element>
            <element>
                <name>rti.monitor.config.qos_profile</name>
                <value>MyMonitoringQosProfile</value>
            </element>
           </value>
    </property>
</participant_qos>
```

Persistence Service failed to load the profile "MyMonitoringQosLibrary::MyMonitoringQosProfile" (for example, from the file **USER_QOS_PROFILES.xml** in the current directory).

This problem has been resolved and the above configuration will work in any of the *Persistence Service* participants, as long as the specified QoS profile exists in one of the standard QoS configuration files, such as **USER_QOS_PROFILES.xml**.

[RTI Issue ID PERSISTENCE-78]

5.8 Unexpected Duplicate Samples Received by DataReaders Associated with Durable Subscription

In previous releases, a *DataReader* that is member of a durable subscription with quorum *n* may have received a sample even if that sample was already received and acknowledged by *n* different *DataReader* members of the same durable subscription.

For example, consider the following scenario:

1. Start a *DataWriter*
2. Start *Persistence Service*
3. Create a Durable Subscription with name 'DUR1' and quorum 1
4. Publish same samples
5. Start a *DataReader* for 'DUR1' that receives previous samples
6. Stop the *DataReader* and start a new *DataReader* for 'DUR1'

The second *DataReader* should not receive the samples because they were received by the first *DataReader* on the same Durable Subscription. However, in previous releases, that was not the case. This problem has been resolved.

Notice that even after fixing this issue, there are still some conditions under which duplicate samples may be received. For example, in the scenario described above, the second *DataReader* may receive a duplicate if the first *DataReader* crashes or is stopped ungracefully before the Persistence Service *DataWriter* receives the sample acknowledgments.

[RTI Issue ID PERSISTENCE-79]

5.9 'Out of memory' errors when Persisting Keyed Topics and using File Storage on AIX Platforms

Running *Persistence Service* on an AIX platform using file storage and persisting keyed topics could have produced the following error:

```
error: S1000 -1 [SQLite]out of memory
```

The error was fatal and non-recoverable. This problem has been resolved.

[RTI Issue ID PERSISTENCE-80]

5.10 'Out of memory' Errors when using Durable Subscriptions and File Storage on AIX Platforms

Running *Persistence Service* on an AIX platform using file storage and having durable subscriptions enabled (the default case) could have produced the following error after new durable subscriptions were created:

```
error: S1000 -1 [SQLite]out of memory
```

The error was fatal and non-recoverable. This problem has been resolved.

[RTI Issue ID PERSISTENCE-81]

5.11 "Indicator variable required but not supplied" Error when using Durable Subscriptions or Application Acknowledgment and MySQL as External Database

When using durable subscriptions or application-level acknowledgment with *Persistence Service*, you may have seen an "Indicator variable required but not supplied" error. This error only occurred with MySQL as the external database.

```
!fetch virtual writer info - ODBC: error: 22002 0 [unixODBC][MySQL][ODBC
5.1 Driver][mysqld-5.1.44-community]Indicator variable required but not
supplied
```

This problem has been resolved.

[RTI Issue ID PERSISTENCE-82]

5.12 Data only Persisted for First Participant, Other Participants Ignored

Persistence Service only persisted data for the first participant declared in the XML configuration. Any additional participants in the XML configuration were ignored. For example, when using the following configuration, *Persistence Service* did not persist data for *DataWriters* in domain 57 because the second participant was ignored.

```
<persistence_service name="MultipleParticipantService">
    <participant name="Participant1">
        <domain_id>56</domain_id>

        <persistence_group name="Group1">
            <filter>*</filter>
        </persistence_group>
    </participant>

    <participant name="Participant2">
        <domain_id>57</domain_id>

        <persistence_group name="Group1">
            <filter>*</filter>
        </persistence_group>
    </participant>
</persistence_service>
```

This problem has been resolved.

[RTI Issue ID PERSISTENCE-84]

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.