RTI Persistence Service

Release Notes

Version 5.0.0

Your systems. Working as one.

Printed in U.S.A. First printing.
August 2012.

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 Compatibility

RTI® Persistence Service is included with *RTI Connext™ Professional Edition*. 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 compatible with *Connext*, as well as *RTI Data Distribution Service* 4.5[b-e], 4.4d, 4.3e and 4.2e; it is supported on the architectures listed in Table 1.1.

Note: *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 Core Libraries and Utilities Release Notes*.

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
INTEGRITY	INTEGRITY 10.0.0	pentiumInty10.0.0.pcx86	Library
Linux®	CentOS 5.4, 5.5 (2.6 kernel)	i86Linux2.6gcc4.1.2 x64Linux2.6gcc4.1.2	Executable
	Red Hat Enterprise Linux 5.0	i86Linux2.6gcc4.1.1 x64Linux2.6gcc4.1.1	Executable
	Red Hat Enterprise Linux 5.1, 5.2, 5.4, 5.5	i86Linux2.6gcc4.1.2 x64Linux2.6gcc4.1.2	Executable
	Red Hat Enterprise Linux 6.0, 6.1 *(no external database support)*	i86Linux2.6gcc4.4.5 x64Linux2.6gcc4.4.5	Executable
	Ubuntu® Server 10.04 (2.6 kernel) *(no external database support)*	i86Linux2.6gcc4.4.3 x64Linux2.6gcc4.4.3	Executable
Solaris™	Solaris 2.10 *(no external database support)*	sparcSol2.10gcc3.4.2 sparc64Sol2.10gcc3.4.2	Executable

Table 1.1 **Supported Architectures**

Platforms	Operating System	Architecture	Format
Windows®	Windows 7	i86Win32VS2010 x64Win64VS2010	Executable
	Windows Server® 2008 R2	x64Win64VS2010	Executable
	Windows 2000	i86Win32VS2005	Executable
	Windows 2003	i86Win32VS2005	Executable
	Windows Vista®	i86Win32VS2005 i86Win32VS2008	Executable
	Windows XP Professional	i86Win32VS2005 i86Win32VS2008	Executable

1.1 Command-Line Options Compatibility

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

1.2 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**

1.3 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.

1.4 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 in 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*.

2 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 RTI's Self Service Portal.)
- 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 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**, **Connext API Reference**, **Durability and Persistence**.

3 What's New in 5.0.0

3.1 Durable Subscriptions Support

With the Durable Subscriptions feature, you can configure *Persistence Service* to keep samples in memory or persistent storage until they are received by a set of named subscriptions.

Related XML configuration tags:

- **<durable_subscriptions>** under **<participant>**
- **<purge_samples_after_acknowledgment>** under **<persistence_service>**

- **<synchronize_durable_subscriptions>** and **<durable_ subscription_ synchronization_ period>** under **<synchronization>**
- **<allow_durable_subscriptions>** under **<persistence_group>**

For additional information on Durable Subscriptions, see the *RTI Persistence Service* documentation in the *RTI Core Libraries and Utilities User's Manual*.

3.2 Delegated Reliability Support

The *delegated reliability* feature offloads the reliability-protocol tasks (processing of NACK traffic and HB sending) from the original DataWriter to one or more *Persistence Service* instances. This feature is useful in slow consumers scenarios where a DataWriter should not slow down due to the presence of a matching slow DataReader in the system.

With delegated reliability, a RELIABLE DataWriter configured with TRANSIENT or PERSISTENT durability will communicate using BEST-EFFORT channels with matching RELIABLE DataReaders. Lost samples will be repaired by one or more instances of *Persistence Service*.

For additional information on Delegated Reliability, see the *RTI Persistence Service* documentation in the *RTI Core Libraries and Utilities User's Manual*.

3.3 Sample Logging Support

Sample logging allows you to decouple a PRSTDataReader from the corresponding PRSTDataWriter by buffering the read samples onto disk before they get persisted into the PRSTDataWriter queue. Using the sample logging feature in combination with delegated reliability addresses slow-consumers scenarios where you do not want to slow down the original DataWriter.

The sample logging feature is only supported on Linux and Windows platforms.

Related XML configuration tags:

- **<sample_logging>** under **<persistence_group>**

For additional information on using sample logging, see the *RTI Persistence Service* documentation in the *RTI Core Libraries and Utilities User's Manual*.

3.4 Performance Improvements

The following new XML tags can be used to improve the performance of *RTI Persistence Service* when it runs in PERSISTENT mode:

- **<late_joiner_read_batch>**
- **<writer_ack_period>**
- **<writer_checkpoint_period>**
- **<writer_checkpoint_volume>**

For additional information on these tags, see the *RTI Persistence Service* documentation in the *RTI Core Libraries and Utilities User's Manual*.

3.5 Waitset Support

RTI Persistence Service can be configured to use waitsets to read data from the PRSTDataReader using the new XML tag **<use_wait_set>.**

For additional information on this tag, see the *RTI Persistence Service* documentation in *RTI Core Libraries and Utilities User's Manual*.

3.6 Application-Level Acknowledgment

Starting with release 5.0.0, you can configure the PRSTDataWriters and PRSTDataReaders created by *RTI Persistence Service* to do application-level acknowledgment using the existing QoS tags **<datawriter_qos>** and **<datareader_qos>** under **<persistence_group>**.

For additional information on application-level acknowledgment, see the *RTI Core Libraries and Utilities User's Manual*.

3.7 Support for Environment Variables in XML Configuration Files

This new feature allows you to refer to an environment variable within an XML tag. When the *Connext* XML parser parses the configuration file, it will expand the environment variable. To refer to an environment variable, use the format $(MY_VARIABLE).

For example:

```
<element>
    <name>The name is $(MY_NAME)</name>
    <value>The value is $(MY_VALUE)</value>
</element>
```

3.8 Attribute 'is_default_qos' No Longer Supported

The **is_default_qos** attribute is no longer supported in *Persistence Service*. (For more information, please see Section 4.1.)

3.9 Support for Extensible Types

Persistence Service includes partial support for the "*Extensible and Dynamic Topic Types for DDS*" specification from the Object Management Group (OMG). See Section 27.13 in the *RTI Core Libraries and Utilities User's Manual* for details.

3.10 Integrated Support for Distributed Logger

The *RTI Distributed Logger* library is now included with *Persistence Service*.

When you enable the *Distributed Logger* library, *Persistence Service* will publish its log messages to a *Connext* domain. Then you can visualize the log message data with *RTI Monitor*, a separate GUI application that can run on the same host as your application or on a different host. Since the data is provided in a *Connext* topic, you can also use *rtiddsspy* or even write your own visualization tool.

For details on how to enable the *Distributed Logger* library, see the chapter on *Configuring Persistence Service* in the *RTI Core Libraries and Utilities User's Manual*, as well as the *RTI Distributed Logger Getting Started Guide*. These documents will show you how to use the new XML configuration tag, <distributed_logger>.

3.11 Propagation of Service Version as a DomainParticipant Property

In this release, the *Persistence Service* version number is propagated as a *DomainParticipant* property called "rti.service.version". The format of the value is as follows:

```
<major>.<minor>.<release>.rev<revision>
```

The version property is set in all the *DomainParticipants* created by the service.

4 What's Fixed in 5.0.0

4.1 Remote Administration Commands not Received

Persistence Service did not ignore the **is_default_qos** setting in input XML files. As a consequence, you may have seen unexpected behavior. For example, if remote administration was enabled, you may have run into a scenario where the *DataWriter* and *DataReader* created by *Persistence Service* to receive administration commands had PERSISTENT Durability. This could have occurred when running *Persistence Service* from a directory that contained a file called **USER_QOS_PROFILES.xml** with **is_default_qos** set to TRUE in a profile where the *DataWriter* and *DataReader* QoS had PERSISTENT durability. Your original intention may have been to use **USER_QOS_PROFILES.xml** to configure your application. However, because *Persistence Service* runs from the same directory as the entities that it creates, it used the application default profile as well.

To be consistent with *RTI Routing Service* and a provide a better out-of-the-box experience, the **is_default_qos** attribute is no longer supported in *Persistence Service*.

[RTI Issue ID PERSISTENCE-56]

5 Known Issues

5.1 TCP Transport not Supported

Persistence Service does not support the TCP transport.

5.2 Coherent Changes are Not Propagated as a Coherent Set

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

5.3 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.