

# **RTI Connex DDS**

**Core Libraries**

**Getting Started Guide**

**Addendum for Database Setup**

**Version 5.2.0**



© 2015 Real-Time Innovations, Inc.  
All rights reserved.  
Printed in U.S.A. First printing.  
June 2015.

## **Trademarks**

Real-Time Innovations, RTI, NDDS, RTI Data Distribution Service, DataBus, Connex, Micro DDS, 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.

## **Technical Support**

Real-Time Innovations, Inc.

232 E. Java Drive

Sunnyvale, CA 94089

Phone: (408) 990-7444

Email: [support@rti.com](mailto:support@rti.com)

Website: <https://support.rti.com/>

# Chapter 1 Database Setup

In most cases, *RTI® Connexxt™ DDS* does not use or require a database. However, if you want to use some of the durability features of Connexxt DDS, or to use *RTI Database Integration Service*, you need a relational database, such as MySQL. This section describes how to install and configure a database for use with Connexxt DDS.

- **Durable Writer History**—This feature allows a *DataWriter* to locally persist its history cache so that it can survive shutdowns, crashes and restarts. When an application restarts, each *DataWriter* that has been configured to have durable writer history automatically loads all the data in its history cache from disk and can carry on sending data as if it had never stopped executing. To the rest of the system, it will appear as if the *DataWriter* was temporarily disconnected from the network and then reappeared.
- **Durable Reader State**—This feature allows a *DataReader* to persist its state and remember which data it has already received. When an application restarts, each *DataReader* that has been configured to have durable reader state automatically loads its state from disk and can carry on receiving data as if it had never stopped executing. Data that was received by the *DataReader* before the restart will be suppressed, so it is not sent over the network.
- **RTI Persistence Service**—*Persistence Service* provides reliable data availability. The service saves data from Connexxt DDS publishing applications in memory or permanent storage, so it can be delivered to subscribing applications that join the system at a later time—even if the publishing application has already terminated. *Persistence Service* implements the optional Persistence Profile, defined in the OMG's Data Distribution Service for Real-time Systems specification.

*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.*

**Note:** At the time of this writing, MySQL is not yet supported on Red Hat® Enterprise Linux® 6 systems.

RTI has tested *Persistence Service*, *Durable Writer History*, and *Durable Reader State* with MySQL 5.1.44 with MySQL ODBC 5.1.6.

To use MySQL as the database, you need all of the following:

- MySQL 5.1.44 or higher

The installation of MySQL is beyond the scope of this document. Please refer to the MySQL Reference Manual for the process to install and configure MySQL.

- MySQL ODBC 5.1.6 driver or higher

The driver is not bundled with the MySQL server and must be installed separately.

The ODBC connector can be downloaded from:

<http://dev.mysql.com/downloads/connector/odbc/5.1.html>

The installation guide can be found here:

<http://dev.mysql.com/doc/refman/5.1/en/connector-odbc-installation.html>

- UnixODBC 2.2.12 or higher (for non-Windows systems)

The MySQL ODBC driver requires an ODBC driver manager. We recommend using UnixODBC, a complete, free and open ODBC solution for Linux systems. The driver manager can be downloaded from <http://www.unixodbc.org>.

## 1.1 Creating a Data Source for MySQL on a Linux System

Connexx DDS uses the MySQL ODBC driver to access data sources.

The connection information for each data source is stored in an ODBC INI file. This can be a file called **.odbc.ini** in the user's home directory, or it can be configured using the environment variable ODBCINI. The stored information describes each data source in detail, specifying the driver name, a description, and any additional information the driver needs to connect to the data source.

**To create the ".odbc.ini" file with a DSN named "Example", follow these steps:**

1. Create a new text file named ".odbc.ini" in your home directory.
2. Insert these lines in the file:

```
[ODBC Data Source]
Example=MySQL Driver
[Example]
DRIVER=/usr/lib/libmyodbc5.so
Database=test
```

Make sure that DRIVER points to the valid location of the MySQL ODBC driver on you system.

## 1.2 Creating a Data Source for MySQL on a Windows System

Connex DDS uses the MySQL ODBC driver to access data sources.

The connection information for each data source is stored in the Windows registry. The stored information describes each data source in detail, specifying the driver name, a description, and any additional information the driver needs to connect to the data source.

**To add a data source named "Example", follow these steps:**

1. Open the ODBC Data Source Administrator:
  - On Windows 2000 systems: choose **Start, Control Panel, Performance and Maintenance, Administrative Tools, Data Sources (ODBC)**.
  - On Windows XP systems: choose **Start, Settings, Control Panel, Administrative Tools, Data Sources (ODBC)**.
2. Select the **System DSN** tab.
3. Click **Add**; the Create New Data Source dialog appears.
4. Select the MySQL driver from the list of drivers.
5. Click **Finish**; the MySQL ODBC Driver Configuration dialog appears.
6. Fill out the fields in the dialog.
  - Enter "Example" as the Data Source Name (DSN).
  - Enter a valid username and password.
  - Select a database (for example, "test").
  - All other fields can be left empty.
  - Click **OK**.