

RTI Database Integration Service

Release Notes

Version 5.3.1



© 2018 Real-Time Innovations, Inc.
All rights reserved.
Printed in U.S.A. First printing.
February 2018.

Trademarks

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

Third Party Copyright Notices

The Oracle® TimesTen® In-Memory Database and the Oracle® Database are products of Oracle.

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. All software and documentation (whether in hard copy or electronic form) enclosed are subject to the license agreement. The software and documentation may be used or copied only under the terms of the license agreement.

The programs in this book have been included for their instructional value. RTI does not offer any warranties or representations in respect of their fitness for a particular purpose, nor does RTI accept any liability for any loss or damage arising from their use.

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 and System Requirements.....	1-1
1.1	ODBC Driver Requirements.....	1-2
2	Compatibility	1-2
2.1	Compatibility with Unbounded Types	1-2
2.2	Compatibility with RTI Connex DDS and RTI Data Distribution Service	1-2
2.3	Compatibility with Other Versions of Database Integration Service.....	1-3
2.4	Compatibility with Other Versions of Database Integration Service.....	1-3
2.4.1	Meta-Table Compatibility	1-3
2.4.2	New Conversion of SQL BIGINT Timestamps when Published as a 'resolution_column'	1-3
2.5	Configuration Compatibility	1-3
2.5.1	Configuration File Format.....	1-3
2.5.2	Configuration File Loading.....	1-4
2.5.3	Command-Line Options.....	1-4
2.6	ODBC Driver Compatibility.....	1-4
3	What's Fixed in 5.3.1.....	1-4
3.1	Defining Types with Extensible Types Features in XML Created Subscriptions with Incorrect Type Definition	1-4
4	What's New in 5.3.0	1-5
4.1	New Name for Column that Stores JSON Data in User Table	1-5
4.2	New XML Tag to Configure Default Subscription Settings	1-5
4.3	Support for 64-byte Table Names in PostgreSQL	1-5
4.4	New Configuration Parameter to Force Autocommit Mode when Storing Samples in Database Table.....	1-5
4.5	Support for Waitsets when Storing Samples in Tables	1-5
4.6	Support for Subscribing to and Storing DDS Samples in a PostgreSQL Database.....	1-6
4.7	Timestamp Storage and Publication in MySQL and PostgreSQL.....	1-6
4.8	JSON Storage of DDS Samples in MySQL and PostgreSQL	1-6
4.9	Support for Native Heap Monitoring.....	1-6
4.10	MySQL Queue Operation to Unregister a Thread and Free Resources	1-7
5	What's Fixed in 5.3.0.....	1-7
5.1	Unexpected Error Message when Setting Delete Attribute to True in <subscriptions> Tag ...	1-7
5.2	Error Subscribing to Some Topics when TypeCode not Sent.....	1-7
5.3	Error Parsing Configuration Files Containing Types with XTypes Features	1-8
5.4	Support for Invalid UTF-8 Strings in JSON Mode.....	1-8
6	Known Issues.....	1-8

6.1	No Support for Unbounded Types	1-8
6.2	Manual Table Creation does not Trigger Daemon to Create Publication/Subscription when Typecode Unknown1-9	
6.3	WCHAR and WVARCHAR Not Supported as Primary Keys for MySQL	1-9
6.4	WCHAR and WVARCHAR Not Published Correctly for MySQL	1-9
6.5	IdentifierSeparatorChar Cannot Be '.' for MySQL	1-9
6.6	IdentifierSeparatorChar Cannot Be '.' for TimesTen Cache Connect to Oracle.....	1-9
6.7	Applications with Disabled Inline-Keyhash.....	1-9
6.8	Table Initialization in Database Replication Scenarios may Require Keeping Copy of Table in Memory for Oracle and MySQL.....	1-9
6.9	Type Code Limitation for SQL Server.....	1-9
6.10	No Support for NCHAR Type for SQL Server.....	1-10
6.11	Change Tracking Retention Period for SQL Server not Configurable	1-10
6.12	Publishing Database Changes for a Table Cannot be Re-enabled for SQL Server.....	1-10
6.13	Cannot Run as Windows Service over Shared Memory.....	1-10
6.14	Types wchar, wstring, and long double not Supported with PostgreSQL Database.....	1-10
6.15	Adding New Subscriptions to RTIDDS_SUBSCRIPTION Table at Run Time not Supported for PostgreSQL Database	1-10
6.16	Database Integration Service to MySQL UTF Support.....	1-10

Release Notes

1 Supported Platforms and System Requirements

RTI® Database Integration Service requires *RTI Connext® DDS* software with the same version number.

Database Integration Service supports Oracle® TimesTen™ In-Memory Database, Oracle Database, Microsoft® SQL Server™, MySQL™, and PostgreSQL®. You must have at least one of these installed; use the version shown in [Table 1.1](#).

Note: Requires Microsoft Visual C++ 2010 Redistributable Package on computers that do not have Visual C++ 2010 installed. The Redistribution Package can be obtained from the following Microsoft website:

- For x86 architectures: <http://www.microsoft.com/download/en/details.aspx?id=5555>
- For x64 architectures: <http://www.microsoft.com/download/en/details.aspx?id=14632>

Table 1.1 **Supported Platforms¹**

Operating System	CPU	Oracle TimesTen ²	Oracle Database ³	SQL Server ⁴	PostgreSQL ⁵	MySQL
CentOS 5.4, 5.5	x86	YES	YES			YES
	x64	YES	YES			YES
Red Hat® Enterprise Linux 5.0	x86	YES	YES			YES
	x64	YES	YES			YES
Red Hat Enterprise Linux 5.1, 5.2, 5.4, 5.5	x86	YES	YES			YES
	x64	YES	YES			YES
SUSE® Linux Enterprise Server 11 SP 3	x64				YES	
Ubuntu® 12.04 LTS	x86					YES
	x64					YES
Windows® 7	x86	YES	YES	YES		YES
	x64	YES	YES	YES		YES
Windows 8 (Visual Studio® 2012)	x86		YES	YES		
	x64		YES	YES		YES ⁶
Windows 8 (Visual Studio 2013)	x64					YES ⁶

Table 1.1 **Supported Platforms¹**

Operating System	CPU	Oracle TimesTen ²	Oracle Database ³	SQL Server ⁴	PostgreSQL ⁵	MySQL
Windows 10 (Visual Studio 2015)	x86					YES ⁶
	x64					
Windows Server 2012 R2 (Visual Studio 2012)	x64		YES	YES		YES ⁶
Windows Server 2012 R2 (Visual Studio 2013, Visual Studio 2015)	x64					YES ⁶
Windows Server 2016	x64					YES ⁶

1. Additional platforms not listed in this document may be supported through special development and maintenance agreements. Contact your RTI sales representative for details.

2. Tested with Oracle TimesTen 11.2.1 unless otherwise noted.

3. Tested with Oracle Database 11g R2.

4. Tested with SQL Server 2012 SP1.

5. Tested with PostgreSQL 9.5.2+ on SUSE 11 R2 systems with x64 CPUs. Only subscriptions are supported.

6. Tested with MySQL 5.7 on Windows Server 2012 only.

1.1 ODBC Driver Requirements

MySQL

The *Database Integration Service-to-MySQL* daemon requires the separate installation of the MySQL ODBC 5.1.6 (or higher) driver. For non-Windows platforms, the installation of UNIX-ODBC 2.2.12 (or higher) is also required. See [ODBC Driver Compatibility \(Section 2.6\)](#).

For non-Windows platforms, UNIX-ODBC 2.2.12 (or higher) must also be installed. See [ODBC Driver Compatibility \(Section 2.6\)](#).

PostgreSQL

The *Database Integration Service-to-PostgreSQL* daemon requires the separate installation of the PostgreSQL ODBC driver. This release of *Database Integration Service* has been tested with PostgreSQL ODBC driver version 09.05.0300.

The UnixODBC driver manager is also required. This release of *Database Integration Service* has been tested with UnixODBC driver manager version 2.3.4.

2 Compatibility

2.1 Compatibility with Unbounded Types

Database Integration Service does not support topics that contain unbounded types. Any topic containing unbounded sequences or strings will be ignored by *Database Integration Service*.

2.2 Compatibility with RTI Connex DDS and RTI Data Distribution Service

Database Integration Service is not compatible with *RTI Data Distribution Service* 4.2c and lower.

Database Integration Service 5.2.0 is compatible with *Connex DDS* 5.2.0 and lower, as well as *RTI Data Distribution Service* 4.5[b-e], 4.4, 4.3 and 4.2e, except as noted in the *RTI Connex DDS Core Libraries Release Notes* for 5.2.0.

2.3 Compatibility with Other Versions of Database Integration Service

Database Integration Service 4.5 and higher is compatible with *Real-Time Connect* 4.2 - 4.4, with the same exceptions that apply to *Connex DDS*. Therefore, *Database Integration Service* 4.5 and higher is not compatible with an older *Real-Time Connect* version if it is not compatible with the associated *Connex DDS* version.

2.4 Compatibility with Other Versions of Database Integration Service

2.4.1 Meta-Table Compatibility

In *Database Integration Service* 5.2.0.2, the table schema used by the RTIDDS_PUBLICATIONS and RTIDDS_SUBSCRIPTIONS when using MySQL has changed.

While *Database Integration Service* 5.2.0.2 can use tables created by previous versions, this functionality will not be available until the tables have been modified to match the new schema described in the *User's Manual*.

If you have an existing database created with a previous version of the service and you want to use it with 5.2.0.2, you will need to alter the schema of the RTIDDS_PUBLICATIONS and RTIDDS_SUBSCRIPTIONS as follows:

```
ALTER TABLE RTIDDS_SUBSCRIPTIONS ADD COLUMN metadata$timestamp_type
VARCHAR(20) AFTER dr$multipcast$receive_port;
ALTER TABLE RTIDDS_SUBSCRIPTIONS ADD COLUMN metadata$include_fields
VARCHAR(1000) AFTER metadata$timestamp_type;
ALTER TABLE RTIDDS_SUBSCRIPTIONS ADD COLUMN metadata$exclude_fields
VARCHAR(1000) AFTER metadata$include_fields;
```

```
ALTER TABLE RTIDDS_PUBLICATIONS ADD COLUMN metadata$timestamp_type
VARCHAR(20) AFTER dw$res_limits$max_instances;
ALTER TABLE RTIDDS_PUBLICATIONS ADD COLUMN metadata$include_fields
VARCHAR(1000) AFTER metadata$timestamp_type;
ALTER TABLE RTIDDS_PUBLICATIONS ADD COLUMN metadata$exclude_fields
VARCHAR(1000) AFTER metadata$include_fields;
```

In the above SQL statements, replace '\$' with the identifier separator character used to create the RTIDDS_PUBLICATIONS and RTIDDS_SUBSCRIPTIONS tables.

Running *Database Integration Service* 5.2.0.2 with the old database will generate the following error:

```
[DDSQLDaemonCore_checkColumnCounts,line 4103:ERROR:53] The schema of
'RTIDDS_PUBLICATIONS' is not as expected. This may happen when using a
database created by a previous Database Integration Service release. Mod-
ify the table to match the schema described in the User's Manual or drop it
restart the Database Integration Service daemon to recreate it again.
```

2.4.2 New Conversion of SQL BIGINT Timestamps when Published as a 'resolution_column'

Timestamps used during publishing must be converted from a SQL representation to seconds and nano-seconds. This conversion has changed. Specifically, the conversion from a SQL BIGINT has changed. See the description of **resolution_column** in *User's Manual*.

2.5 Configuration Compatibility

2.5.1 Configuration File Format

Starting with *Database Integration Service* 4.5b, the format of the configuration file changed from INI to XML. The deprecated format is still functional to preserve backwards compatibility. However it should not be used as it may be removed in future releases.

2.5.2 Configuration File Loading

Starting with *Database Integration Service* 4.5b, the way configuration files are loaded has changed. These are the new approaches, listed in load order:

- ❑ `<NDDSHOME>/resource/xml/NDDS_QOS_PROFILES.xml`
- ❑ Files in the environment variable `NDDS_QOS_PROFILES`
- ❑ `<working directory>/USER_QOS_PROFILES.xml`
- ❑ `<NDDSHOME>/resource/xml/RTI_REAL_TIME_CONNECT.xml`
- ❑ `<working directory>/USER_REAL_TIME_CONNECT.xml`
- ❑ File specified using the command line parameter `-cfgFile`

The following configuration loading options have been deprecated:

- ❑ `$RTIRTCHOME/resource/xml/RTI_RTC_QOS_PROFILES.xml`
- ❑ `$RTIRTCHOME/resource/ini/RTI_RTC.ini`
- ❑ File specified in the `RTIRTC_INI` environment variable (deprecated in 4.4d)
- ❑ File specified using the command line parameter `-infile`

Although the old options are still functional to preserve backwards compatibility, its usage should be avoided as they may be removed in future releases.

2.5.3 Command-Line Options

Starting with *Database Integration Service* 4.5b, the following command-line options have been deprecated:

- ❑ `-infile` (Replaced with `-cfgFile`)
- ❑ `-loglevel` (Replaced with `-verbosity`)

The deprecated options are still functional to preserve backwards compatibility. However they should not be used as they may be removed in future releases.

2.6 ODBC Driver Compatibility

Database Integration Service to MySQL links to the UnixODBC library `libodbc.so.1`. In release 2.3.1, UnixODBC changed the library version from 1 to 2. If after installing *UnixODBC Database Integration Service* cannot find `libodbc.so`, create a symlink to `libodbc.so.1` from `libodbc.so.2`.

3 What's Fixed in 5.3.1

3.1 Defining Types with Extensible Types Features in XML Created Subscriptions with Incorrect Type Definition

When creating subscriptions with a topic type defined in an XML file, and that topic type contained Extensible Types features (e.g., with optional members), the resulting subscription was created with an incompatible type. The *DataReader* was created for a type without Extensible Types extensions. This problem has been resolved.

[RTI Issue ID RTC-310]

4 What's New in 5.3.0

4.1 New Name for Column that Stores JSON Data in User Table

The name of the column that stores JSON data in a user-table has changed from "payload" to "data". This change only affects subscriptions created with `table_schema` set to JSON or JSONB.

4.2 New XML Tag to Configure Default Subscription Settings

This release adds a new XML configuration tag to a *Database Integration Service* connection that allows you to configure the default settings of the list of subscriptions defined under `<subscriptions>`.

For example:

```
<postgresql_connection>
  <subscription_default_settings>
    <table_owner>MyOwner</table_owner>
    <domain_id>56</domain_id>
    <table_schema>JSON</table_schema>
    <metadata_fields>
      <included>
        <field>ALL</field>
      </included>
    <timestamp_type>TIMESTAMP</timestamp_type>
  </metadata_fields>
</subscription_default_settings>
<subscriptions>
  <subscription>
    <table_name>MyTbl</table_name>
    <topic_name>MyTopic</topic_name>
    <type_name>MyType</type_name>
  </subscription>
</subscriptions>
</postgresql_connection>
```

In the example above, the `<subscription>` under `<subscriptions>` will use the default settings under `<subscription_default_settings>` unless it explicitly overrides them.

4.3 Support for 64-byte Table Names in PostgreSQL

This release increases the maximum table name for PostgreSQL from 30 bytes to 64 bytes. The 64 bytes limit is imposed by PostgreSQL.

4.4 New Configuration Parameter to Force Autocommit Mode when Storing Samples in Database Table

This release includes a new configuration parameter for PostgreSQL that forces autocommit when storing DDS samples in a table. The new parameter can be configured with the XML tag `<force_autocommit>` under `<postgresql_connection>`.

The existing setting `<commit_type>` is not applicable when setting `<force_autocommit>` to true.

4.5 Support for Waitsets when Storing Samples in Tables

In previous releases, *Database Integration Service* stored samples in database tables using the `DataReader`'s `on_data_available()` listener callback. This limited scalability because the thread invoking the callback was shared by all the `DataReaders` created by the service. Using a different thread per

DataReader (to increase concurrency) required changing the receive port for each DataReader, which posed a usability issue.

This release introduces support for DDS WaitSets to read and store data in the database tables. There is a new tag called `<wait_set>` under `<general_options>` which can be used to configure this behavior. When enabled, *Database Integration Service* creates a thread pool to process the samples read using the Waitset. The number of threads in this pool can be configured and this is what provides real concurrency when storing samples in multiple tables.

For more information, see the *RTI Database Integration Service User's Manual*.

4.6 Support for Subscribing to and Storing DDS Samples in a PostgreSQL Database

This release adds the ability to subscribe to DDS topics and store DDS samples in a PostgreSQL database table.

Limitations:

- ❑ This version of *Database Integration Service* does not support creating new subscriptions after startup. Subscription must be defined in the configuration XML file or added to the `RTIDDS_-SUBSCRIPTION` table before starting the *Database Integration Service* daemon.
- ❑ Publication of database changes is not supported.

4.7 Timestamp Storage and Publication in MySQL and PostgreSQL

Database Integration Service can now store the source and reception timestamps associated with DDS data samples. The timestamps may also be used as the source timestamp of published samples (only in MySQL).

This feature is only supported for MySQL, and PostgreSQL databases.

4.8 JSON Storage of DDS Samples in MySQL and PostgreSQL

This release adds the ability to store DDS data in JSON format. To select JSON as the format for DDS data, you must set the `table_schema` field in a DDS subscription to `JSON` or `JSONB`.

This feature is only supported for MySQL and PostgreSQL databases.

4.9 Support for Native Heap Monitoring

Database Integration Service incorporates a native heap memory monitor, which provides a way to analyze the heap allocations performed at the service and *RTI Connex DDS* layers. You can use heap monitoring through the command line with the following options:

- ❑ **-heapSnapshotPeriod <sec>** Enables heap monitoring and generates a heap snapshot every `<sec>`.
- ❑ **-heapSnapshotDir <dir>** Specifies the output directory where the heap monitoring snapshots are dumped. The filenames of the generated dump files have the following format:

RTI_heap_<appName>_<processId>_<index>.log

where `<appName>` is the name you assigned to the service execution through the `-appName` parameter, `<processId>` is the process ID of the service execution, and `<index>` is an integer that automatically increases each snapshot period.

For details related to the format of the snapshot files, see the *RTI Connex DDS API Reference HTML documentation* (under **Modules, RTI Connex DDS API Reference, Unsupported Utilities, NDDS_UTILITY_take_heap_snapshot**).

4.10 MySQL Queue Operation to Unregister a Thread and Free Resources

This only applies to Database Integration Service to MySQL.

In previous releases, updating a table being monitored by *Database Integration Service* from a server-scheduled event resulted in unbounded memory growth in the MySQL server. For instance, the following operation caused unbounded memory growth on the MySQL server side:

```
CREATE EVENT myevent
ON SCHEDULE EVERY 1 SECOND
DO
UPDATE mytable set message="message" where id=id;
```

The reason for this memory growth is because every time an event was triggered, the MySQL server creates a new thread on the server side to process the event. If the table was being monitored by *Database Integration Service*, this resulted in allocation of per-thread resources that were never released by MySQL server on thread finalization.

To resolve this situation, this release of *Database Integration Service* provides a new operation, **MySQLNddsQueue_unregister_thread()**, which can be invoked at the end of an scheduled event to free per-thread resources allocated by *RTI Connex DDS*. This is an example of how to execute the operation.

```
delimiter |

CREATE EVENT myevent
ON SCHEDULE EVERY 1 SECOND
DO
BEGIN
UPDATE mytable set message="message" where id=id;
SELECT MySQLNddsQueue_unregister_thread();
END |

delimiter ;
```

Note that this operation is not necessary for updates outside scheduled events.

5 What's Fixed in 5.3.0

5.1 Unexpected Error Message when Setting Delete Attribute to True in <subscriptions> Tag

When using version 5.2.3.4, you may have seen the following error message when setting the delete attribute to true in the <subscriptions> tag:

```
[DDSQLDaemonMetaTables_populatePublicationsTable,line 1318:ERROR:2048]
Error deleting table 'RTIDDS_SUBSCRIPTIONS': [unixODBC]ERROR: relation
"rtiddds_publications" does not exist;
```

This error was not fatal and the service continued running. This problem has been resolved.

[RTI Issue ID RTC-269]

5.2 Error Subscribing to Some Topics when TypeCode not Sent

Database Integration Service may have failed when subscribing to some Topics. You may have seen the following error:

```
[DDSQLDaemonMetaTables_tblInfoAssertTypeCode,line 3620:ERROR:2048:32512]
[DDSQLDaemonCore_onUpdateMetaTableEntry,line 4112:ERROR:1] Error asserting
typecode into database
```

```
[DDSQLDaemonCore_ConnectionThreadProc,line 5261:ERROR:4] Error creating
subscription associated with the table 'RTS_SD.RTI_PixelStatus'
```

The above error occurred when the publishing application did not send TypeCode on the wire but TypeObject. This could happen because the participant resource limit **participant_qos.resource_limits.type_code_max_serialized_length** was not large enough or because the Topic's Type used the following XTypes features:

- ❑ Optional members
- ❑ Mutable types

This problem has been resolved.

[RTI Issue ID RTC-294]

5.3 Error Parsing Configuration Files Containing Types with XTypes Features

Using types that were defined in the XML file with XTypes extensions caused parsing errors and the application did not start. For instance, the following types were not supported in the XML file configuration:

```
<types>
  <struct name="XTypesTestBaseType">
    <member name="aMember" type="long" id="1"/>
    <!-- Not supported: Unexpected attribute 'id' -->
  </struct>

  <struct name="XTypesTestDerivedType" baseType="XTypesTestBaseType"
    extensibility="final"> <!--Not supported: Unexpected attribute
    'baseType' and 'extensibility' -->
    <member name="requiredMember" type="string" id="2"/>
    <!-- Not supported: Unexpected attribute 'id' -->
    <member name="optionalMember" type="float" id="3" optional="true"/>
    <!-- Not supported: Unexpected attribute 'id' and 'optional' -->
  </struct>
</types>
```

This problem has been resolved.

[RTI Issue ID RTC-299]

5.4 Support for Invalid UTF-8 Strings in JSON Mode

In *Database Integration Service* version 5.2.3.8, only data samples containing valid UTF-8 strings were allowed when working in JSON mode. This problem has been resolved.

[RTI Issue ID RTC-324]

6 Known Issues

6.1 No Support for Unbounded Types

Database Integration Service does not support topics that contain unbounded types. Hence, any topic containing any unbounded sequence or string will be ignored by *Database Integration Service*.

6.2 Manual Table Creation does not Trigger Daemon to Create Publication/Subscription when Typecode Unknown

This issue only applies when using the Oracle or MySQL databases.

If an entry is inserted into the RTIDDS_PUBLICATIONS or RTIDDS_SUBSCRIPTIONS table *and* the typecode for the data type specified in the entry has not yet been discovered by the daemon, the daemon will delay the creation of the DDS publication/subscription if the table does not already exist in the database. When either the daemon discovers the typecode or the user manually creates the table, the daemon should create the corresponding DDS publication/subscription.

However, for the Oracle 10g and MySQL databases only, in the situation described above, if you create a table manually for a pending entry, the daemon is not triggered to create the corresponding publication/subscription. The workaround is to update the entry by modifying the entry in the corresponding meta-table. This will trigger the daemon to create the publication/subscription.

6.3 WCHAR and VARCHAR Not Supported as Primary Keys for MySQL

Due to a bug in MyODBC (MySQL bug# 17983), tables with a WCHAR or VARCHAR column in the primary key are not supported in conjunction with MySQL.

6.4 WCHAR and VARCHAR Not Published Correctly for MySQL

The contents of WCHAR and VARCHAR fields are not published correctly by *Database Integration Service* Publications in MySQL. Zero (0) is published instead of the correct value.

6.5 IdentifierSeparatorChar Cannot Be '.' for MySQL

When using MySQL, the IdentifierSeparatorChar cannot be '.' due to a bug in MyODBC (MySQL bug# 15547). The default IdentifierSeparatorChar for MySQL is '\$'.

6.6 IdentifierSeparatorChar Cannot Be '.' for TimesTen Cache Connect to Oracle

If your application uses IDL types that contain strings, long doubles, or hierarchical IDL types, the IdentifierSeparatorChar cannot be '.' when using TimesTen Cache Connect to Oracle. This is due to a bug in Cache Connect to Oracle that causes quoted column names containing the character '.' (such as "A.B") to be handled incorrectly.

6.7 Applications with Disabled Inline-Keyhash

If the *Connex* DDS application has a keyed data-type and has **DataWriterProtocolQosPolicy.disable_inline_keyhash** set to TRUE (not the default setting), *Database Integration Service* may misinterpret samples as being from the wrong instance or report deserialization errors.

6.8 Table Initialization in Database Replication Scenarios may Require Keeping Copy of Table in Memory for Oracle and MySQL

If *Database Integration Service* is configured in table replication mode using the INI attribute, **TableReplicationMode**, the daemon may end up keeping a copy of the whole table in memory. This may be a problem for databases such as Oracle and MySQL where tables may become quite large.

To avoid the problem, disable table initialization by setting **dw.durability.kind** in RTIDDS_PUBLICATIONS and **dw.durability.kind** in RTIDDS_SUBSCRIPTIONS to VOLATILE_DURABILITY_QOS.

6.9 Type Code Limitation for SQL Server

The maximum type-code length that can be stored in SQL Server is 8,000 bytes.

[RTI Issue ID RTC-187]

6.10 No Support for NCHAR Type for SQL Server

This release does not support the NCHAR type for use with SQL Server.

[RTI Issue ID RTC-189]

6.11 Change Tracking Retention Period for SQL Server not Configurable

Monitoring a SQL Server database table for changes and publishing them to a DDS Topic depends on the Change Tracking facility in SQL Server. The retention period is fixed at two days, with automatic cleanup enabled.

[RTI Issue ID RTC-172]

6.12 Publishing Database Changes for a Table Cannot be Re-enabled for SQL Server

If a table is added to the publications in a SQL Server database so that changes to the table are published by the *Database Integration Service* daemon and it is subsequently removed, that table may not be added back to the publications unless the *Database Integration Service* daemon is restarted. Otherwise, *Database Integration Service* will silently fail to publish changes to that table.

[RTI Issue ID RTC-190]

6.13 Cannot Run as Windows Service over Shared Memory

If you start *Database Integration Service* as a Windows service, the shared-memory transport is not supported. For details on how to configure DDS applications to use different transport settings, please see the *RTI Connext DDS Core Libraries User's Manual* (Section 8.5.7, TRANSPORT_BUILTIN QosPolicy).

[RTI Issue ID RTC-198]

6.14 Types wchar, wstring, and long double not Supported with PostgreSQL Database

Subscriptions to topics whose types contain wchar, wstring, or long double members are not supported with PostgreSQL integration.

[RTI Issue ID RTC-263]

6.15 Adding New Subscriptions to RTIDDS_SUBSCRIPTION Table at Run Time not Supported for PostgreSQL Database

This version of *Database Integration Service* does not support creating new subscriptions after startup for a PostgreSQL database. Subscription will have to be defined in the configuration XML file or added to the RTIDDS_SUBSCRIPTION table before starting the *Database Integration Service* daemon.

[RTI Issue ID RTC-264]

6.16 Database Integration Service to MySQL UTF Support

Database Integration Service does not support MySQL databases with UTF-8 character set. Consequently, communication between the database service and the *Database Integration Service* daemon (rtirtc_mysql) is not established and the updates on the publication and subscription tables are not propagated. You may detect this behavior by checking if the table RTIRTC_TBL_INFO has not been created.

You can check the current database character set with the following SQL statement:

```
SELECT DEFAULT_CHARACTER_SET_NAME, DEFAULT_COLLATION_NAME FROM INFORMATION_SCHEMA.SCHEMATA WHERE SCHEMA_NAME = '<dbname>';
```

To fix this situation you need to specify the latin1 character set. Here are two ways to do this:

Method 1

Change the default character set of the database used by *Database Integration Service*:

1. Stop the *Database Integration Service* daemon.
2. Run the following SQL statement to change the database character set:

```
ALTER DATABASE DEFAULT CHARACTER SET latin1;
```
3. Delete all the *Database Integration Service* metatables: RTIDDS_PUBLICATIONS, RTIDDS_-SUBSCRIPTIONS, RTIRTC_LOG, and RTIRTC_TBL_INFO.
4. Restart the *Database Integration Service* daemon.

Method 2

Comment out the following lines or change them to “latin1” inside the file **my.ini** in your home directory or MySQL installation directory.

```
default-character-set=utf8  
character-set-server=utf8
```

The MySQL service should be restarted after changing the values. Also, the database should be recreated after restarting:

```
drop database <dbname>;  
create database <dbname>;
```

[RTI Issue ID RTC-141]