RTI DDS Toolkit

Release Notes

Version 3.1.1

© 2021 Real-Time Innovations, Inc.
All rights reserved.
Printed in U.S.A. First printing.
July 2021.

Trademarks

Real-Time Innovations, RTI, NDDS, RTI Data Distribution Service, DataBus, Connext, 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
Website: https://support.rti.com/

Contents

Chapter 1 Supported Platforms **1**

Chapter 2 Compatibility

2.1 Versions 3.1.x Incompatible with Versions Older than 2.0.0 2

2.2 Upgrading from a Version Older than 3.1.1 2

2.2.1 Enums must have 32-bit Representation 2

2.2.2 Direct Calls through Call Library Functions No Longer Supported 2

2.2.3 ComplexTypes VIs must be Regenerated 2

2.3 Incompatible with Connext DDS 5.1 and Lower when Using UDPv6 and Shared Memory 3

2.4 Incompatible with Connext DDS 5.0 and lower 3

2.5 Incompatibility when Regenerating Complex Data Types Created with Version 2.0.0.104 3

2.6 Additional Steps when Upgrading from Versions Older than 1.2.0.90 4

2.7 Toolkit Uses String Length of 1024 4

2.8 Improved Performance when Managing Large Data 4

Chapter 3 What's Fixed in 3.1.1

3.1 Top-level Enum Data Types were not Supported in 3.1.0 5

Chapter 4 Previous Releases

4.1 What's New in 3.1.0 6

4.1.1 Support for Arrays of Clusters 6

4.1.2 Support for Arrays of Strings 6

4.1.3 Support for Non-consecutive and Negative Enum Values 6

4.1.4 Ability to Read Multiple Samples with a Single Call 6

4.1.5 Multi-dimensional Arrays Mapped as Arrays, Regardless of ForceArrayMapping Option ... 7

4.1.6 Improved ComplexType Generator Checks for Type-Definition Errors 7

4.2 What's Fixed in 3.1.0 7

4.2.1 Error 5002 when Unregistering Instance with Non-keyed String 7

4.3 What's New in 3.0.0 7

4.3.1 New Minimum Version of LabVIEW 7
4.3.2 Support for Encrypted Keys 7
4.3.3 Ability to Show RTI Connext DDS Core Log Messages 7
4.3.4 Support for Blocking Reads 8
4.3.5 Ability to Run Complex Type Generator Programmatically 8
4.3.6 Compatibility with LabVIEW Applications and Source Distributables 8
4.3.7 Sample Info Includes Message with Meaning of secs and nanosecs 8
4.4 What's Fixed in 3.0.0 8
4.4.1 Continuous errors if logger queue size was zero 8
4.4.2 Sequences sent with fewer elements than maximum size for the sequence were not read correctly ... 8
Chapter 5 Known Issues
5.1 Monitoring Library cannot be used as DomainParticipant's Base Profile when Creating Custom Secure Profile 9
5.2 No Data Received if access_scope in Publisher/Subscriber Presentation QoS is TOPIC_PRESENTATION . 9
5.3 Warning when Closing LabVIEW after Generating VIs with ComplexType Generator or DDS Generate Custom Types VI 9
5.4 Visualization Issues with LabVIEW 2016 10
5.5 Issues Using Security Settings when LabVIEW 2019 SP1 or Later is Installed 10
5.6 Built-in VIs for Reading and Writing Strings use a Fixed Size of 1024 10
Chapter 6 Additional Documentation **11**

Chapter 1 Supported Platforms

RTI® DDS Toolkit is supported on these platforms:

- Windows® Systems:
  - Windows 8.1 (32-bit and 64-bit)
  - Windows 10 (32-bit and 64-bit)
  - Windows Server 2012 R2 (64-bit)
- Real-Time Targets:
  - NI™ Linux® 3 on ARMv7 CPU (tested on cRIO-9068 target)
  - NI Linux 3 on 64-bit Intel CPU (tested on cRIO-9031 target)

You will also need:

- National Instruments® LabVIEW® 2016 or later (32-bit)
- JKI VI Package Manager 2016 or later

Chapter 2 Compatibility

2.1 Versions 3.1.x Incompatible with Versions Older than 2.0.0

DDS Toolkit 3.1.0 and 3.1.1 are not compatible with versions before 2.0.0. This is due to a change described below.

Versions 3.1.0 and higher require all Custom Types to be generated using the *RTI DDS ComplexType Generator* wizard. Versions before 2.0.0 did not have this wizard, so to provide compatibility with pre-2.0.0 versions, version 3.0.0 allowed you to make direct calls to *DDS Toolkit* functions by using Call Library Functions (CLFs). However, starting with version 3.1.0, using CLFs is no longer supported. Therefore versions 3.1.0 and 3.1.1 are not compatible with versions before 2.0.0.

2.2 Upgrading from a Version Older than 3.1.1

2.2.1 Enums must have 32-bit Representation

DDS Toolkit 3.1.1 supports non-consecutive and negative values in enums. The way an enum is represented in a Type Definition has also changed. Now it can be represented as a “Ring” (which allows DDS Toolkit to use non-consecutive and negative values) or an “Enum” control. Rings must have an I32 representation and Enums must have a U32 representation.

2.2.2 Direct Calls through Call Library Functions No Longer Supported

Versions of *DDS Toolkit* before 3.1.0 supported direct calls to the library by using Call Library Functions (CLFs) for compatibility with versions older than 2.0.0. This has been deprecated in 3.1.0. Direct calls using CLFs will not work anymore, even if the version is 2.0.0 or higher.

2.2.3 ComplexTypes VIs must be Regenerated

When updating *DDS Toolkit* from a version older than 3.1.0, the specific *Complex Types* VIs must be regenerated with the *RTI DDS ComplexType Generator*. These generated VIs will be com-

patible with the current code. Therefore, after replacing the existing custom type VIs with the generated ones, the code will work with version 3.1.0 and 3.1.1.

2.3 Incompatible with Connext DDS 5.1 and Lower when Using UDPv6 and Shared Memory

RTI Connext® DDS 5.1.0 and earlier releases used a UDPv6 locator kind that was not compliant with the value in the RTPS specification. The value used in *Connext DDS* 5.1.0 was 5 while the RTPS specification specifies a value of 2. Because of this issue, *Connext DDS* could not interoperate with other DDS vendors over UDPv6.

This problem is resolved starting with *Connext DDS* 5.2.0. Note, however, that out-the-box backward compatibility with *Connext DDS* 5.1.0 and lower, when using both the UDPv6 and SHMEM transports, is broken.

For details on how to resolve this issue, see the Connext DDS Core Libraries Release Notes for 5.2.0.

2.4 Incompatible with Connext DDS 5.0 and lower

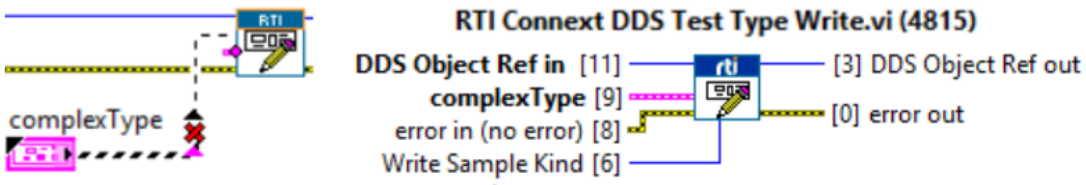
In *Connext DDS* 5.1.0, the default **message_size_max** for the UDPv4, UDPv6, TCP, Secure WAN, and shared-memory transports changed to provide better out-of-the-box performance.

DDS Toolkit 1.1.0 and higher also uses the new default value for **message_size_max** and therefore are not out-of-the-box compatible with applications running older versions of *Connext DDS* (5.0 or lower) or *RTI Data Distribution Servic*e.

For details on how to resolve this issue, see the Transport Compatibility section of the Connext DDS Core Libraries Release Notes for 5.1.0.

2.5 Incompatibility when Regenerating Complex Data Types Created with Version 2.0.0.104

The ‘complexType’ input of the previous *Write* subVI has been changed from the 2nd left pin to the 3rd one. This doesn’t affect the functionality if the *Write* subVI is not regenerated (because it is saved and used as it was originally generated). However if you regenerate the type, the VIs that were using it will no longer be able to use it until the wire is reconnected to the correct pin.



2.6 Additional Steps when Upgrading from Versions Older than 1.2.0.90

If you are upgrading from a version older than 1.2.0.90, there are important steps you must take. Briefly, changes are required because:

- The *Create Reader/Writer* subVIs have been deprecated. We strongly recommend that you upgrade the VIs to use the *Simple Create Reader/Writer* or *Advanced Create Reader/Writer*.
- The *RTI DDS Toolkit* library name changed from **lvdds.1.0.dll** to **rtilvdds.dll**.

See the *Getting Started Guide* for details on upgrading.

2.7 Toolkit Uses String Length of 1024

DDS Toolkit uses a default string length is 1024 characters. This may create incompatibilities with other DDS data types in your system that use string lengths != 1024. See Section 6.2 in the *Getting Started Guide* to learn how to change the string length.

2.8 Improved Performance when Managing Large Data

We have improved the performance when managing large data by setting the following properties (note: they cannot be changed from the QoS XML file):

Dynamic data:

- **serialization.trim_to_size** = DDS_BOOLEAN_TRUE
- **serialization.max_size_serialized** = DDS_LENGTH_UNLIMITED
- **serialization.min_size_serialized** = TypeCode's minimum serialized size

Data Writer:

- **dds.data_writer.history.memory_manager.fast_pool.pool_buffer_max_size** = 1024

Data Reader:

- **dds.data_reader.history.memory_manager.fast_pool.pool_buffer_max_size** = 1024

Chapter 3 What's Fixed in 3.1.1

3.1 Top-level Enum Data Types were not Supported in 3.1.0

In 3.1.0, DDS Enums could only be represented as Rings. Now you can use LabVIEW Enums or Rings.

LabVIEW Rings are more flexible because they allow non-consecutive and negative numeric values.

[RTI Issue ID LABPLG-880]

Chapter 4 Previous Releases

4.1 What's New in 3.1.0

4.1.1 Support for Arrays of Clusters

Arrays of clusters are now supported. They work the same way as regular arrays. All arrays must be initialized, including those in nested members or inside other arrays. See *Setting up Arrays*, Section 6.10 in the *RTI DDS Toolkit Getting Started Guide*, which shows how to initialize arrays.

4.1.2 Support for Arrays of Strings

Arrays of strings are now supported. The first string in the array specifies the length of all strings in the array (the same way as regular strings). All arrays must be initialized. See *Setting up Arrays of Strings*, Section 6.10.2 in the *RTI DDS Toolkit Getting Started Guide*, which shows how to initialize arrays of strings.

4.1.3 Support for Non-consecutive and Negative Enum Values

Non-consecutive values and negative values in enums are now supported. This requires you to use Rings instead of Enums in your LabVIEW control. LabVIEW Rings will be translated to DDS Enums. LabVIEW Enums are no longer supported.

4.1.4 Ability to Read Multiple Samples with a Single Call

The *RTI DDS ComplexType Generator* wizard generates a VI named **<Type Name> Read Array.vi**, which reads multiple samples. Its output is an array that contains the received samples. This VI includes a "Max Samples" input, which controls the maximum number of samples that can be read at a time (use -1 to read all available samples).

4.1.5 Multi-dimensional Arrays Mapped as Arrays, Regardless of ForceArrayMapping Option

If the option **ForceArrayMapping** from the Advanced Configuration Reader/Writer was false and the Type Definitions contained multi-dimensional arrays, creating a Reader/Writer failed because sequences cannot be multi-dimensional. In version 3.1.0, multi-dimensional arrays are always mapped as arrays.

4.1.6 Improved ComplexType Generator Checks for Type-Definition Errors

To avoid erroneous VIs, the *RTI DDS ComplexType Generator* wizard now checks for issues in the Type Definition. This includes checking for unlabeled members, members in the same cluster with the same name, and LabVIEW Enums. This ensures that only VIs with supported types are generated. If any of these problems are found, the wizard reports an error message, showing the problem member and cause.

4.2 What's Fixed in 3.1.0

4.2.1 Error 5002 when Unregistering Instance with Non-keyed String

Disposing or unregistering instances for data types that included a string that was not part of the key may have caused error 5002 to be thrown. This issue has been resolved.

[RTI Issue ID LABPLG-854]

4.3 What's New in 3.0.0

4.3.1 New Minimum Version of LabVIEW

The minimum version of LabVIEW supported is now 2016 (instead of 2015).

4.3.2 Support for Encrypted Keys

This release adds support for encrypted private keys on Windows and CompactRIO™ systems. If a private key is encrypted, a password must be supplied. For more information, see the *Getting Started Guide*: Section 6.8.1, Managing Custom Security Profiles with the Security Panel (Windows Systems)" (despite the section name, this feature is also supported on CompactRIO systems) and Section 6.8.2, Managing Custom Security Profiles with SubVIs.

4.3.3 Ability to Show RTI Connext DDS Core Log Messages

This release includes an option to show internal *RTI Connext DDS* core log messages. You can enable this feature in the Administration Panel.

4.3.4 Support for Blocking Reads

The *Read* VI now allows you to perform a read that can stay blocked until a timeout expires or a new sample arrives. This helps reduce CPU usage since there is no need to continuously poll for new samples.

4.3.5 Ability to Run Complex Type Generator Programmatically

The ComplexType Generator functionality now can be used programmatically. The Tools palette has a new VI called *DDS Generate Custom Type VIs*. This VI encapsulates the Complex Type Generator functionality with the same configuration parameters.

4.3.6 Compatibility with LabVIEW Applications and Source Distributables

DDS Toolkit DLL dependencies can now be packed into LabVIEW applications (.exe's) and source distributables.

4.3.7 Sample Info Includes Message with Meaning of secs and nanosecs

The 'DDS Sample Info' cluster includes a message with the meaning of *sec* and *nanosec*. This message is only explanatory, it does not affect the new or existing code at all.

4.4 What's Fixed in 3.0.0

4.4.1 Continuous errors if logger queue size was zero

Setting the internal logger queue size to 0 caused a continuous flow of error messages in the LabVIEW debugging window, such as:

```
LVDDS_Logger_log_new_message: Error when registering the message.
```

This problem has been resolved.

[RTI Issue ID LABPLG-720]

4.4.2 Sequences sent with fewer elements than maximum size for the sequence were not read correctly

When sending sequences with fewer elements than the maximum size for that sequence, a DataReader received values for *all* the elements in the sequence and the values were always default values. Now sequences are read correctly.

[RTI Issue ID LABPLG-771]

Chapter 5 Known Issues

5.1 Monitoring Library cannot be used as DomainParticipant's Base Profile when Creating Custom Secure Profile

When creating a DomainParticipant from a Custom Secure Profile, non-secure Monitoring cannot be enabled for that DomainParticipant. If this situation occurs, the toolkit will throw error 5080, which means that the DomainParticipant cannot be created.

[RTI Issue ID LABPLG-474]

5.2 No Data Received if access_scope in Publisher/Subscriber Presentation QoS is TOPIC_PRESENTATION

If you set the **access_scope** in the Presentation QoS of the Publisher/Subscriber as seen below, Readers won't be able to read data:

```
<subscriber_qos>
    <presentation>
        <access_scope>TOPIC_PRESENTATION_QOS</access_scope>
        <ordered_access>true</ordered_access>
    </presentation>
</subscriber_qos>
```

[RTI Issue ID LABPLG-567]

5.3 Warning when Closing LabVIEW after Generating VIs with ComplexType Generator or DDS Generate Custom Types VI

In some situations, you may see a warning when closing LabVIEW after generating VIs. We recommend closing LabVIEW after generating VIs.

[RTI Issue ID LABPLG-878]

5.4 Visualization Issues with LabVIEW 2016

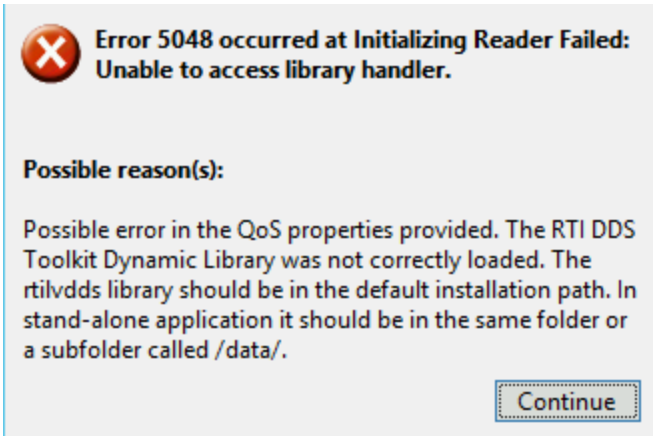
Some VIs may not be shown correctly due to an issue in LabVIEW 2016 with the `Scale and layout` settings on Windows systems. If this happens, please change the `Scale and layout` in the Windows Display Settings to 100% instead of a greater percentage.

[RTI Issue ID LABPLG-879]

5.5 Issues Using Security Settings when LabVIEW 2019 SP1 or Later is Installed

LabVIEW 2019 SP1 uses an updated version of National Instruments Secure Sockets Layer (NISSL) that is incompatible with the *DDS Toolkit's* security library. NISSL is shared among all installed versions of LabVIEW that have the same bitness. So having LabVIEW 2019 SP1 or later installed affects how security works in older versions.

This issue occurs if you are using security with *DDS Toolkit*, you close all DDS-related VIs (which unloads the *DDS Toolkit* library), and then you try to use DDS VIs again (which reloads the *DDS Toolkit* library). Then you will see this error:



[RTI Issue ID LABPLG-881]

5.6 Built-in VIs for Reading and Writing Strings use a Fixed Size of 1024

In version 3.1.0 and higher, **SimpleCreateType_WriterString.vi** and **SimpleCreateType_ReaderString.vi** create a string Data Type with a fixed length of 1024. It cannot be changed by writing the desired size, as can be done in previous versions.

[RTI Issue ID LABPLG-882]

Chapter 6 Additional Documentation

RTI DDS Toolkit uses *RTI Connext DDS* for communication. For details on *RTI Connext DDS* and the Quality of Service (QoS) settings, visit http://community.rti.com/documentation.