

RTI Ada Language Support

Release Notes

Version 6.0.1



© 2019 Real-Time Innovations, Inc.
All rights reserved.
Printed in U.S.A. First printing.
November 2019.

Trademarks

RTI, Real-Time Innovations, Connex, NDDS, 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.

This is an independent publication and is neither affiliated with, nor authorized, sponsored, or approved by, Microsoft Corporation.

The security features of this product include software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>).

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 Release Notes | 1 |
| 2 System Requirements | |
| 2.1 Supported Operating Systems | 2 |
| 2.2 Disk and Memory Usage | 2 |
| 2.3 Networking Support | 2 |
| 3 Compatibility | |
| 3.1 Wire-Protocol Compatibility | 3 |
| 3.2 Code Compatibility (Differences with Other Connexx DDS APIs) | 3 |
| 4 What's New in 6.0.1 | 5 |
| 5 Previous Release | |
| 5.1 What's New in 6.0.0 | 6 |
| 5.1.1 New platforms | 6 |
| 5.1.2 Removed platforms | 6 |
| 5.1.3 Manual user type registration no longer needed in Ada | 6 |
| 5.1.4 Support for Zero Copy transfer over shared memory in Ada | 7 |
| 5.2 What's Fixed in 6.0.0 | 7 |
| 5.2.1 Function DDS.To_DDS_Wide_String in Ada failed, raising exception | 7 |
| 5.2.2 Wrong conversion in Ada functions To_Duration and To_Duration_T | 7 |
| 5.2.3 Converting DDS.Wide_String to Standard.Wide_String failed | 8 |
| 5.2.4 DDS.WriteParams_T argument in Write_W_Params API could not be updated | 8 |
| 6 Limitations | 9 |
| 7 Known Issues | |
| 7.1 Reopening IDL Modules not Supported by rtiddsgen for Ada | 11 |
| 7.2 API Reference HTML Documentation for Ada May be Inaccurate Regarding Internal APIs | 11 |
| 7.3 API Reference HTML Documentation for Ada May Use Incorrect Entity Names | 12 |

1 Release Notes

This document provides information on *RTI® Ada Language Support* and supplements the *RTI Connex® DDS Release Notes*. All requirements, compatibility, and known issues described in the *RTI Connex DDS Core Libraries Release Notes* are also applicable to *RTI Ada Language Support*.

2 System Requirements

2.1 Supported Operating Systems

Ada Language Support provides the libraries required to build *Connex DDS* Ada applications for the following platforms when using AdaCore GNAT Pro 18.2 (<http://www.adacore.com>):

Table 2.1 Supported Platforms

| Operating System | CPU | Compiler | RTI Architecture Abbreviation |
|--|-----|-----------|-------------------------------|
| CentOS™ 7.0 | x64 | gcc 4.8.2 | x64Linux3gcc4.8.2 |
| Red Hat® Enterprise Linux® 7.0, 7.3, 7.5 | | | |
| Ubuntu® 14.04 LTS | | | |

2.2 Disk and Memory Usage

Disk usage for the combined host and target installation is approximately 350 MB.

2.3 Networking Support

Networking support is the same as described in the *RTI Connex DDS Core Libraries Release Notes*.

3 Compatibility

Below is basic compatibility information for this release.

For backward compatibility information between 6.0.1 and previous releases, see the *Migration Guide* on the RTI Community Portal (<https://community.rti.com/documentation>).

3.1 Wire-Protocol Compatibility

Ada Language Support communicates over the wire using Real-Time Publish-Subscribe (RTPS) protocol. RTPS 1.0 was introduced in 2001. The current version is 2.3. RTI plans to maintain interoperability between middleware versions based on RTPS 2.x.

Ada Language Support is compatible with *Connex DDS 5.x* and higher, and 4.5f, as well as *RTI Data Distribution Service 4.2 - 4.5*, except as noted in the *RTI Connex DDS Core Libraries Release Notes* for 5.3.1 or the *Migration Guide* for 6.0.1.

3.2 Code Compatibility (Differences with Other Connex DDS APIs)

Like the *Connex DDS Core Libraries*, *Ada Language Support* uses an API that is an extension of the DCPS layer of the OMG Data Distribution Service (DDS) standard API, version 1.4.

Ada Language Support supports the DDS Standard APIs as well as some RTI extension APIs.

API Differences:

The following features are supported in other languages, but are *not* supported in *Ada Language Support*:

- Dynamic Data
- TypeCode, TypeCodeFactory
- Custom flow controllers

- Custom content filters
- Transport Registration and configuration through the NDDS_Transport_Support API
- Request-Reply API

The above unsupported features are not part of the OMG DDS specification; they are RTI extensions.

4 What's New in 6.0.1

There are no changes to *Ada Language Support* in release 6.0.1.

5 Previous Release

5.1 What's New in 6.0.0

- This release is compatible with *Connex DDS* 6.0.1.
- This release uses GNAT Pro Ada 18.2 and GPRbuild 18.2.

5.1.1 New platforms

Table 5.1 Platforms Added in 6.0.0

| Operating System | CPU | Compiler | RTI Architecture Abbreviation |
|--|-----|-----------|-------------------------------|
| CentOS 7.0 | x64 | gcc 4.8.2 | x64Linux3gcc4.8.2 |
| Red Hat Enterprise Linux 7.0, 7.3, 7.5 | | | |
| Ubuntu 14.04 LTS | | | |

5.1.2 Removed platforms

Table 5.2 Platforms Removed in 6.0.0

| Operating System | CPU | Compiler | RTI Architecture Abbreviation |
|--|-----|-----------|-------------------------------|
| CentOS 6.0, 6.2-6.4 | x64 | gcc 4.4.5 | x64Linux2.6gcc4.4.5 |
| Red Hat Enterprise Linux® 6.0-6.5, 6.7 | | | |

5.1.3 Manual user type registration no longer needed in Ada

In previous versions of *Connex DDS*, user types had to be registered manually in *DomainParticipants* before creating a *Topic* that used them. This was achieved by using the `Register_Type` procedure in the type's generated `TypeSupport` package, which expects the *DomainParticipant* as its first argument.

Starting with this release, manual registration is no longer needed. The generated TypeSupport package body contains code that will add the user type to a list of types to be registered automatically when creating new *DomainParticipants*. This means every *DomainParticipant* created in the application will have out-of-the-box support for all user types whose TypeSupport package is included in the application by means of a "with" clause.

Note: Failing to include the TypeSupport package for a user type is unlikely because `Create_Topic` needs the type name, and the type name is obtained using `Get_Type_Name` from the TypeSupport package.

5.1.4 Support for Zero Copy transfer over shared memory in Ada

Zero Copy transfer over shared memory allows reducing the number of copies to zero for communications within the same host. This feature accomplishes zero copies by using the shared memory builtin transport to send references to samples within a shared memory segment owned by the *DataWriter*, instead of using the shared memory builtin transport to send the serialized sample content by making a copy.

With Zero Copy transfer over shared memory, there is no need for the *DataWriter* to serialize a sample, and there is no need for the *DataReader* to deserialize an incoming sample, since the sample is accessed directly on the shared memory segment created by the *DataWriter*. The new `TransferModeQosPolicy` specifies the properties of a Zero Copy *DataWriter*.

For more information, see the chapter "Sending Large Data" in the *RTI Connext DDS Core Libraries User's Manual*.

5.2 What's Fixed in 6.0.0

5.2.1 Function `DDS.To_DDS_Wide_String` in Ada failed, raising exception

When using `DDS.To_DDS_Wide_String` in Ada to convert a `Standard.Wide_String` into a `DDS.Wide_String`, a `STORAGE_ERROR` was raised and the operation failed. This problem has been resolved.

This fix creates backwards-incompatible changes. For more information, see the *Migration Guide* on the RTI Community Portal (<https://community.rti.com/documentation>).

[RTI Issue ID CORE-7388]

5.2.2 Wrong conversion in Ada functions `To_Duration` and `To_Duration_T`

The functions `To_Duration` and `To_Duration_T` interpreted the "nanosec" field in a `DDS.Duration_T` structure as microseconds instead of nanoseconds. This problem has been resolved.

[RTI Issue ID CORE-8530]

5.2.3 Converting DDS.Wide_String to Standard.Wide_String failed

When using Ada's **DDS.To_Standard_Wide_String** function to convert a `DDS.Wide_String` to a `Standard.Wide_String`, the function raised a `CONSTRAINT_ERROR` and failed to convert the string. If used in a *DataReader* listener, the internal reception thread stopped working without producing error messages and no new samples could be received. This problem has been resolved.

This fix creates backwards-incompatible changes. For more information, see the *Migration Guide* on the RTI Community Portal (<https://community.rti.com/documentation>).

[RTI Issue ID CORE-8975]

5.2.4 DDS.WriteParams_T argument in Write_W_Params API could not be updated

The third parameter of the **Write_W_Params** *DataWriter* method has changed from "in" to "in out", meaning the argument can now be modified. Previously, it could not be modified (the operation **Write_W_Params** failed).

This fix creates a backwards-incompatible API change. For more information, see the *Migration Guide* on the RTI Community Portal (<https://community.rti.com/documentation>).

[RTI Issue ID CORE-9171]

6 Limitations

- Bit fields in the IDL are not supported for Ada.
- The **Hello_dynamic** example that is available in other languages is not available in Ada.
- When using the **-example** flag in *rtiddsgen* to generate example code for Ada, the generated project file for compiling and the publisher and subscriber source files are placed under the **samples** directory, instead of at the top-level directory. Use the Ada project file under the **samples** directory to compile the example (or use the generated makefile that is located at the top level to compile).
- The code generated by *rtiddsgen* for Ada language is not in pure Ada—it will contain both C and Ada code (Ada code is a wrapper around the C code); therefore a compatible C compiler is needed to compile the generated code.
- Generated code for nested modules in Ada may not compile. *Code Generator* follows the Object Management Group (OMG) IDL-to-Ada specification in order to map modules:
 - Top level modules (i.e., those not enclosed by other modules) shall be mapped to child packages of the subsystem package, if a subsystem is specified, or root library packages otherwise. Modules nested within other modules or within subsystems shall be mapped to child packages of the corresponding package for the enclosing module or subsystem. The name of the generated package shall be mapped from the module name.
 - The generated code produced by following this specification does not compile when referencing elements from a nested module within the top-level module, as shown in the following example:

```
module Outer
{
  module Inner
  {
    struct Structure
    {
      long id;
```

```
};  
};  
  
struct Objects  
{  
    Inner::Structure nest;  
};  
};
```

This failure to compile happens because Ada does not allow a parent package to reference definitions in child packages. [RTI Issue ID CODEGENII-813]

7 Known Issues

7.1 Reopening IDL Modules not Supported by *rtiddsgen* for Ada

Reopening a module in IDL is not supported when using *rtiddsgen* for Ada. For example, the following IDL file is not supported in Ada:

```
module ModuleA {
    struct Struct1 {
        long longValue;
    };
};
.....
module ModuleA {
    struct Struct2 {
        short shortValue;
    };
};
```

[RTI Issue ID CODEGENII-231]

7.2 API Reference HTML Documentation for Ada May be Inaccurate Regarding Internal APIs

Due to the way the online documentation is generated in this release, some of the internal APIs that are not intended to be called by users (typically with filenames ending with **Low_Level.ads** or **impl.ads**, and those APIs that are lacking detailed descriptions) may also appear in the Ada online documentation and should be ignored. When in doubt, refer to the corresponding documentation for another language to determine which APIs are meant to be public.

The parameter names mentioned in the descriptions of some of the APIs may not exactly match the actual parameter names that appear in the Ada **.ads** file. However, there is usually an obvious one-to-one correspondence between the parameters as described in the descriptions compared to the APIs listed in the Ada **.ads** file.

[RTI Issue ID CORE-6290]

7.3 API Reference HTML Documentation for Ada May Use Incorrect Entity Names

Links in the Ada online documentation may display wrong entity names. For example, for DDS.DomainParticipant, the displayed name may be DomainParticipant.DDS. This only affects the links. This issue is caused by AdaCore's tool for documentation generation. RTI is investigating it with AdaCore.

[RTI Issue ID CORE-6270]