

# *RTI Ada Language Support*

## **Release Notes**

Version 5.3.1



Your systems. Working as one.



© 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.

### **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/>

# Release Notes

This document provides information on *RTI® Ada Language Support* and supplements the *RTI Connex® DDS Core Libraries 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*.

---

## 1 System Requirements

### 1.1 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 17.1 (<http://www.adacore.com>):

Table 1.1 Supported Platforms

Operating System	CPU	Compiler	RTI Architecture Abbreviation
CentOS 6.0, 6.2-6.4 (2.6 kernel)	x64	gcc 4.4.5	x64Linux2.6gcc4.4.5
Red Hat® Enterprise Linux® 6.0-6.5, 6.7 (2.6 kernel)			

### 1.2 Disk and Memory Usage

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

### 1.3 Networking Support

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

---

## 2 Compatibility

### 2.1 Wire-Protocol Compatibility

*Ada Language Support* communicates over the wire using Real-Time Publish-Subscribe (RTPS) protocol 2.1 and is compatible with *Connex DDS* 5.x 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*.

## 2.2 Code Compatibility

### 2.2.1 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 1.2.

*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.

### 2.2.2 Release-Specific Information for Connex DDS 5.3.0

- Known Compatibility Issues
  - Changes to Builtin Topics

Built-in DataWriters and DataReaders now also use the new generics. As a result, convenience methods with “W\_Key” in the their signatures, such as **Register\_Instance\_W\_Key\_W\_Params**, have been removed. The functionality still exists, however, a corresponding object needs to be passed in, as if the Built-in topics were user-generated ones. For instance: **writer\_ref.register\_instance\_w\_key (key)** becomes **writer\_ref.register\_Instance (DDS.KeyedOctets\_Of (Key))**; This object can be created using convenience functions like **DDS.KeyedOctets\_Of**. See below:

```

Procedure Foo
...
  writer_ref : ...
  handle : DDS.InstanceHandle_T := DDS.Null_InstanceHandle_T;
  key    : constant DDS.String := DDS.To_DDS_String ("my_key");
Begin
...
  writer_ref.register_instance_w_key (key) -- 5.2.3
  writer_ref.register_Instance (DDS.KeyedOctets_Of (key)) -- 5.3.0
...
End Foo;

```

- The following utility methods, which modify and manage the PropertyQoSPolicy, present in the DDS package, have been removed:
  - PropertyQosPolicyHelper\_Lookup\_Property
  - PropertyQosPolicyHelper\_Get\_Properties
  - PropertyQosPolicyHelper\_Lookup\_Property
  - PropertyQosPolicyHelper\_Assert\_Property
  - PropertyQosPolicyHelper\_Get\_Number\_Of\_Properties

In place of the above methods, the following have been added to allow you to modify and manage the PropertyQoSPolicy:

- Function Contains(PropertyQosPolicy, Standard.String) return Boolean;
  - Function Contains(PropertyQosPolicy, DDS.String) return Boolean;
  - Function Length(PropertyQosPolicy) return Natural
  - Procedure Append(PropertyQosPolicy, Standard.String, Standard.String, Boolean)
  - Procedure Append(PropertyQosPolicy, DDS.String, DDS.String, Boolean)
  - Procedure Delete (PropertyQosPolicy, Standard.String)
  - Procedure Delete (PropertyQosPolicy, DDS.String)
  - Function Get(PropertyQosPolicy, Standard.String) returns Standard.String
  - Function Get (PropertyQosPolicy, DDS.String) returns DDS.String
- Changes to DDS.KeyedOctets type structure

**In versions 5.2.3 and older:**

```
type KeyedOctets is record
  key      : DDS.String;
  length  : Integer;
  value   : System.Address;
end record with
```

**In version 5.3.0:**

```
type KeyedOctets is record
  Key     : DDS.String;
  Value  : Octets;
end record with
```

---

## 3 What's New in 5.3.1

- This release is compatible with *Connex DDS* 5.3.1.
- This release uses GNAT Pro 17.1 and GPRbuild 17.1 (same versions as 5.3.0).

---

## 4 What's Fixed in 5.3.1

### 4.1 Return Code NOT\_ALLOWED\_BY\_SEC Missing in Ada API

In previous releases, the return code NOT\_ALLOWED\_BY\_SEC was missing in the Ada API. This problem has been resolved.

[RTI Issue ID CORE-8474]

### 4.2 Incorrect Typename in Ada for Types Within a Module

In previous versions of Code Generator, the generated TypeName in Ada for types within a module was incorrect. It did not contain the module name. This problem has been resolved.

[RTI Issue ID CODEGENII-791]

## 5 Previous Release

### 5.1 What's New in 5.3.0

- This release is compatible with *Connex DDS 5.3.0*.
- This release uses GNAT Pro 17.1 (was 7.3.1) and GPRbuild 17.1 (was 2.2.1).
- *Ada Language Support* now uses Ada generics to implement user DataWriters/DataReaders created via *RTI Code Generator*. The existing functionalities remain; however, the type files created by *RTI Code Generator* will no longer include the Ada Body (\*.adb) file for a typed DataWriter/DataReader. Instead, the typed DataWriter/DataReader specification file will instantiate a corresponding generic DataWriter/DataReader.
- Certain user-generated DataWriter/DataReader methods (such as Read) have additional profiles. The main difference with these additional profiles is that certain parameters can now be passed in directly instead of requiring the use of the Ada Access Attribute.

```

procedure Read
  (Self           : not null access Ref;
   Received_Data : not null access Data_Sequences.Sequence;
   Info_Seq      : not null access DDS.SampleInfo_Seq.Sequence;
   Max_Samples   : in DDS.Long := DDS.LENGTH_UNLIMITED;
   Sample_States : in DDS.SampleStateMask := DDS.ANY_SAMPLE_STATE;
   View_States   : in DDS.ViewStateMask := DDS.ANY_VIEW_STATE;
   Instance_States : in DDS.InstanceStateMask := DDS.ANY_INSTANCE_STATE);

```

```

procedure Read
  (Self           : not null access Ref;
   Received_Data : in out Data_Sequences.Sequence;
   Info_Seq      : in out DDS.SampleInfo_Seq.Sequence;
   Max_Samples   : in DDS.Long := DDS.LENGTH_UNLIMITED;
   Sample_States : in DDS.SampleStateMask := DDS.ANY_SAMPLE_STATE;
   View_States   : in DDS.ViewStateMask := DDS.ANY_VIEW_STATE;
   Instance_States : in DDS.InstanceStateMask := DDS.ANY_INSTANCE_STATE);

```

### 5.2 What's Fixed in 5.3.0

#### 5.2.1 Memory Leak when Registering Types

During type registration of both user and builtin types, some memory allocated during type registration was not deallocated. This problem has been resolved.

[RTI Issue ID CORE-7910]

#### 5.2.2 Functions to Get Subscribers and Publishers not Working

Functions `DDS.Participant.Get_Subscribers()` and `DDS.Participant.Get_Publishers()` were not working properly. This problem has been resolved.

[RTI Issue ID CORE-7947]

#### 5.2.3 Create\_DataReader\_With\_Profile not working correctly

When `Autoenable_Created_Entities` was disabled, creation of a DataReader using `Create_DataReader_With_Profile()` would fail. This problem has been resolved.

[RTI Issue ID CORE-8012]

### 5.2.4 Potential for Incorrect Behavior when using Listener Callbacks with Multiple Type Inheritance

There was potential for Listener callbacks to not be called correctly when using a new type which implements the **DDS.DataReaderListener.Ref** interface along with other interface(s). If **DDS.DataReaderListener.Ref** was not the first interface in the list of interfaces implementing this new type, the DataReader callbacks would not function properly due to the underlying object structure. This problem has been resolved.

[RTI Issue ID CORE-8089]

## 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]

### 7.4 Method `To_DDS_Wide_String` in Ada Throws Exception

Using the method `To_DDS_Wide_String()` may cause a "STORAGE\_ERROR" exception.

[RTI Issue ID CORE-7388]