RTI Code Generator 2

for

RTI Connext DDS

Release Notes

Version 2.2.0

Your systems. Working as one.

Trademarks

Copy and Use Restrictions

Technical Support

Real-Time Innovations, Inc.
232 East Java Drive
Sunnyvale, CA 94089
Phone: (408) 990-7444
Email: support@rti.com
Website: https://support.rti.com/

Contents

Release Notes

1 Supported Platforms

You can run *Code Generator 2* as a java application or, for performance reasons, as a native application that invokes Java. See the *Code Generator 2 Getting Started Guide*.

As a java application, *Code Generator 2* is supported all host platforms (listed in the *RTI Core Libraries and Utilties Release Notes*[1]) by using the script *rtiddsgen2*.

As a native application, *Code Generator 2* is supported on the following platforms by using the script *rtiddsgen2_server*:

❏ CentOS 6.0, 6.2, 6.3, 6.4 (gcc 4.4.5)

❏ RedHat Enterprise Linux 5.0 (gcc 4.1.1)

❏ RedHat Enterprise Linux 6.0, 6.1, 6.2, 6.3, 6.4 (gcc 4.4.5)

❏ Windows 7

❏ Windows 8

❏ Windows Server 2003

❏ Windows Server 2008 R2

❏ Windows Server 2012 R2

❏ Windows Vista®

❏ Windows XP Professional

For details on these platforms, see the *RTI Core Libraries and Utilties Release Notes* for version 5.1.0.

2 What's New in 2.2.0

2.1 Ability to Convert to IDL

This version supports a new command-line option, **-convertToIDL**, which converts an input file to IDL format.

[1] This document is available from the RTI Community Portal's Documentation page.

3 What’s Fixed in 2.2.0

3.1 Input File with Unsupported Extension Caused Failure with NULL Exceptions

If the input file had an unsupported extension, *rtiddsgen* failed with a NULL exception. Now if this error occurs, you will see an error message that lists the supported file extensions.

[RTI Issue ID CODEGENII-2]

3.2 Incorrect Typecode Name when Using rtiddsgen -package

When using *rtiddsgen* with the **-package** option in Java, the typecode name included the package name. This is inconsistent with the typecode name in C++, which does not include the package prefix. Starting with this release, the package name will no longer be included in the typecode name.

This fix introduces a backward-compatibility issue with 4.5x *DataWriters;* see the Compatibility section in the *Core Libraries and Utilities Release Notes*.

[RTI Issue ID CODEGEN-28]

3.3 Period in IDL Filename Caused Compilation Errors in Generated C/C++ Code

3.4 IDL Filenames with Periods or Hyphens Caused Compilation Errors in Generated C/C++ Code

If the name of an IDL file contained a period or hyphen (such as **msg.one.idl** or **msg-one.idl**), the generated C/C++ code failed to build. This problem has been resolved.

[RTI Issue ID CODEGEN-349]

3.5 Invalid Value for max_blocking_time Tag in Generated USER_QOS_PROFILES.xml

When *rtiddsgen* is used with the **-example** command-line option, it generates an example QoS profile file called USER_QOS_PROFILES.xml. In this file, the <max_blocking_time> was set to this value:

```
<max_blocking_time>
    <sec>60</sec>
</max_blocking_time>
```

Although the intent was to set max_blocking_time to 60 seconds, the actual value was INFINITE because the XML file did not set the tag <nanosec> under <max_blocking_time> and the default value for <nanosec> is INFINITE.

This problem has been resolved by explicitly setting <nanosec> to 0, as follows:

```
<max_blocking_time>
    <sec>60</sec>
    <nanosec>0</nanosec>
</max_blocking_time>
```

[RTI Issue ID CODEGEN-566]

3.6 Error from rtiddsgen when NDDSHOME Ended with "\"—Windows Systems Only

On Windows systems, if the NDDSHOME environment variable was set to a path name that ended with a backwards slash "\", *rtiddsgen* reported a java.lang.NoClassDefFoundError error. This problem has been resolved.

[RTI Issue ID CODEGEN-600]

3.7 Pointers not Supported when Generated Code Compiled in Standalone Mode in C/C++

Pointers were not supported when generated C/C++ code was compiled in standalone mode.

For example, consider the following IDL file:

```
struct MyType {
    long * m1;
};
```

Trying to compile the generated code in standalone mode for the above type caused compilation errors in the previous release. This problem has been resolved.

[RTI Issue ID CODEGEN-606]

3.8 DataReader could Provide Samples with Invalid Values for Enumeration Fields

A *DataReader* subscribing to a topic for which the type is extensible or mutable, and where the last member is an enumeration, may have provided samples to the application in which the enumeration value was invalid. This may have occurred if a *DataWriter* published a compatible type in which the same enumeration had additional values. For example:

DataWriter type:

```
enum MyEnum {
 ENUM_1,
 ENUM_2,
 ENUM_3
};

struct MyStruct {
  MyEnum m1;
};
```

DataReader type:

```
enum MyEnum {
 ENUM_1,
 ENUM_2
};

struct MyStruct {
  MyEnum m1;
};
```

In the above example, it was possible for the *DataWriter* to send a sample where the value of **m1** was ENUM_3. When the *DataReader* received that sample, it should report a deserialization error and discard the sample because it does not recognize ENUM_3. However, due to this bug the *DataReader* assigned ENUM_3 to the enumeration value. This issue has been resolved.

[RTI Issue ID CODEGEN-622]

3.9 Serialization of Optional Members in Extensible Types Possibly Wrong in C/C++ and Java

Serialization of optional members in extensible types may have been wrong in C/C++ and Java if the member was a non-primitive member whose serialized size was greater than 65535 bytes. Consequently, this would cause *DataReaders* to fail to deserialize incoming samples.

For example:

```
struct MyStruct {
    char payload[80000]; //@Optional
}; //@Extensibility EXTENSIBLE_EXTENSIBILITY
```

Samples from the above type would not have been serialized correctly. Notice that the problem did not affect data structures marked as MUTABLE. For example:

```
struct MyStruct {
    char payload[80000]; //@Optional
}; //@Extensibility MUTABLE_EXTENSIBILITY
```

This issue has been resolved.

[RTI Issue ID CODEGEN-624]

3.10 Multidimensional Arrays of Enumerations not Supported in .NET API

Declaring multidimensional arrays of enumerations in IDL resulted in the following error when trying to send data using a .NET *Connext* application:

```
enum MyEnum {
    ENUM_1,
    ENUM_2
};

struct MyStruct {
    MyEnum m1[2][2];
};

at DDS.CdrStream.serialize_enum_array(Array elems, Int32 total_length) in
c:\ndds_head\modules\dds_dotnet.1.0\srccpp\managed\managed_cdr.cpp:line
1183 at XTypeBasePlugin.serialize(TypePluginDefaultEndpointData
endpoint_data, XTy peBase sample, CdrStream& stream, Boolean
serialize_encapsulation, UInt16 encapsulation_id, Boolean serialize_sample,
Object endpoint_plugin_qos) in c:\ndds_head\modules\nddsgen.1.0\src-
cpp\xtype\xtypeplugin.cpp:line 18633 at DDS.TypePlu-
gin`4.serialize_forwarder(Void* endpoint_data, Void* sample, RTICdrStream*
stream, Int32 serialize_encapsulation, UInt32 encapsulation_id, Int32
serialize_sample, Void* endpoint_plugin_qos) in c:\ndds_head\mod-
ules\dds_dotnet.1.0\srccpp\managed\managed_data.cpp:line 685
PRESWriterHistoryDriver_initializeSample:!serialize
WriterHistoryMemoryPlugin_addEntryToSessions:!initialize sample
WriterHistoryMemoryPlugin_getEntry:!add virtual sample to sessions
WriterHistoryMemoryPlugin_addSample:!get entry
PRESWriterHistoryDriver_addWrite:!add_sample
PRESPsWriter_writeInternal:!collator addWrite
```

This problem has been resolved.

[RTI Issue ID CODEGEN-632]

3.11 Input File with Unsupported Extension Caused Failure with NULL Exceptions

If the input file had an unsupported extension, *rtiddsgen* failed with a NULL exception. Now if this error occurs, you will see an error message that lists the supported file extensions.

[RTI Issue ID CODEGENII-2]

3.12 Using Typedef of Enum as Union Discriminator was Unsupported

When using a typedef of an enumerator as a discriminator in a union, *rtiddsgen2* failed to resolve the enumerator labels as case labels in a union. The problem has been resolved.

[RTI Issue ID CODEGENII-5]

3.13 Possible Exception if Type Inherited from Typedef of Struct

If an IDL file contained a type that inherited from a typedef of a struct, *rtiddsgen2* might have thrown a null pointer exception. This problem has been resolved.

[RTI Issue ID CODEGENII-84]

3.14 Parsing Error if Struct Inherited from Typedef of Typedef with '@resolve-name false' Annotation

Given this use case:

```
typedef A B; //@resolve-name false
typedef B C;

struct D: C{
   long a;
}
```

The *rtiddsgen2* parser reported this error:

```
"Unexpected base type. Valid types are 'struct' or 'valuetype'"
```

due to the presence of the //@resolve-name false annotation.

The problem has been resolved and the use case is now supported.

[RTI Issue ID CODEGENII-86]

3.15 Incorrect Code Generated for Type Inherited from Keyed Type with Forward Declaration

Given the following scenario:

```
valuetype A{
    long m1; //@key
}
valuetype B:A{
    long m2;
}
```

The generated code for type B was incorrect because it did not consider that type A was keyed. This problem has been resolved.

[RTI Issue ID CODEGENII-89]

3.16 Failure to Check if Forward Declaration Type was Actually Defined Later

rtiddsgen2 did not check to make sure that a type declared in a forward declaration type was actually defined later. For example:

```
struct MyFwdStruct;

struct MyStruct {
    MyFwdStruct * m2;
};
```

In previous releases, *rtiddsgen2* generated code that did not compile. This issue has been resolved. In this release, *rtiddsgen2* reports the following error when parsing the above IDL:

```
ERROR com.rti.ndds.nddsgen.Main MyType.idl line 1:0 forward declaration
MyFwdStruct is not defined
```

[RTI Issue ID CODEGENII-97]

3.17 Generated C++ Code for 'long long' Constants did not Compile on Some Architectures

On some architectures such as VxWorks, you may have seen this error when compiling the C++ generated code for an IDL file that included a long long constant:

```
error: integer constant is too large for 'long' type
```

The issue has been resolved by adding an 'LL' suffix to the value when initializing the constant.

[RTI Issue ID CODEGENII-117]

3.18 Incorrect XML Generated for Unions with '@top-level false' Annotation

An XML file generated from IDL with a union that used the '@top-level false' annotation resulted in a badly formatted XML file. The annotation was written in the discriminator tag instead of the union tag. This problem has been resolved.

[RTI Issue ID CODEGENII-120]

3.19 Schema File Not Found After using -convertToXML if NDDSHOME Not Set

When using *rtiddsgen2's* **-convertToXML** option, the resulting XML included an incorrectly formatted variable, **xsi:noNamespaceSchemaLocation** (which contains the path to the schema file).

Consequently, if you input the resulting XML file to *rtiddsgen2* (with the **-inputXML** option), *rtiddsgen2* may not have been able to find the schema file (unless the NDDSHOME environment variables was defined). You may have seen this warning:

```
WARN com.rti.ndds.nddsgen.xml.XMLParser Schema file not found. Unable to
locate file : rti_dds_topic_types.xsd
```

This problem has been resolved; the absolute path to the schema file is now stored correctly.

[RTI Issue ID CODEGENII-121, CODEGENII-125]

3.20 Failure to Check if Constant used as Dimension of Array was Actually Declared

rtiddsgen2 did not check whether a constant used as a dimension value for an array was actually declared. For example:

```
struct myArrayStruct {
  char myCharArray[5][6][constModule::otherConst]; //@key
  short anotherSort;
};
```

Compilation of the generated code for **myArrayStruct** would have failed because **constModule::otherConst** cannot be found. In this release, the code generator will detect this situation and print the following message:

```
ERROR com.rti.ndds.nddsgen.Main MyType.idl line 2:25 symbol 'constMod-
ule::otherConst' not found
```

[RTI Issue ID CODEGENII-122]

3.21 Null Pointer Exception when Setting Extensibility of Struct/Valuetype with Base Type of Typedef of Struct

rtiddsgen2 threw a null-pointer exception when setting the extensibility of a struct/valuetype whose base type was a typedef of a struct. This problem has been resolved.

[RTI Issue ID CODEGENII-139]

3.22 Union Forward Declarations were Considered Struct Forward Declarations

Due to this bug the conversion from IDL to XML was wrong. For example:

```
union MyUnion;

struct MyStruct {
    MyUnion * m1;
};
```

The conversion of the above IDL snippet to XML (using the **-convertToXml** option) would generate the following forward declaration in XML:

```
<forward_dcl name="MyUnion" kind="struct"/>
```

when it should be:

```
<forward_dcl name="MyUnion" kind="union"/>
```

This issue has been resolved.

[RTI Issue ID CODEGENII-140]

3.23 Forward Declaration Support was not Compliant with Rules Defined in IDL Specification

The forward declaration was not in compliance with the rules defined in the IDL specification. Specifically:

- ❑ It was possible to inherit from a forward-declared value type (structure) whose definition has not yet been seen. This is disallowed in the new specification.
- ❑ If a structure or union is forward declared, the compiler did not check that the definition of that structure or union must follow the forward declaration.
- ❑ It was possible to refer an incomplete type in some scenarios that are not supported. For example:

  ```
  struct MyStruct

  struct MyStruct2 {
      MyStruct m1;
  };
  struct MyStruct {
      long m1;
  };
  ```

  The above IDL snippet is illegal IDL. However, the code generator did not report the problem and it generated code that did not compile in C and C++.

  Notice that the previous example would be legal if the member "m1" was optional or a pointer. For example:

  ```
  struct MyStruct

  struct MyStruct2 {
      MyStruct m1; //@Optional
  };
  struct MyStruct {
      long m1;
  };
  ```

This problem has been resolved.

[RTI Issue ID CODEGENII-141]

3.24 Generated C/C++ Code with Typecodes may have Included Incorrect End-of-Line Characters

Generated C/C++ code that included typecode information might have used Windows-style end-of-line characters, even if *rtiddsgen* was run on a Linux system. This problem has been resolved.

[RTI Issue ID CODEGENII-142]

3.25 Error when Generating Code from IDL Containing Unbounded Sequence of Types with Composed Name

When generating code from IDL that contained one of the following unbounded sequences: sequence<long long> sequence<long double> sequence<unsigned long> sequence<unsigned long long> or sequence<unsinged short>, *rtiddsgen* reported this error:

```
java.lang.IndexOutOfBoundsException: Index: 2, Size: 2
```

The problem has been resolved.

[RTI Issue ID CODEGENII-145]

3.26 Hexadecimal Values for Enumerators not Supported

In previous releases, hexadecimal values for enumerators were not supported. For example:

```
enum MyEnum {
    ONE = 0x00,
    TWO = 0x10
};
```

The code generation for the above enumeration would have failed with an error similar to this:

```
ERROR com.rti.ndds.nddsgen.Main Fail: For input string: "0x00"
```

The problem has been resolved.

[RTI Issue ID CODEGENII-151]

4 Third-Party Licenses

Portions of *RTI Code Generator 2* were developed using:

❑ Apache log4j™ from the Apache Software Foundation (http://logging.apache.org/log4j/)

❑ Apache Velocity™ from the Apache Software Foundation (http://velocity.apache.org/)

❑ ANTLR v3 (http://www.antlr3.org/)

4.1 Apache Software License Version 2.0

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document. "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

4.2 ANTLR 3 License

[The BSD License]

Copyright (c) 2010 Terence Parr

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- ❑ Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- ❑ Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- ❑ Neither the name of the author nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN