

RTI CORBA Compatibility Kit

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 94089
Phone: (408) 990-7444
Email: support@rti.com
Website: <https://support.rti.com/>

Release Notes

1 Supported Platforms

RTI® CORBA Compatibility Kit is supported on the following architectures:

- ACE 5.6a - TAO 1.6a

| Operating System | CPU | Compiler | RTI Architecture Abbreviation |
|------------------|----------|-----------|-------------------------------|
| LynxOS® 5.0 | PPC 7400 | gcc 3.4.3 | ppc7400Lynx5.0.0gcc3.4.3 |

- Java (JacORB 2.2.4)

| Operating System | |
|------------------|--|
| Linux® | All Linux platforms on x86/x64 CPUs listed in the <i>RTI Core Libraries Release Notes</i> for the same version number that support Java. |
| Solaris™ | All Solaris architectures listed in the <i>RTI Core Libraries Release Notes</i> |
| Windows® | All Windows architectures listed in the <i>RTI Core Libraries Release Notes</i> |

Please see the *RTI Core Libraries Platform Notes* for more information on these supported architectures, including their required system libraries, compiler flags, etc.

2 Compatibility with CORBA

When used with the **-corba** option, *rtiddsgen* generates type-specific code that is compatible with the OMG CORBA-IDL mapping. As a result, the generated code will be compatible with a large set of CORBA distributions. RTI tests compatibility against the OCI CORBA source-code distribution for C++, JacORB for Java, and ACE+TAO from the Distributed Object Computing (DOC) Group for Distributed Real-time and Embedded (DRE).

This version of *CORBA Compatibility Kit* is intended for *RTI Connex DDS* with the same version number and:

- The DOC Group's distribution of ACE 6.0.1 - TAO 2.0.1 for C++ for the architectures listed in [Section 1](http://download.dre.vanderbilt.edu/). (<http://download.dre.vanderbilt.edu/>)
- OCI's distribution of ACE 5.6a - TAO 1.6a for C++ for the architectures listed in [Section 1](http://www.theaceorb.com/downloads/index.html). (<http://www.theaceorb.com/downloads/index.html>)

- JacORB 2.2.4 for Java for architectures listed in [Section 1](#). (<http://www.jacorb.org/download.html>)
-

3 What's New in 5.3.1

There are no changes to *CORBA Compatibility Kit* in release 5.3.1.

4 What's New in 5.3.0

This release adds support for platforms using JacORB 2.2.4 and the Ubuntu 16.04 LTS operating system (architectures i86Linux3gcc5.4.0 and x64Linux3gcc5.4.0).

4.1 Platforms on Legacy Operating Systems

The following legacy operating systems have reached end-of-life from their corresponding vendors. Please contact RTI support or your account manager if you require version 5.3 to run on these platforms:

- CentOS™ 5.x
- Red Hat® Enterprise Linux 5.x
- SUSE® 11

4.2 Removed Platforms

Platforms on the following operating systems are no longer supported:

- Red Hat Enterprise Linux 5.2 with Real-Time Extensions
 - Solaris 2.9
 - Windows Vista, Windows XP Pro, Windows 2003
-

5 Additional Instructions for LynxOS 5.0 and OCI ACE 5.6a - TAO 1.6a

If you are building for Lynx target version 5.0.0 and using the OCI ACE 5.6a - TAO 1.6a package, you will need to make the following change:

In `ACE_wrappers/include/makeinclude/platform_lynxos.GNU` (line #110), replace:

```
LIBS      += -lnetinet -lnsl
```

with:

```
ifeq (5.0.0,$(VERSION))
  LIBS      += -lnetinet
else
  LIBS      += -lnetinet -lnsl
endif
```

The above modification omits the **libnsl** library for version 5.0.0. This change is needed because according to LinuxWorks, the Name Service library (**libnsl**) is not supported in Lynx 5.0.0. Including **libnsl** will cause a link error when building for CORBA.

6 Known Issues

6.1 Unsupported IDL Types

When using *rtiddsgen* with the `-corba` option, some IDL types are *not* supported. For more information about supported IDL types, see the *Data Types and Data Samples* chapter in the *RTI Connext DDS Core Libraries User's Manual*.

6.2 Extensible Types Not Supported

RTI CORBA Compatibility Kit does not support the "Extensible and Dynamic Topic Types for DDS" (DDS-XTypes) specification from the Object Management Group (OMG).

Specifically, *RTI CORBA Compatibility Kit* does not support the following features:

- Optional members
- Mutable types
- Extensible types. You may still mark a type as Extensible using the Extensibility annotation. However, both the *DataWriter* and *DataReader* must have the same type definition to interoperate. For example:

DataWriter:

```
struct BaseType {
    long m1;
};
```

DataReader:

```
struct DerivedType {
    long m1;
    long m2;
};
```

When using *RTI CORBA Compatibility Kit*, the *DataWriter* and *DataReader* above will match. However, the *DataReader* will fail to deserialize the samples from the *DataWriter*.

[RTI Issue ID CORE-6247]

6.3 LNK2005 Error When Using an ACE-TAO Class that Inherits from a Template Base Class (Windows Only)

If you create a Windows DLL that uses an ACE-TAO class, and that class inherits from a template base class, you may see a LNK2005 error complaining about one or more multiple defined symbols.

This is a known issue when using Microsoft® Visual Studio®.

As a possible workaround, you can explicitly import the template base class in the IDL file using the `// @copy-c-declaration` directive. For example, if you see the following error:

```
TAO.lib(TAO.dll): error LNK2005: "public: class TAO_Unbounded_Sequence<unsigned short> & __thiscall TAO_Unbounded_Sequence<unsigned short>::operator=(class TAO_Unbounded_Sequence<unsigned short> const &)" (?4?$TAO_Unbounded_Sequence@G@@QA>>EAAV0@ABV0@@Z) already defined in SendData.obj TAO.lib(TAO.dll) : error LNK2005: "public: unsigned short const * __thiscall TAO_Unbounded_Sequence<unsigned short>::get_buffer(void) const" (?get_buffer@?$TAO_Unbounded_Sequence@G@@@QBEPBGXZ) already defined in SendDataPlugin.obj TAO.lib(TAO.dll) : error LNK2005: "public: unsigned short * __thiscall TAO_Unbounded_Sequence<unsigned short>::get_buf-
```

```
fer(bool)" (?get_buffer@?$TAO_Unbounded_Sequence@G@@QAEPAZ) already  
defined in SendDataPlugin.obj
```

You can add the following line at the beginning of your IDL file:

```
//@copy-c-declaration template class __declspec(dllimport) TAO_Unbound-  
ed_Sequence<unsigned short>;
```