

# **RTI Connex Core Libraries**

**Getting Started Guide**

**Addendum for Embedded Systems**

**Version 7.1.0**



## Trademarks

RTI, Real-Time Innovations, Connex, NDDS, 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 solely under and subject to RTI's standard terms and conditions available at <https://www.rti.com/terms> and in accordance with your License Acknowledgement Certificate (LAC) and Maintenance and Support Certificate (MSC), except to the extent otherwise accepted in writing by a corporate officer of RTI.

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/>). This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

## Notices

### *Early Access Software*

“Real-Time Innovations, Inc. (“RTI”) licenses this Early Access release software (“Software”) to you subject to your agreement to all of the following conditions:

- (1) you may reproduce and execute the Software only for your internal business purposes, solely with other RTI software licensed to you by RTI under applicable agreements by and between you and RTI, and solely in a non-production environment;
- (2) you acknowledge that the Software has not gone through all of RTI’s standard commercial testing, and is not maintained by RTI’s support team;
- (3) the Software is provided to you on an “AS IS” basis, and RTI disclaims, to the maximum extent permitted by applicable law, all express and implied representations, warranties and guarantees, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, satisfactory quality, and non-infringement of third party rights;

(4) any such suggestions or ideas you provide regarding the Software (collectively , “Feedback”), may be used and exploited in any and every way by RTI (including without limitation, by granting sub-licenses), on a non-exclusive, perpetual, irrevocable, transferable, and worldwide basis, without any compensation, without any obligation to report on such use, and without any other restriction or obligation to you; and

(5) TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT WILL RTI BE LIABLE TO YOU FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, EXEMPLARY OR PUNITIVE OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR FOR LOST PROFITS, LOST DATA, LOST REPUTATION, OR COST OF COVER, REGARDLESS OF THE FORM OF ACTION WHETHER IN CONTRACT, TORT (INCLUDING WITHOUT LIMITATION, NEGLIGENCE), STRICT PRODUCT LIABILITY OR OTHERWISE, WHETHER ARISING OUT OF OR RELATING TO THE USE OR INABILITY TO USE THE SOFTWARE, EVEN IF RTI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.”

### *Deprecations and Removals*

Any deprecations or removals noted in this document serve as notice under the Real-Time Innovations, Inc. Maintenance Policy #4220 and/or any other agreements by and between RTI and customer regarding maintenance and support of RTI’s software.

*Deprecated* means that the item is still supported in the release, but will be removed in a future release. *Removed* means that the item is discontinued or no longer supported. By specifying that an item is deprecated in a release, RTI hereby provides customer notice that RTI reserves the right after one year from the date of such release and, with or without further notice, to immediately terminate maintenance (including without limitation, providing updates and upgrades) for the item, and no longer support the item, in a future release.

### **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/>

# Contents

---

<b>Chapter 1 Addendum for Embedded Platforms</b> .....	<b>1</b>
<b>Chapter 2 Getting Started on QNX Embedded Systems</b>	
2.1 Building and Running a Hello World Example .....	<b>2</b>

# Chapter 1 Addendum for Embedded Platforms

In addition to enterprise-class platforms like Microsoft Windows and Linux, *RTI® Connex*® supports a wide range of embedded platforms. This document is especially for users of those platforms. It describes how to configure some of the most popular embedded systems for use with *Connex* and to get up and running as quickly as possible. The code examples covered in this document can be generated for your platform(s) using *RTI Code Generator (rtiddsgen)*, which accompanies *Connex*.

This document assumes at least minimal knowledge with the platforms it describes and is not a substitute for the documentation from the vendors of those platforms. For further instruction on the general operation of your embedded system, please consult the product documentation for your board and operating system.

# Chapter 2 Getting Started on QNX Embedded Systems

This document provides instructions on building and running *Connex* applications on embedded systems such as QNX® systems. It will guide you through the process of generating, compiling, and running a Hello World application on an embedded QNX system by expanding on Hands-On 1 of *Introduction to Publish/Subscribe*, in the [RTI Connex Getting Started Guide](#). Please read the following alongside that section.

In the following steps:

- All commands must be executed in a command shell that has all the required environment variables. For details, see Set Up Environment Variables (rtisetenv), in "Hands-On 1" of *Introduction to Publish/Subscribe*, in the [RTI Connex Getting Started Guide](#).
- You need to know the name of your target architecture (look in your **NDDSHOME/lib** directory). Use it in place of *<architecture>* in the example commands. For example, your architecture might be 'armv8QNX7.1qcc\_gpp8.3.0'.
- We assume that you have **make** installed. If you have **make**, you can use the generated makefile to compile. If you do not have **make**, use your normal compilation process. (Note: the generated makefile assumes the correct version of the compiler is already in your path and that **NDDSHOME** is set.)

## 2.1 Building and Running a Hello World Example

This section describes the basic steps for building and running an *rtiddsgen*-generated example on an embedded target such as QNX.

1. Create a directory to work in. In this example, we use a directory called **myhello**.
2. In the **myhello** directory, create a file called **HelloWorld.idl** that contains a user-defined data type:

```
struct HelloWorld {
    string<128> msg;
};
```

3. Use the `rtiddsgen` utility to generate sample code and a makefile as shown below. Substitute `<architecture>` with your target architecture string, such as `armv8QNX7.1qcc_gpp8.3.0`.

For C++:

```
rtiddsgen -language C++ -example <architecture> HelloWorld.idl
```

Edit the example code to add this line:

```
printf(instance->msg, "Hello World! (%d)", count);
```

It should look like this:

```
for (count=0; (sample_count == 0) || (count < sample_count); ++count) {

    printf("Writing HelloWorld, count %d\n", count);

    /* Modify the data to be written here */
    printf(instance->msg, "Hello World! (%d)", count);

    /* Write data */
    retcode = HelloWorldDataWriter_write(
        HelloWorld_writer, instance, &instance_handle);
    if (retcode != DDS_RETCODE_OK) {
        fprintf(stderr, "write error %d\n", retcode);
    }

    NDDS_Utility_sleep(&send_period);
}
```

4. With the `NDDSHOME` environment variable set, build the Publisher and Subscriber modules using the generated makefile.

```
make -f makefile_HelloWorld_<architecture>
```

For details on setting up the `NDDSHOME` environment variable, see Set Up Environment Variables (`rtisetenv`), in "Hands-On 1" of *Introduction to Publish/Subscribe*, in the [RTI Connexx Getting Started Guide](#).

After compiling, find the application executables in `myhello/objs/<architecture>`.

5. Connect to the QNX target (using `ssh`, for example) and start the subscriber application, **HelloWorld\_subscriber**.

```
HelloWorld_subscriber
```

In this shell, you should see that the subscriber is waking up every 4 seconds to print a message. Here is a C++ example:

```
No data after 1 second
No data after 1 second
No data after 1 second
```

6. Connect to the QNX target and start the publisher application, **HelloWorld\_publisher**.

```
HelloWorld_publisher
```

In this second (publishing) shell, you should see:

```
Writing HelloWorld, count 0
Writing HelloWorld, count 1
Writing HelloWorld, count 2
```

7. Look back in the first (subscribing) shell. You should see that the subscriber is now receiving messages from the publisher.

For example, in C++:

```
Received data
  msg: "Hello World! (0) "
Received data
  msg: "Hello World! (1) "
Received data
  msg: "Hello World! (2) "
```