

RTI Connex DDS Core Libraries

Getting Started Guide

Addendum for INtime Systems

Version 6.0.1



© 2020 Real-Time Innovations, Inc.
All rights reserved.
Printed in U.S.A. First printing.
March 2020.

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

Chapter 1 INtime Systems	1
Chapter 2 Installing and Configuring INtime SDK	
2.1 Prerequisites	2
2.2 Installing Connex DDS for INtime Development	3
Chapter 3 INtime Development Model	
3.1 Motivation	4
3.2 Inter-Node Communication	4
3.3 INtime Software Development Kit	4
Chapter 4 Using Connex DDS Utilities on INtime for Windows	
4.1 RTI DDS Ping App	6
4.2 RTI DDS Spy App	7
4.3 RTI DDS Prototyper App	9
Chapter 5 Using the Connex DDS C and C++ Libraries with INtime on VS 2017	
5.1 Setting Up	10
5.1.1 Environment Variables	10
5.1.2 INtime Kernel Configuration	11
5.2 Building and Running a "Hello, World" Example	13
5.2.1 Start the Publisher	14
5.2.2 Start the Subscriber	15

Chapter 1 INtime Systems

This document provides instructions to install and configure *RTI® Connex® DDS* for INtime® for Windows development and run *Connex DDS* applications on INtime systems. Please refer to the documentation provided by TenAsys® for more information on the INtime operating system: <http://www.tenasys.com/support/documentation/>.

This section will guide you through the process of generating, compiling, and running a Hello World application on an INtime system by expanding on [Generating Code with RTI Code Generator, in the RTI Connex DDS Core Libraries Getting Started Guide](#).

[Chapter 2 Installing and Configuring INtime SDK on page 2](#) describes how to install and configure *Connex DDS* for INtime development.

[Chapter 3 INtime Development Model on page 4](#) gives an overview of the INtime development model.

If you want to use the *Connex DDS* Ping, Spy, or Prototyper utilities, please read [Chapter 4 Using Connex DDS Utilities on INtime for Windows on page 6](#)

[Chapter 5 Using the Connex DDS C and C++ Libraries with INtime on VS 2017 on page 10](#) will guide you through the steps to generate, modify, build, and run an example HelloWorld application on an INtime target. You will use `rtiddsngen` to generate example source code.

Chapter 2 Installing and Configuring INtime SDK

2.1 Prerequisites

1. The development machine needs to meet the following minimum requirements:
 - Pentium class (or better) processor
 - 40MB of free RAM for INtime and your real-time application
 - 250MB hard disk space for tools, examples, and documentation
 - Windows 10, Windows 8, Windows 7, Windows Vista, Windows Server 2016, Windows Server 2012/2012R2, Windows Server 2008/2008R2, Windows Server 2003 R2, or Windows XP with Service Pack 3 (See the Knowledge Base at www.tenasys.com/support for updated Windows version support information). Both 32- and 64-bit versions of Windows are supported.
 - Windows must be loaded using the Windows bootloader. Third-party bootloaders will not load INtime successfully.
 - Microsoft Visual Studio (2017, 2015, 2013, 2012, 2010, or 2008) (See the Knowledge Base at www.tenasys.com/support for updated Visual Studio version support information).

INtime for Windows applications run with Windows 10, Windows 8, Windows 7, Windows Vista, Windows XP, Windows XP Embedded, Windows Server 2016, 2012R2, 2008R2, and 2003R2.

2. Install Microsoft Visual Studio (2017, 2015, 2013, 2012, 2010, or 2008), available from <https://visualstudio.microsoft.com/>. Instructions and further documentation are available from <https://docs.microsoft.com/en-us/visualstudio/ide/>.

3. Install the INtime 6 SDK (Software Development Kit) with the *INtime for Windows* configuration. For instructions, consult the *INtime 6 Quick Start Guide* (pp. 5-10), available from <http://www.tenasys.com/support/documentation/>.

2.2 Installing Connex DDS for INtime Development

You will need to install a host archive and a target archive. The host archive contains documentation, header files, and other files you need for development; the target archive contains the libraries and other files you need for deployment. Unpack them as described in [RTI Connex DDS Installation Guide](#).

Chapter 3 INtime Development Model

INtime is a real-time operating system (RTOS) developed by TenAsys. It can be run in one of two usage configurations:

- *INtime for Windows*: INtime RTOS runs alongside Microsoft Windows OS on the same hardware platform.
- *INtime Distributed RTOS*: INtime runs as a stand-alone RTOS.

This document will only cover development of *Connex DDS* applications for the *INtime for Windows* configuration.

3.1 Motivation

INtime for Windows is intended for use in applications that incorporate both feature-rich and real-time components. Feature-rich components like machine-user interfaces, graphical representation, mathematical modeling, and I/O support benefit from running in a Windows environment, while real-time components like advanced CNC, PLC, medical equipment, and robotics run on the INtime RTOS. *INtime for Windows* manages communication between these applications, allowing them to interact at the process level.

3.2 Inter-Node Communication

As described above, *INtime for Windows* allows INtime and Windows to run simultaneously on the same hardware platform. *INtime for Windows* partitions system resources between the two environments to ensure that they do not conflict with each other and remain independent. An INtime partition is known as a "node" in the INtime environment.

3.3 INtime Software Development Kit

All development is performed with the INtime SDK running on a Windows PC and comes as a complete solution with tools and utilities to facilitate the development, deployment, and field

maintenance of embedded real-time applications. Code development uses Microsoft Visual Studio on a standard Windows based PC platform, programming in C and C++.

Chapter 4 Using Connex DDS Utilities on INtime for Windows

You can use the Connex DDS *rtiddsping*, *rtiddsspy*, and *rtiddsprototyper* utilities on *INtime for Windows* through the File Explorer or Command Prompt.

4.1 RTI DDS Ping App

To use *rtiddsping* on INtime:

1. Load and run the INtime 6 Kernel on your system. From Windows desktop, click on the INtime icon on the right-hand side of the taskbar and select "Start <node>".
2. Open the file **rtiddsping.rta**. This can be done with File Explorer or Command Prompt.

With File Explorer: navigate to **\$NDDSHOME/bin** and double click on **rtiddsping.rta**.

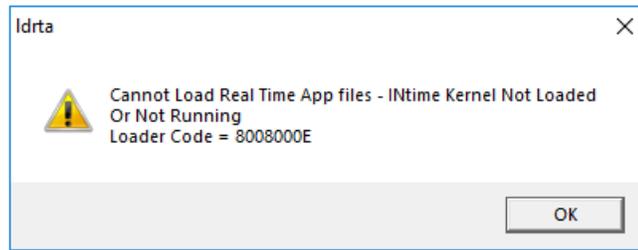
With Command Prompt: change directories to **\$NDDSHOME/bin**, as shown in the example below:

```
C:\> cd $NDDSHOME/bin
```

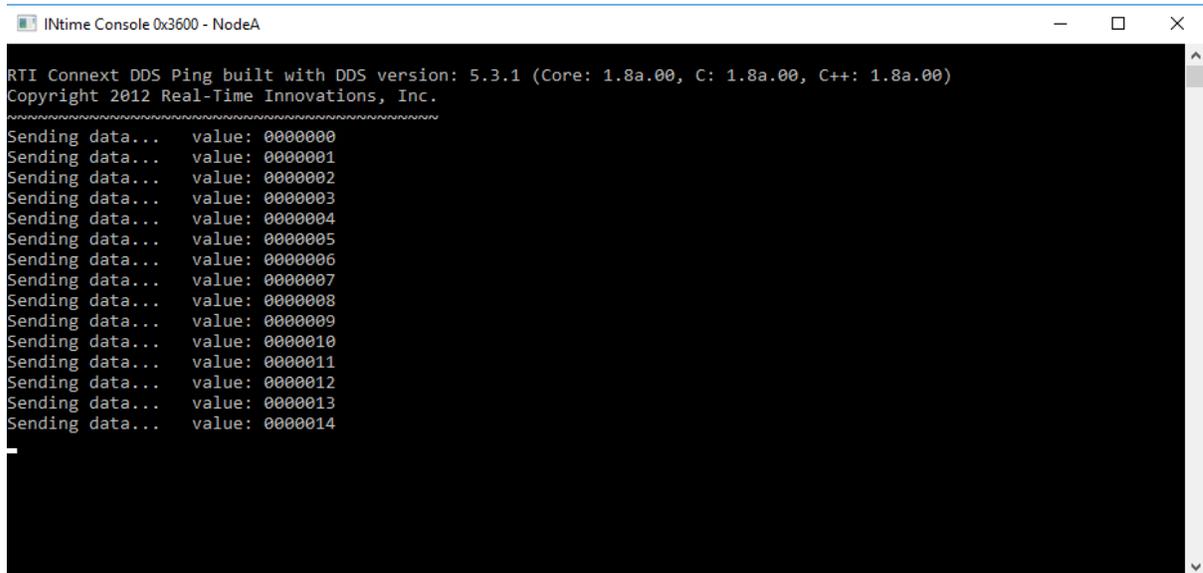
Then call **rtiddsping.rta**:

```
C:\Program Files\rti_connex_dds-6.0.1\bin> rtiddsping
```

NOTE: If a window appears with the following error message, make sure that the INtime 6 Kernel is running on your system; see step 1.



3. INtime will start running *rtiddsping* and open an INtime Console window.



Data from *rtiddsping* is displayed in the INtime Console window.

4.2 RTI DDS Spy App

To use *rtiddsspy* on INtime:

1. Load and run the INtime 6 Kernel on your system. From Windows desktop, click on the INtime icon on the right-hand side of the taskbar and select "Start <node>".
2. Open the file **rtiddsspy.rta**. This can be done with File Explorer or Command Prompt.

With File Explorer: navigate to **\$NDDSHOME/bin** and double click on **rtiddsspy.rta**.

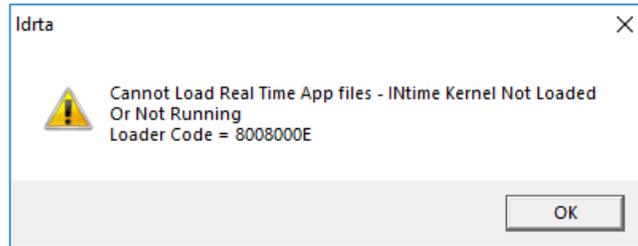
With Command Prompt: change directories to **\$NDDSHOME/bin**, as shown in the example below:

```
C:\> cd $NDDSHOME/bin
```

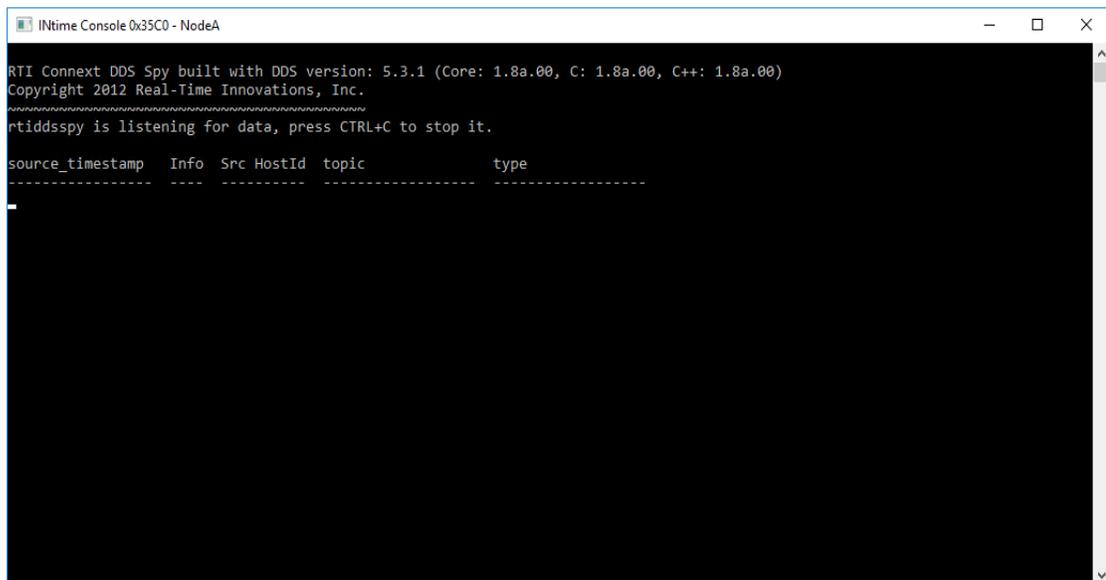
Then call **rtiddsspy.rta**:

```
C:\Program Files\rtdi_connex_dds-6.0.1\bin> rtdiddsspy
```

NOTE: If a window appears with the following error message, make sure that the INtime 6 Kernel is running on your system; see step 1.



3. INtime will start running *rtdiddsspy* and open an INtime Console window.



Data from *rtdiddsspy* is displayed in the INtime Console window. If *rtdiddsping* is running simultaneously, *rtdiddsspy* should receive and display matching data. For more information on running *rtdiddsping*, see [4.1 RTI DDS Ping App on page 6](#).

```

RTI Connexx DDS Ping built with DDS version: 5.3.1 (Core: 1.8a.00, C: 1.8a.00, C++: 1.8a.00)
Copyright 2012 Real-Time Innovations, Inc.
Sending data... value: 000000
Found 1 additional ping subscriber(s).
Current subscriber tally is: 1
Sending data... value: 000001
Sending data... value: 000002
Sending data... value: 000003
Sending data... value: 000004
Sending data... value: 000005
Sending data... value: 000006
Sending data... value: 000007
Sending data... value: 000008
Sending data... value: 000009
Sending data... value: 000010
Sending data... value: 000011
Sending data... value: 000012
Sending data... value: 000013
Sending data... value: 000014
Sending data... value: 000015
Sending data... value: 000016
Sending data... value: 000017
Sending data... value: 000018

RTI Connexx DDS Spy built with DDS version: 5.3.1 (Core: 1.8a.00, C: 1.8a.00, C++: 1.8a.00)
Copyright 2012 Real-Time Innovations, Inc.
rtiddsspy is listening for data, press CTRL+C to stop it.

-----
source_timestamp  Info  Src HostId  topic  type
-----
1532987943.062355  d +M  0A0A64DF  PingTopic  PingType
1532987944.062422  d +M  0A0A64DF  PingTopic  PingType
1532987945.062496  d +M  0A0A64DF  PingTopic  PingType
1532987946.062567  d +M  0A0A64DF  PingTopic  PingType
1532987947.062647  d +M  0A0A64DF  PingTopic  PingType
1532987948.062937  d +M  0A0A64DF  PingTopic  PingType
1532987949.062885  d +M  0A0A64DF  PingTopic  PingType
1532987950.062888  d +M  0A0A64DF  PingTopic  PingType
1532987951.062981  d +M  0A0A64DF  PingTopic  PingType
1532987952.063056  d +M  0A0A64DF  PingTopic  PingType
1532987953.063129  d +M  0A0A64DF  PingTopic  PingType
1532987954.063204  d +M  0A0A64DF  PingTopic  PingType
1532987955.063301  d +M  0A0A64DF  PingTopic  PingType
1532987956.063381  d +M  0A0A64DF  PingTopic  PingType
1532987957.063448  d +M  0A0A64DF  PingTopic  PingType
1532987958.063524  d +M  0A0A64DF  PingTopic  PingType
1532987959.060796  d +M  0A0A64DF  PingTopic  PingType
1532987960.060643  d +M  0A0A64DF  PingTopic  PingType
1532987961.061443  d +M  0A0A64DF  PingTopic  PingType

```

4.3 RTI DDS Prototyper App

To use *rtiddsprototyper* on INtime:

1. Load and run the INtime 6 Kernel on your system. From Windows desktop, click on the INtime icon on the right-hand side of the taskbar and select "Start <node>".
2. Open the file *rtiddsprototyper.rta*. This can be done with File Explorer or Command Prompt.

With File Explorer: navigate to `$NDDSHOME/bin` and double click on *rtiddsprototyper.rta*.

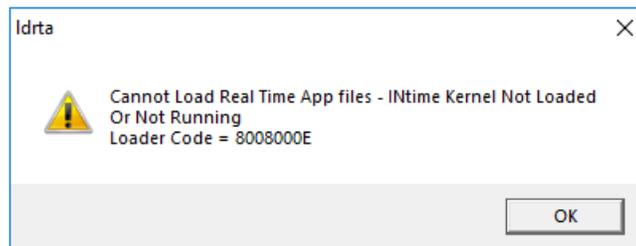
With Command Prompt: change directories to `$NDDSHOME/bin`, as shown in the example below:

```
C:\> cd $NDDSHOME/bin
```

Then call *rtiddsprototyper.rta*:

```
C:\Program Files\rti_connexx_dds-6.0.1\bin> rtiddsprototyper
```

NOTE: If a window appears with the following error message, make sure that the INtime 6 Kernel is running on your system; see step 1.



3. INtime will start running *rtiddsprototyper*.

Chapter 5 Using the Connex DDS C and C++ Libraries with INtime on VS 2017

This section describes how to build a *Connex DDS* application with C++ for an INtime target. The process for building a *Connex DDS* application with C is similar to that for C++. The following instructions and screenshots are for C++; any differences in the instructions for C will be called out in the affected steps. Also note that when developing with C, instructions that include the text “Cpp” should be replaced with “C”, such as in filenames, App names, etc. For example, HelloCppSubscriber for C++ would be HelloCSubscriber for C.

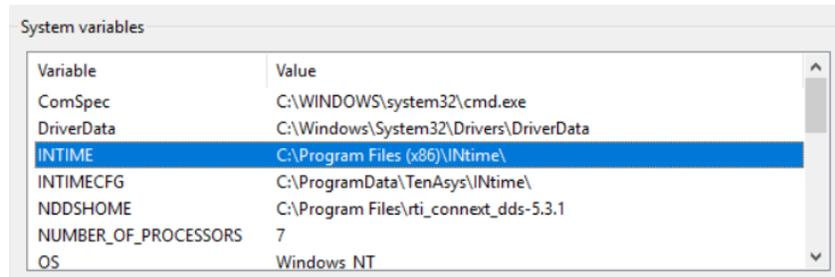
5.1 Setting Up

There are a few steps to take before building and running a Hello World application for INtime.

5.1.1 Environment Variables

The **NDDSHOME** environment variable should be set as described in [Step 1, Set up the Environment, in the RTI Connex DDS Core Libraries Getting Started Guide](#).

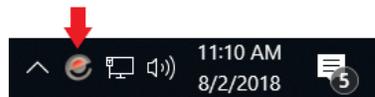
1. Create a new environment variable called **INTIME**.
2. Set the **INTIME** environment variable to your INtime installation directory. By default, the INtime root installation path is **C:\Program Files (x86)\INtime** as shown in the figure below.



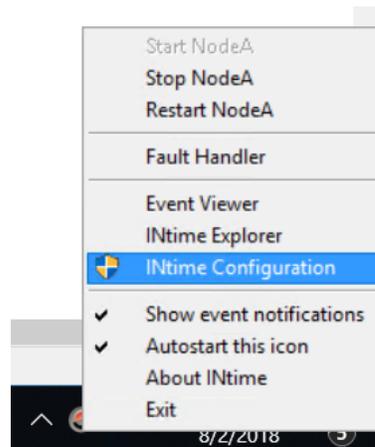
5.1.2 INtime Kernel Configuration

The INtime Kernel, by default, will search for dynamic libraries in pre-defined paths. The location of the RTI dynamic libraries must be added to the list of paths that INtime will search.

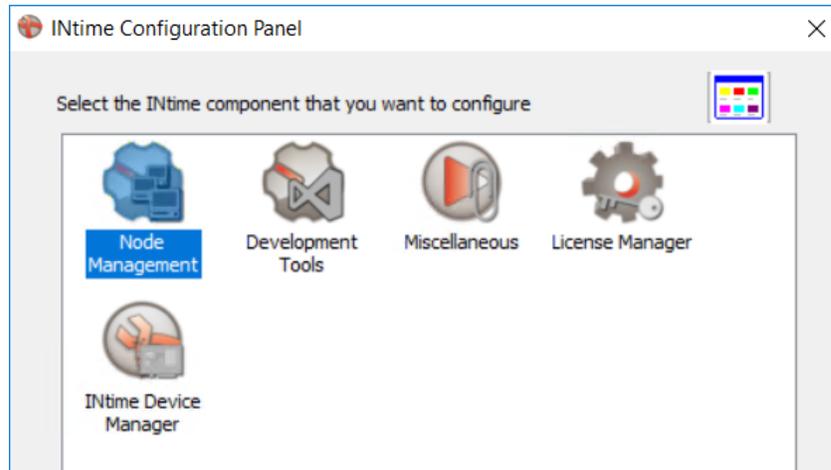
1. Click on the INtime Kernel node in the task bar.



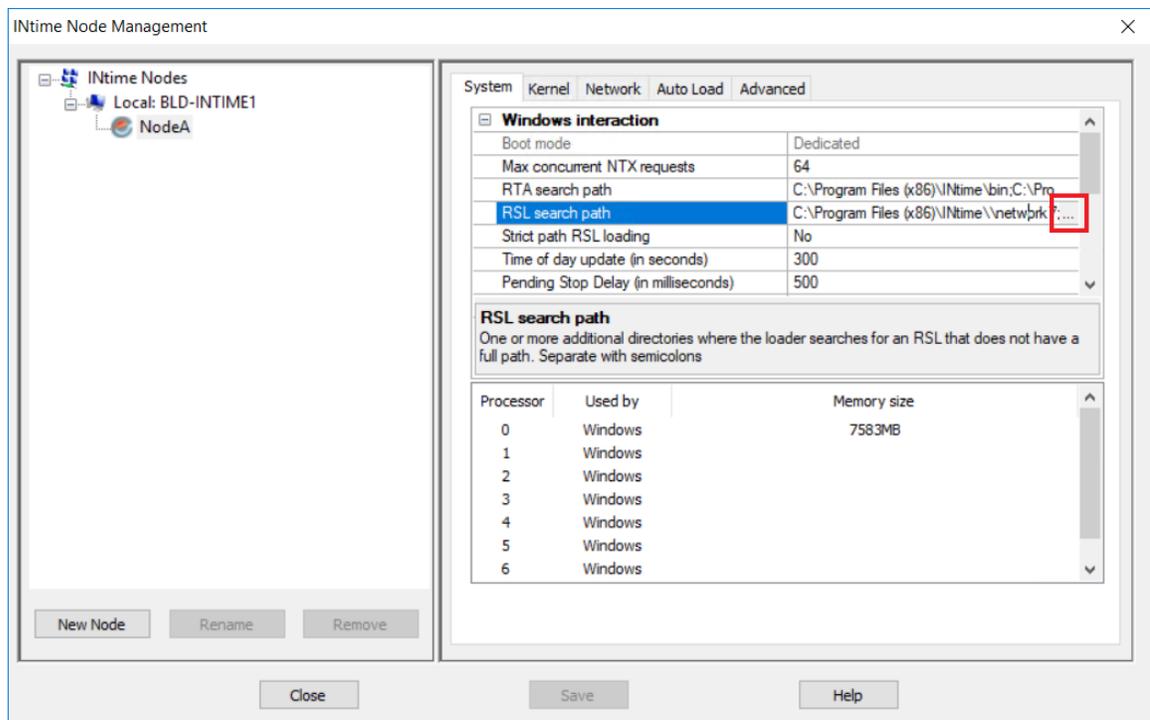
2. Select **INtime Configuration**.



3. Select **Node Management**.

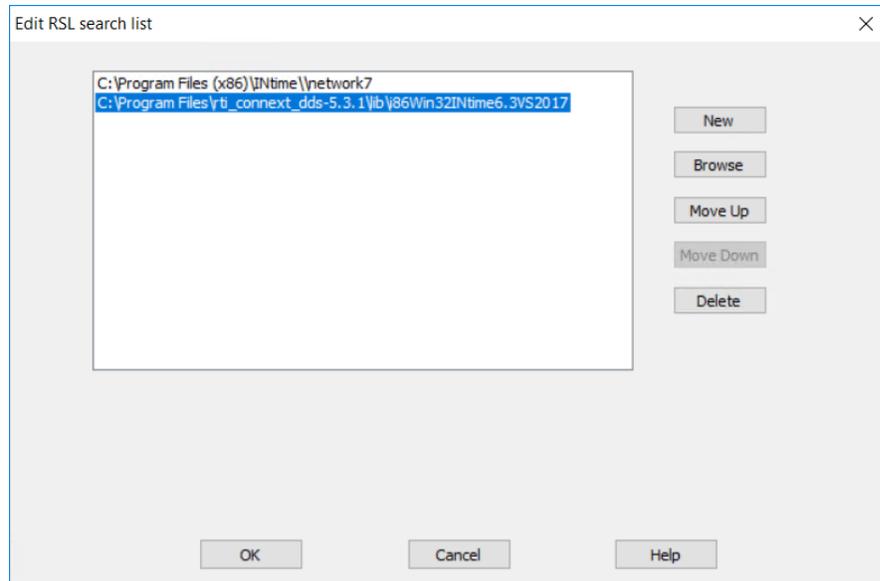


4. Select the applicable nodes from the left panel.
5. Select the **System** tab.
6. In the **System** tab, select the field "**Search RSL Path**". Click ... on the right-hand side of the row.



7. Click **New**. Add the path <NDDSHOME>/lib/i86INtime6.3VS2017 to the RSL search list, substituting <NDDSHOME> with your installation path to *Connex DDS*. Click **OK**.

NOTE: The `NDDSHOME` variable **cannot** be used in the path you specify in the RSL search list; the full system path **must** be given. If *Connex DDS* is installed at the default location, the system path will match the one shown in the figure below.



5.2 Building and Running a "Hello, World" Example

This section describes the basic steps for building and running an *rtiddsgen*-generated "Hello, World" example for an *INtime for Windows* system target.

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:

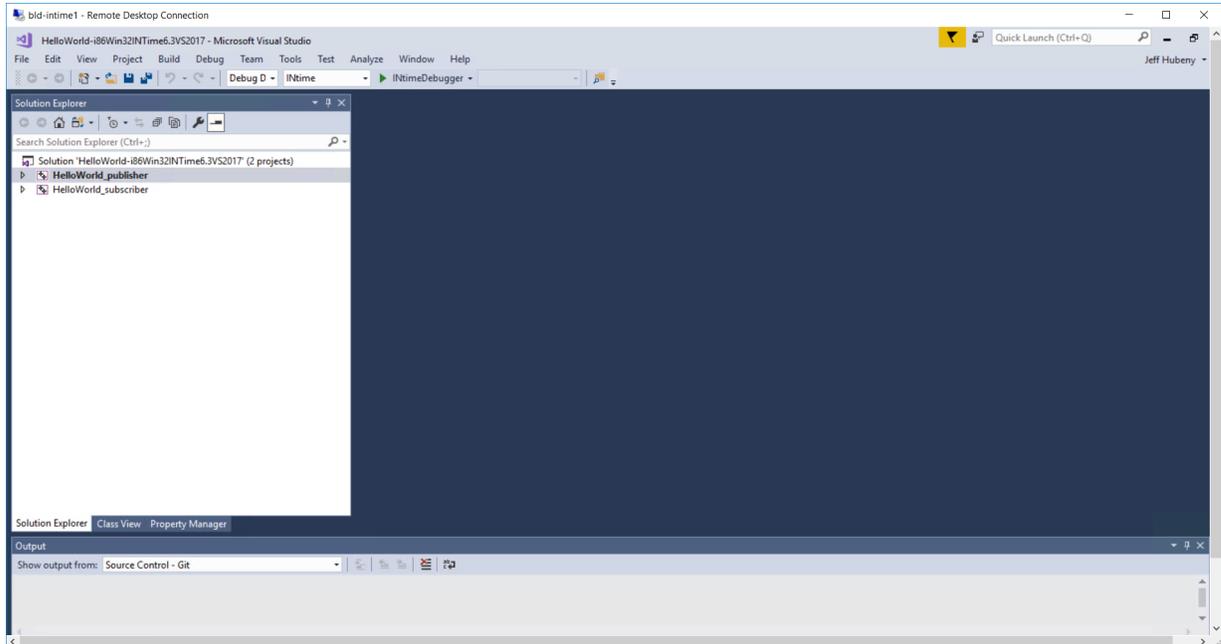
```
struct HelloWorld {
    string<128> msg;
};
```

3. Use the *rtiddsgen* utility to generate sample code.

Open **Command Prompt**, change directories to the location of **HelloWorld.idl**, and enter the following:

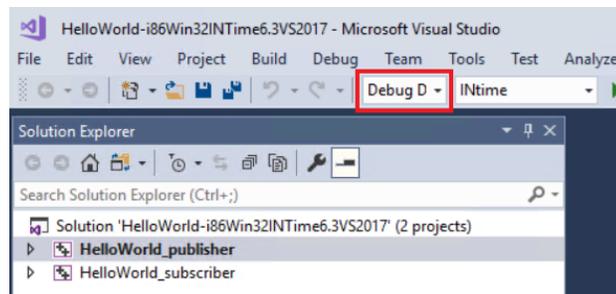
```
rtiddsgen -language C++ -example i86INtime6.3VS2017 HelloWorld.idl
```

4. *rtiddsgen* will output a Microsoft Visual Studio solution file with the extension **.sln**. Open this solution in Visual Studio; it will automatically open in the *INtime* perspective within Visual Studio.



You should see that there are two projects: **HelloWorld_publisher** and **HelloWorld_subscriber**.

5. Select the required build mode from the drop-down menu in the toolbar. There are 4 available modes: **Release**, **Debug**, **Release RSL**, and **Debug RSL**.



6. Select **Build** from the menu bar.
7. Click on **Build Solution**.

5.2.1 Start the Publisher

From your command shell, go to **examples\connext_dds\c++\hello_world** and type:

```
> piperta objs\i86INtime6.3VS2017\HelloWorld_publisher.rta
```

Note that **i86INtime6.3VS2017** is the architecture name for *INtime for Windows*; see the contents of the **examples** directory for a list of available architectures.

5.2.2 Start the Subscriber

From your command shell, go to **examples\connect_dds\c++\hello_world** and type:

```
> piperta objs\i86INtime6.3VS2017\HelloWorld_subscriber.rta
```

Note that **i86INtime6.3VS2017** is the architecture name for *INtime for Windows*; see the contents of the **examples** directory for a list of available architectures.

You should be able to see communication between **HelloWorld_publisher** and **HelloWorld_subscriber** while both are running.

Note: If you are using a **USER_QOS_FILE** to set QoS properties, be sure to place this file in the same directory as the **.rta** executable so that the QoS file is picked up correctly at runtime.