

# **RTI Connex DDS**

**Core Libraries**

**Custom Support for  
QNX Neutrino 6.6.0 Platforms  
on ARMv7 and x86 CPUs**

**Version 5.2.3**



© 2016 Real-Time Innovations, Inc.  
All rights reserved.  
Printed in U.S.A. First printing.  
April 2016.

## **Trademarks**

Real-Time Innovations, RTI, NDDS, RTI Data Distribution Service, DataBus, Connex, Micro DDS, 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.

## **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/>

# Chapter 1 Supported Platforms

This document supplements the [Release Notes](#) and [Platform Notes](#). It provides information specifically for the platform listed in [Custom Supported QNX Neutrino 6.6.0 Platforms](#). Please consult the *Platform Notes* for other information, such as thread configuration.

## Custom Supported QNX Neutrino 6.6.0 Platforms

Operating System	CPU	Compiler	RTI Architecture Abbreviation
QNX® Neutrino® 6.6.0	ARMv7a Cortex A9	qcc_cpp 4.7.3 with Dinkumware libraries	armv7aQNX6.6.0qcc_cpp4.7.3
	x86	qcc_cpp 4.7.3 with Dinkumware libraries	i86QNX6.6qcc_cpp4.7.3

The libraries for these platforms were built with the following patches and tested on systems that have these patches installed:

- QNX Software Development Platform 6.6 Graphics Patch (Patch ID 3875)
- QNX Software Development Platform 6.6 Header Files Patch (Patch ID 3851)
- Patch-660-3885-diskimage.tar for the BSP

# Chapter 2 Transports

These transports are supported:

- Shared memory
- UDPv4
- UDPv6: Supported. The transport is not enabled by default; the peers list must be modified to support IPv6. No Traffic Class support. To use the UDPv6 transport, the network stack must provide IPv6 capability. Enabling UDPv6 may involve switching the network stack server and setting up IPv6 route entries.
- TCP
- TLS

# Chapter 3 Features

These features are supported:

- Modern C++ API
- Multicast
- Monotonic clock
- Control of CPU core affinity for RTI threads
- Distributed Logger

These features are not supported:

- Durable Writer History and Durable Reader State

# Chapter 4 Compiling and Running

[Building Instructions for QNX Neutrino 6.6.0 Architectures](#) lists the compiler flags and libraries you will need to link into your application.

[Running Instructions for QNX Neutrino 6.6.0 Architectures](#) shows the environment variables required to be set at run time.

[Library-Creation Details for QNX Neutrino 6.6.0 Architectures](#) provides details on how these custom libraries were built. This table is provided strictly for informational purposes; you do not need to use these parameters to compile your application. You may find this information useful if you are involved in any in-depth debugging.

## Building Instructions for QNX Neutrino 6.6.0 Architectures

API	Library Format	RTI Libraries <sup>a b c</sup>	Required System Libraries	Required Compiler Flags
C++ (Traditional and Modern APIs)	Static Release	libniddscppz.a or libniddscpp2z.a libniddscz.a libniddscorez.a librticonnextmsgcppz.a	-lm -lsocket	-DRTI_QNX
	Static Debug	libniddscppzd.a or libniddscpp2zd.a libniddsczd.a libniddscorezd.a librticonnextmsgcppzd.a		
	Dynamic Release	libniddscpp.so or libniddscpp2.so libniddsc.so libniddscore.so librticonnextmsgcpp.so		
	Dynamic Debug	libniddscppd.so or libniddscpp2d.so libniddscd.so libniddscored.so librticonnextmsgcppd.so		

<sup>a</sup>The C/C++ libraries are in \$(NDDSHOME)/lib/<architecture>/. (where \$(NDDSHOME) is where Connex DDS is installed, such as /home/your user name/rti\_connex\_dds-5.x.y)

<sup>b</sup>The \*rticonnextmsg\* library only applies if you have the *Connex DDS* Professional, Evaluation, or Basic package type. It is not provided with the *Connex DDS* Core package type.

<sup>c</sup>Choose libniddscpp\*. for the Traditional C++ API or libniddscpp2\*. for the Modern C++ API.

## Building Instructions for QNX Neutrino 6.6.0 Architectures

API	Library Format	RTI Libraries <sup>a b c</sup>	Required System Libraries	Required Compiler Flags
C	Static Release	libniddscz.a libniddscorez.a librticonnextmsgcz.a	-lm -lsocket	-DRTI_QNX
	Static Debug	libniddsczd.a libniddscorezd.a librticonnextmsgzd.a		
	Dynamic Release	libniddsc.so libniddscore.so librticonnextmsgc.so		
	Dynamic Debug	libniddscd.so libniddscored.so librticonnextmsgcd.so		

## Running Instructions for QNX Neutrino 6.6.0 Architectures

RTI Architecture	Library Format (Release & Debug)	Environment Variables
All supported QNX architectures	Static	None required
	Dynamic	LD_LIBRARY_PATH= \${NDDSHOME}/lib/<architecture>: \${LD_LIBRARY_PATH} <sup>d</sup>

<sup>a</sup>The C/C++ libraries are in \$(NDDSHOME)/lib/<architecture>/. (where \$(NDDSHOME) is where Connexx DDS is installed, such as /home/your user name/rti\_connexx\_dds-5.x.y)

<sup>b</sup>The \*rticonnextmsg\* library only applies if you have the *Connexx DDS* Professional, Evaluation, or Basic package type. It is not provided with the *Connexx DDS* Core package type.

<sup>c</sup>Choose libniddscpp\*. for the Traditional C++ API or libniddscpp2\*. for the Modern C++ API.

<sup>d</sup>`\${NDDSHOME}` represents the root directory of your Connexx DDSConnexx DDSConnexx DDS installation. `\${LD\_LIBRARY\_PATH}` represents the value of the LD\_LIBRARY\_PATH variable prior to changing it to support Connexx DDS. When using niddsjava.jar, the Java virtual machine (JVM) will attempt to load release versions of the native libraries. When using niddsjava.jar, the JVM will attempt to load debug versions of the native libraries.



## Library-Creation Details for QNX Neutrino 6.6.0 Architectures

RTI Architecture	Library Format (Static & Dynamic)	Compiler Flags Used by RTI
armv7aQNX6.6.0gcc_cpp4.7.3	Release	gcc -Vgcc/4.7.3,gcc_ntoarmv7le_cpp -lang-c -fPIC -fexceptions -DFD_SETSIZE=512 -O -Wall -Wno-unknown-pragmas -DPtrIntType=long -DCSREAL_IS_FLOAT -DCPU=ARMV7 -DTARGET="armv7aQNX6.6.0gcc_cpp4.7.3" -DNDEBUG
	Debug	gcc -Vgcc/4.7.3,gcc_ntoarmv7le_cpp -lang-c -fPIC -fexceptions -DFD_SETSIZE=512 -g -Wall -Wno-unknown-pragmas -DPtrIntType=long -DCSREAL_IS_FLOAT -DCPU=ARMV7 -DTARGET="armv7aQNX6.6.0gcc_cpp4.7.3"
i86QNX6.6gcc_cpp4.7.3	Release	gcc -Vgcc/4.7.3,gcc_ntox86 -Y_cpp -m32 -march=i386 -mtune=generic -lang-c -fPIC -fexceptions -DFD_SETSIZE=512 -O -Wall -Wno-unknown-pragmas -DPtrIntType=long -DCSREAL_IS_FLOAT -DCPU=i80586 -DTARGET="i86QNX6.6gcc_cpp4.7.3" -DNDEBUG
	Debug	gcc -Vgcc/4.7.3,gcc_ntox86 -Y_cpp -m32 -march=i386 -mtune=generic -lang-c -fPIC -fexceptions -DFD_SETSIZE=512 -g -Wall -Wno-unknown-pragmas -DPtrIntType=long -DCSREAL_IS_FLOAT -DCPU=i80586 -DTARGET="i86QNX6.6gcc_cpp4.7.3"