

RTI Connex DDS

Core Libraries

**Custom Support for
QNX Neutrino 6.5.0 Platforms
on PPC CPUs**

Version 5.3.0



© 2017 Real-Time Innovations, Inc.

All rights reserved.

Printed in U.S.A. First printing.

June 2017.

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

Website: <https://support.rti.com/>

Custom Support for QNX Neutrino 6.5.0 Platforms

1 Supported Platforms

This document supplements the [RTI Connex DDS Core Libraries Release Notes](#) and [RTI Connex DDS Core Libraries Platform Notes](#) and provides information specifically for the platform listed below. Please see the *Platform Notes* for other information on using a QNX platform (such as thread configuration).

Table 1 Custom Supported QNX Neutrino 6.5.0 Platform

Operating System	CPU	Compiler	RTI Architecture Abbreviation
QNX® Neutrino® 6.5.0	PowerPC™ E500 V2 core	qcc (gcc 4.4.2)	ppce500v2QNX6.5.0qcc_cpp4.4.2

2 Features

These features are supported:

- Modern C++ API
- Multicast
- Monotonic clock
- Distributed Logger
- Request/Reply

These features are not supported:

- Durable Writer History and Durable Reader State
- Control of CPU core affinity for RTI threads
- Security Plugins

3 Transports

These transports are supported:

- Shared memory
- UDPv4
- UDPv6: Supported but 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.

These transports are not supported:

- TCP/IPv4
- Secure WAN
- TLS Support

4 Compiling and Running

[Table 2 Building Instructions for QNX Neutrino 6.5.0](#) lists the compiler flags and libraries you will need to link into your application.

[Table 3 Running Instructions for QNX Neutrino 6.5.0](#) shows the environment variables required to be set at run time.

[Table 4 Library-Creation Details for QNX Neutrino 6.5.0](#) 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.

Table 2 Building Instructions for QNX Neutrino 6.5.0

API	Library Format	RTI Libraries ^{a bc}	Required System Libraries	Required Compiler Flags
C++ (Traditional and Modern APIs)	Static Release	libnndscppz.a or libnndscpp2z.a libnndscz.a libnndscorez.a librticonnextmsgcppz.a	-lm -lsocket	-DRTI_QNX
	Static Debug	libnndscppzd.a or libnndscpp2zd.a libnndsczd.a libnndscorezd.a librticonnextmsgcppzd.a		
	Dynamic Release	libnndscpp.so or libnndscpp2.so libnndsc.so libnndscore.so librticonnextmsgcpp.so		
	Dynamic Debug	libnndscppd.so or libnndscpp2d.so libnndscd.so libnndscored.so librticonnextmsgcppd.so		
C	Static Release	libnndscz.a libnndscorez.a librticonnextmsgcz.a	-lm -lsocket	-DRTI_QNX
	Static Debug	libnndsczd.a libnndscorezd.a librticonnextmsgczd.a		
	Dynamic Release	libnndsc.so libnndscore.so librticonnextmsgc.so		
	Dynamic Debug	libnndscd.so libnndscored.so librticonnextmsgcd.so		

^aThe 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)

^bThe *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.

^cChoose libnndscpp*. for the Traditional C++ API or libnndscpp2*. for the Modern C++ API.

Table 3 Running Instructions for QNX Neutrino 6.5.0

RTI Architecture	Library Format (Release & Debug)	Environment Variables
ppce500v2QNX6.5.0qcc_cpp4.4.2	Static	None required
	Dynamic	LD_LIBRARY_PATH= \${NDDSHOME}/lib/<architecture>: \${LD_LIBRARY_PATH} ^a

Table 4 Library-Creation Details for QNX Neutrino 6.5.0

RTI Architecture	Library Format (Static & Dynamic)	Compiler Flags Used by RTI
ppce500v2QNX6.5.0qcc_cpp4.4.2	Release	qcc Vgcc/4.4.2.gcc_ntoppcbespe -Y_cpp -v -mcpu=8540 -me500v2 -mno-isel -mspe -mhard-float -Wl,-relax -fPIC -fexceptions -O -DNDEBUG
	Debug	qcc Vgcc/4.4.2.gcc_ntoppcbespe -Y_cpp -v -mcpu=8540 -me500v2 -mno-isel -mspe -mhard-float -Wl,-relax -fPIC -fexceptions -g

^a \${NDDSHOME} represents the root directory of your Connex DDS installation. \${LD_LIBRARY_PATH} represents the value of the LD_LIBRARY_PATH variable prior to changing it to support Connex DDS. When using nddsjava.jar, the Java virtual machine (JVM) will attempt to load release versions of the native libraries. When using nddsjavad.jar, the JVM will attempt to load debug versions of the native libraries.