RTI Connext DDS

Core Libraries

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

Version 5.3.0

Printed in U.S.A. First printing.
May 2017.

Trademarks

Copy and Use Restrictions

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 6.6.0 Platforms

1 Supported Platforms

This document supplements the RTI Connext DDS Core Libraries Release Notes and RTI Connext DDS Core Libraries Platform Notes. It provides information specifically for the platform listed in Table 1 Custom Supported QNX Neutrino 6.6.0 Platforms. Please see the *Platform Notes* for other information, such as thread configuration.

Table 1 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

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

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

4 Compiling and Running

Table 2 Building Instructions for QNX Neutrino 6.6.0 Architectures lists the compiler flags and libraries you will need to link into your application.

Table 3 Running Instructions for QNX Neutrino 6.6.0 Architectures shows the environment variables required to be set at run time.

Table 4 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.

Table 2 Building Instructions for QNX Neutrino 6.6.0 Architectures

API	Library Format	RTI Libraries[a] [bc]	Required System Libraries	Required Compiler Flags
C++ (Traditional and Modern APIs)	Static Release	libnddscppz.a or libnddscpp2z.a libnddscz.a libnddscorez.a librticonnextmsgcppz.a	-lm -lsocket	-DRTI_QNX
	Static Debug	libnddscppzd.a or libnddscpp2zd.a libnddsczd.a libnddscorezd.a librticonnextmsgcppzd.a		
	Dynamic Release	libnddscpp.so or libnddscpp2.so libnddsc.so libnddscore.so librticonnextmsgcpp.so		
	Dynamic Debug	libnddscppd.so or libnddscpp2d.so libnddscd.so libnddscored.so librticonnextmsgcppd.so		

[a]The C/C++ libraries are in $(NDDSHOME)/lib/<architecture>/. (where $(NDDSHOME) is where Connext DDS is installed, such as /home/your user name/rti_connext_dds-5.x.y)

[b]The *rticonnextmsg* library only applies if you have the *Connext DDS* Professional, Evaluation, or Basic package type. It is not provided with the *Connext DDS* Core package type.

[c]Choose libnddscpp*.* for the Traditional C++ API or libnddscpp2*.* for the Modern C++ API.

Table 2 Building Instructions for QNX Neutrino 6.6.0 Architectures

API	Library Format	RTI Libraries[a] [bc]	Required System Libraries	Required Compiler Flags
C	Static Release	libnddscz.a libnddscorez.a librticonnextmsgcz.a	-lm -lsocket	-DRTI_QNX
	Static Debug	libnddsczd.a libnddscorezd.a librticonnextmsgczd.a		
	Dynamic Release	libnddsc.so libnddscore.so librticonnextmsgc.so		
	Dynamic Debug	libnddscd.so libnddscored.so librticonnextmsgcd.so		

Table 3 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}[d]

[a]The C/C++ libraries are in $(NDDSHOME)/lib/<architecture>/. (where $(NDDSHOME) is where Connext DDS is installed, such as /home/your user name/rti_connext_dds-5.x.y)

[b]The *rticonnextmsg* library only applies if you have the *Connext DDS* Professional, Evaluation, or Basic package type. It is not provided with the *Connext DDS* Core package type.

[c]Choose libnddscpp*.* for the Traditional C++ API or libnddscpp2*.* for the Modern C++ API.

[d]${NDDSHOME} represents the root directory of your Connext DDS installation. ${LD_LIBRARY_PATH} represents the value of the LD_LIBRARY_PATH variable prior to changing it to support Connext 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.

Table 4 Library-Creation Details for QNX Neutrino 6.6.0 Architectures

RTI Architecture	Library Format *(Static & Dynamic)*	Compiler Flags Used by RTI
armv7aQNX6.6.0qcc_cpp4.7.3	Release	qcc -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=\"armv7aQNX6.6.0qcc_cpp4.7.3\" -DNDEBUG
	Debug	qcc -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=\"armv7aQNX6.6.0qcc_cpp4.7.3\"
i86QNX6.6qcc_cpp4.7.3	Release	qcc -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=\"i86QNX6.6qcc_cpp4.7.3\" -DNDEBUG
	Debug	qcc -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=\"i86QNX6.6qcc_cpp4.7.3\"