RTI Connext DDS

Core Libraries

Custom Support for RedHawk Linux 6.0 Platforms

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 RedHawk Linux 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 in Table 1 Custom Supported RedHawk Linux 6.0 Platforms .

Table 1 Custom Supported RedHawk Linux 6.0 Platforms

Operating System	CPU	Compiler	RTI Architecture Abbreviation
RedHawk Linux 6.0	x64	gcc 4.4.5	x64Linux2.6gcc4.4.5

2 Transports

These transports are supported:

- Shared memory: Supported and enabled by default
- UDPv4: Supported and enabled by default
- UDPv6. Not enabled by default and the peers list must be modified to support IPv6. Mapping of the TransportPriority QoS is supported.

These transports are not supported:

- TCP/IPv4
- RTI Secure WAN Transport

3 Features

These features are supported:

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

These features are not supported:

- Native POSIX Thread Library (NPTL)
- RTI Distributed Logger

4 Compiling and Running

Table 2 Building Instructions lists the compiler flags and libraries you will need to link into your application.

Table 3 Running Instructions shows the environment variables required to be set at run time.

Table 4 Library-Creation Details 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

API	Library Format	Required 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 librticonnextmsgcz.a	-ldl -lnsl -lm -lpthread -lrt	-DRTI_UNIX -m64
	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++/Java libraries are in <NDDSHOME>/lib/<architecture>; the jar files are in <NDDSHOME>/lib/java (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

API	Library Format	Required RTI Libraries[a] [bc]	Required System Libraries	Required Compiler Flags
C	Static Release	libnddscz.a libnddscorez.a include: librticonnextmsgcz.a	-ldl -lnsl -lm -lpthread -lrt	-DRTI_UNIX -m64
	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

RTI Architecture	Library Format (Release and Debug)	Environment Variables[d]
x64Linux2.6gcc4.4.5	Static	None required
	Dynamic	LD_LIBRARY_PATH= ${NDDSHOME}/lib/<architecture>: ${LD_LIBRARY_PATH}

[a]The C/C++/Java libraries are in <NDDSHOME>/lib/<architecture>; the jar files are in <NDDSHOME>/lib/java (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 is where Connext DDS is installed. $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

RTI Architecture	Library Format	Compiler Flags Used by RTI
i86Linux2.6gcc4.4.5	Release	gcc -m32 -fPIC -DLINUX -DRTI_GCC4 -DRTI_LINUX26 -DRTI_LINUX -DRTI_POSIX_THREADS -DRTI_POSIX_SEMAPHORES -DRTI_CPU_AFFINITY -O -Wall -Wno-unknown-pragmas -DRTI_UNIX -DPtrIntType=long -DCSREAL_IS_FLOAT -DCPU=I80586 -DRTI_ENDIAN_LITTLE -DRTI_THREADS -DRTI_MULTICAST -DRTI_SHARED_MEMORY -DRTI_IPV6 -DTARGET=\\"i86Linux2.6gcc4.4.5\\" -DNDEBUG -Wp,-MD
	Debug	gcc -m32 -fPIC -DLINUX -DRTI_GCC4 -DRTI_LINUX26 -DRTI_LINUX -DRTI_POSIX_THREADS -DRTI_POSIX_SEMAPHORES -DRTI_CPU_AFFINITY -g -DRTI_PRECONDITION_TEST -Wall -Wno-unknown-pragmas -DRTI_UNIX -DPtrIntType=long -DCSREAL_IS_FLOAT -DCPU=I80586 -DRTI_ENDIAN_LITTLE -DRTI_THREADS -DRTI_MULTICAST -DRTI_SHARED_MEMORY -DRTI_IPV6 -DTARGET=\\"i86Linux2.6gcc4.4.5\\" -Wp,-MD