

RTI Connex

Core Libraries and Utilities

Custom Support for Mistral Linux Platforms

Version 5.0.0



Your systems. Working as one.



© 2012 Real-Time Innovations, Inc.
All rights reserved.
Printed in U.S.A. First printing.
August 2012.

Trademarks

Real-Time Innovations, RTI, and Connexant are trademarks or registered trademarks of Real-Time Innovations, Inc. All other trademarks used in this document are the property of 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 Mistral Linux Platforms

1 Supported Platforms

This document supplements the *RTI® ConnexT™ Release Notes* and *Platform Notes*. It provides information specifically for the platform listed in [Table 1.1](#).

Table 1.1 Custom Supported Mistral Linux Platforms

Operating System	CPU	Compiler	RTI Architecture Abbreviation
Mistral Linux Kernel 2.6.32	ARMv7 (OMAP3530)	Sourcery G++ Lite 2009q3-67 gcc 4.4.1 (arm-none-linux-gnueabi-gcc 4.4.1)	armv7leLinux2.6gcc4.4.1

1.1 Supported Transports

Shared memory: Supported and enabled by default. To clean up shared memory resources, reboot the kernel.

UDpv4: Supported and enabled by default.

UDpv6: Not supported.

TCP/IPv4: Supported.

1.2 Supported Features

These features are supported:

- ☐ Multicast
- ☐ Monotonic clock

These features are not supported:

- ☐ Control of CPU core affinity for RTI threads

2 Compiling and Running

[Table 2.1](#) lists the compiler flags and libraries you will need to link into your application.

[Table 2.2](#) shows the environment variables required to be set at run time.

[Table 2.3](#) 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 2.1 **Building Instructions**

API	Library Format	Required RTI Libraries ^a	Required System Libraries	Required Compiler Flags
C++	Static Release	libniddscppz.a libniddscz.a libniddscorez.a <i>For Connexrt Messaging, also include:</i> librticonnextmsgcppz.a	-ldl -lnsl -lm -lpthread -lrt	-DRTI_UNIX
	Static Debug	libniddscppzd.a libniddsczd.a libniddscorezd.a <i>For Connexrt Messaging, also include:</i> librticonnextmsgcppzd.a		
	Dynamic Release	libniddscpp.so libniddsc.so libniddscore.so <i>For Connexrt Messaging, also include:</i> librticonnextmsgcpp.so		
	Dynamic Debug	libniddscppd.so libniddscd.so libniddscored.so <i>For Connexrt Messaging, also include:</i> librticonnextmsgcppd.so		
C	Static Release	libniddscz.a libniddscorez.a <i>For Connexrt Messaging, also include:</i> librticonnextmsgcz.a	-ldl -lnsl -lm -lpthread -lrt	-DRTI_UNIX
	Static Debug	libniddsczd.a libniddscorezd.a <i>For Connexrt Messaging, also include:</i> librticonnextmsgczd.a		
	Dynamic Release	libniddsc.so libniddscore.so <i>For Connexrt Messaging, also include:</i> librticonnextmsgc.so		
	Dynamic Debug	libniddscd.so libniddscored.so <i>For Connexrt Messaging, also include:</i> librticonnextmsgcd.so		

a. RTI C/C++ libraries are in `$(NDDSHOME)/lib/<architecture>/.`

Table 2.2 **Running Instructions**

RTI Architecture	Library Format	Environment Variables
armv7leLinux2.6gcc4.4.1	Static	None required
	Dynamic	LD_LIBRARY_PATH= <code>\$(NDDSHOME)/lib/<architecture>:</code> <code>\$(LD_LIBRARY_PATH)</code> ^a

a. \${NDDSHOME} represents the root directory of your *Connex* installation. \${LD_LIBRARY_PATH} represents the value of the LD_LIBRARY_PATH variable prior to changing it to support *Connex*.

Table 2.3 **Library-Creation Details**

RTI Architecture	Library Format	Compiler Flags Used by RTI
armv7leLinux2.6gcc4.4.1 ^a	Release	-mthumb -march=armv7-a -mfp=neon -msoft-float -fpic -DLINUX -O -Wall -Wno-unknown-pragmas -DRTS_UNIX DPtrIntType=long -DCPU=ARMV7 -DTARGET=\"armv7leLinux2.6gcc4.4.1\" -DNDEBUG -c -Wp,-MD
	Debug	-mthumb -march=armv7-a -mfp=neon -msoft-float -fpic -DLINUX -g -Wall -Wno-unknown-pragmas -DRTS_UNIX DPtrIntType=long -DCPU=ARMV7 -DTARGET=\"armv7leLinux2.6gcc4.4.1\" -c -Wp,-MD

a. The C++ libniddscpp dynamic libraries were linked using g++; the C dynamic libraries, i.e., libniddscore and libniddsc, were linked using gcc.

3 Known Issues

None.