RTI Secure WAN Transport

Core Libraries and Utilities

Release Notes

Version 5.1.0

Your systems. Working as one.

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/

Release Notes

1 Supported Platforms

This release of *RTI® Secure WAN Transport* is supported on the architectures listed in Table 1.1.

Table 1.1 **Supported Platforms**

Operating System		CPU	Compiler	RTI Architecture Abbreviation
Linux	Red Hat® Enterprise Linux® 5.0 (2.6 kernel)	x86	gcc 4.1.1	i86Linux2.6gcc4.1.1
			Java Platform, Standard Edition JDK 1.7	i86Linux2.6gcc4.1.1jdk
		x64	gcc 4.1.1	x64Linux2.6gcc4.1.1
			Java Platform, Standard Edition JDK 1.7	x64Linux2.6gcc4.1.1jdk
	Ubuntu® Server 12.04 LTS	x86	gcc 4.6.3	i86Linux3.xgcc4.6.3
			Java Platform, Standard Edition JDK 1.7	i86Linux3.xgcc4.6.3jdk
		x64	gcc 4.6.3	x64Linux3.xgcc4.6.3
			Java Platform, Standard Edition JDK 1.7	x64Linux3.xgcc4.6.3jdk
	Wind River® Linux 4 (2.6 kernel)	x64	gcc 4.4.1	x64WRLinux2.6gcc4.4.1
QNX	QNX® Neutrino® 6.5	x86	qcc 4.4.2 with GNU C++ libraries	i86QNX6.5qcc_gpp4.4.2
Solaris	Solaris™ 2.10	Ultra SPARC®	gcc3.4.2	sparcSol2.10gcc3.4.2
			Java Platform, Standard Edition JDK 1.7	sparcSol2.10jdk
Windows	Windows® 2003 Windows Vista® (32-bit Edition) Windows XP Professional	x86	Visual Studio 2005 SP 1	i86Win32VS2005
			Visual Studio 2005 SP 1 (C++/CLI, C# 8.0 or 9.0)	i86Win32dotnet2.0
			Java Platform, Standard Edition JDK 1.7	i86Win32jdk

Table 1.2 lists an additional target library for which RTI offers custom support. If you are interested in using this platform, please contact your local RTI representative or email **sales@rti.com**.

Table 1.2 **Custom Supported Platforms**

Operating System	CPU	Compiler	RTI Architecture Abbreviation
NI Linux Real-Time 3.2	ARM Cortex-A9	gcc 4.4.1	armv7AngstromLinux3.2gcc4.4.1.cortex-a9

2 Compatibility

RTI Secure WAN Transport is an optional product for use with *RTI Connext*™ (formerly, *RTI Data Distribution Service*) with the same version number.

3 What's New in 5.1.0

3.1 New Platforms

This release adds support for the following platforms:

- QNX Neutrino 6.5.0 SP1
- Ubuntu Server 12.04 LTS
- Wind River Linux 4 (2.6 kernel)

3.2 New Default Transport Settings

Some of the default settings for a number of the provided transport plugins have changed to provide better out-of-the-box performance. By increasing the out-of-the-box performance, in most cases you will not have to modify these settings or keep them in sync across all *Connext* applications and services within your system.

The default values for **message_size_max**, **send_socket_buffer_size**, and **recv_socket_buffer_size** have changed. Table 3.1 shows the old and new default values.

Table 3.1 **WAN Transport Changes**

	Old Default (bytes)	New Default (bytes)	
		Non-INTEGRITY Platforms	INTEGRITY Platforms[1]
message_size_max	9216	65507[2]	9216
send_socket_buffer_size	9216	131072	131072
recv_socket_buffer_size	9216	131072	131072

[1] Due to limits imposed by the INTEGRITY platform, the new default settings for all INTEGRITY platforms are treated differently than other platforms. Please see the *RTI Core Libraries and Utilities Platform Notes* for more information on the issues with increasing the **message_size_max** default values on INTEGRITY platforms. Notice that interoperation with INTEGRITY platforms will require updating the transport property **message_size_max** so that it is consistent across all platforms.

[2] The value 65507 represents the maximum user payload size that can be sent as part of a UDP packet.

4 Compatibility

In *Connext* 5.1.0, the default value for **message_size_max** for this transport has changed. *Secure WAN Transport* also uses this value. Consequently, *Secure WAN Transport* 5.1.0 is not off-the-shelf compatible with applications running older versions of this transport. See the *Core Libraries and Utilities Release Notes* for instructions on how to resolve the compatibility issue with older *Connext* and *RTI Data Distribution Service* applications.

5 What's Fixed in 5.1.0

5.1 Error "dtls message too big" when Sending Data over DTLS

When sending data over the DTLS transport, you may have seen the following error message:

```
NDDS_Transport_DTLS_Connection_send:OpenSSL error:
[1410c13e]:[DTLS1_WRITE_APP_DATA_BYTES]:[dtls message too big]
```

This message happens because the OpenSSL DTLS maximum message size is 16384 bytes. This issue has been resolved by enforcing that the DTLS transport property **parent.message_size_max** must be no more than 16384 bytes, therefore preventing the error from ever happening.

[RTI Issue ID COREPLG-39]

5.2 Unable to Set DTLS send_socket_buffer_size to Less than Default

When using the DTLS transport, setting the DTLS **send_socket_buffer_size** to a value less than the default of 9216 would result in the following errors and **create_participant()** would fail:

```
[D0000|ENABLE]NDDS_Transport_UDPv4_Property_verify:send_socket_buffer_size
< message_size_max
[D0000|ENABLE]NDDS_Transport_UDPv4_newI:Invalid transport properties.
[D0000|ENABLE]NDDS_Transport_DTLS_newI:!Failed to allocate base UDPv4
transport
[D0000|ENABLE]DDS_DomainParticipantConfigurator_setup_custom_transports:!cr
eate custom transport plugin
[D0000|ENABLE]DDS_DomainParticipantConfigurator_enable:!install transport
plugin aliases = custom transports
[D0000|ENABLE]DDS_DomainParticipant_enableI:!enable transport configurator
DDSDomainParticipant_impl::createI:ERROR: Failed to auto-enable entity
DomainParticipantFactory_impl::create_participant():!create failure creat-
ing participant
```

These errors would occur even if you set the DTLS **message_size_max** to be less than or equal to the value of the DTLS **send_socket_buffer_size**. This issue has been resolved.

[RTI Issue ID COREPLG-171]

5.3 Memory Leak in DTLS Transport

A graceful shutdown of a *DomainParticipant* that was using the DTLS transport and had communicated with other *DomainParticipants* over DTLS may not have properly freed all DTLS transport resources, resulting in a memory leak. This issue has been resolved.

[RTI Issue ID COREPLG-172]

5.4 Failed to Reopen Closed Connections in Asymmetric TCP

When a large number of clients connected to an asymmetric TCP server, some connections were closed and never reopened, preventing the discovery of DDS entities and/or the delivery of user data. This problem has been resolved.

[RTI Issue ID COREPLG-178]

6 Available Documentation

The following documentation is provided with the *Connext* distribution. (The paths show where the files are located after *Connext* has been installed in **<NDDSHOME>**.)

❑ *RTI Secure WAN Transport Installation Guide* (**<NDDSHOME>/doc/pdf/RTI_Secure_WAN_InstallationGuide.pdf**, also available for download from RTI's Customer Portal.)

❑ *RTI Core Libraries and Utilities User's Manual* (**<NDDSHOME>/doc/pdf/RTI_CoreLibrariesAndUtilities_UsersManual.pdf**)

❑ *RTI Secure WAN Transport* API Reference HTML documentation*:*
Open **<NDDSHOME>/ReadMe.html**, then select *Secure WAN Transport*.

❑ Example code: **<NDDSHOME>/example/<language>/helloWorldWAN**.

7 Known Issues

❑ When communicating over some networks, the WAN and Secure Transport plug-ins may fail to send data larger than the MTU (maximum transmission unit) size available for the network. This is especially likely over wide-area networks. This scenario is also a suggested configuration of the DTLS protocol, according to the DTLS specification, which is IETF RFC 4347.

If problems occur while sending large packets, set the maximum_message_size transport property to the MTU of your network *minus 28 bytes for the DTLS header* and set up your application according to the Large Data Use Cases "How To" provided in the online (HTML) documentation. For example, for an MTU size of 1500 bytes (for standard Ethernet), set **maximum_message_size** to 1500 - 28 = 1472.

One instance of this problem for which there is no workaround is the case where the discovery packets are larger than your network's MTU. This could occur if user data, propagated properties, or type-codes are configured.

❑ An application using the WAN transport may appear to hang for several minutes if the WAN server is shut down and not restarted before the application tries to contact it, or if the application is unable to communicate with the WAN server.

Two scenarios under which the application tries to contact the STUN server are during shut down and while establishing a connection with the initial peers.

This issue is due to a sequence of synchronous STUN transactions with the STUN server. If you need to run WAN transport without a STUN server, here are some recommendations:

- Decrease the blocking time by decreasing the number of STUN retransmissions. To do so, change the property, **stun_number_of_retransmissions**. For example, a change from the default of 7 retries to 5 retries will result in a total period of 3.1 seconds per synchronous operation. Note however, that this may impact the ability to reliably set up connections to peers over a WAN.
- Decrease the blocking time by using a participant ID limit of zero when configuring the initial peer descriptors.

For example, when the peer descriptor **wan://::1:10.10.1.150** is specified, DDS will try to contact five participants with the same WAN ID in different ports. Usually there is only one participant using the same WAN ID. Although the other four participants will never be reachable, the application still tries to establish communication with them by contacting the STUN server.

You can reduce the number of participants to which the application will try to contact to one by using a participant ID limit of zero in the peer descriptor. For example, **0@wan://::1:10.10.1.150**.

For additional information on peer descriptors see the Discovery chapter in the *RTI Core Libraries and Utilities User's Manual*.

8 Third-Party Licenses

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment:

   This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.openssl.org/)
4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.
5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project.
6. Redistributions of any form whatsoever must retain the following acknowledgment:

   This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).