RTI Secure WAN Transport

Release Notes

Version 6.0.1

Printed in U.S.A. First printing.
November 2019.

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/

Contents

Chapter 1 Supported Platforms

This release of *RTI® Secure WAN Transport* is supported on the platforms in Table 1.1 Supported Platforms.

For details on these platforms, see the *RTI Connext DDS Core Libraries Platform Notes*.

Table 1.1 Supported Platforms

Operating System	Version
Android®	All Android platforms listed in the *RTI Connext® DDS Core Libraries Release Notes* for the same version number. Note: RTI WAN Server is not supported.
iOS®	All iOS platforms listed in the *RTI Connext DDS Core Libraries Release Notes* for the same version number. Note: RTI WAN Server is not supported.
Linux®	All Linux platforms in the *RTI Connext DDS Core Libraries Release Notes* for the same version number, except SUSE® Linux Enterprise Server 11 and 12.
macOS®	All macOS platforms listed in the *RTI Connext DDS Core Libraries Release Notes* for the same version number.
QNX®	All QNX Neutrino® 6.5 and higher platforms listed in the *RTI Connext DDS Core Libraries Release Notes* for the same version number.
Windows®	All Windows platforms listed in the *RTI Connext DDS Core Libraries Release Notes* for the same version number.

Secure WAN Transport is also supported on the platforms in Table 1.2 Custom Supported Platforms; these are target platforms for which RTI offers custom support. If you are interested in these platforms, please contact your local RTI representative or email sales@rti.com.

Table 1.2 Custom Supported Platforms

Operating System	Version
Linux	RedHawk™ Linux 6.5 on x86 and x64 CPUs Wind River® Linux 8 on Arm v7 CPU Yocto Project® 2.5 on Arm v7 CPU

Chapter 2 Compatibility

RTI Secure WAN Transport is an optional product for use with *RTI Connext® DDS* with the same version number.

Secure WAN Transport 6.0.1 is API-compatible with OpenSSL® versions 1.1.1a through 1.1.1d. It is not API-compatible with OpenSSL® 1.1.0 or below. Note that *Secure WAN Transport* 6.0.1 has only been tested by RTI using OpenSSL 1.1.1d. If you need *Secure WAN Transport* 6.0.1 to run against older versions of OpenSSL®, please contact support@rti.com.

If you were using OpenSSL 1.0.1: Because *RTI Connext DDS* 5.2.3 and higher uses OpenSSL 1.0.2 or higher, the number of bits of any Diffie-Hellman (DH) parameters must now be at least 1024 (see https://www.openssl.org/blog/blog/2015/05/20/logjam-freak-upcoming-changes/). Therefore, if you are using the property **tls.cipher.dh_param_files** and there is a DH parameter file that has fewer than 1024 bits, you must regenerate the file with at least 1024 bits.

For backward compatibility information between 6.0.1 and previous releases, see the *Migration Guide* on the RTI Community Portal (https://community.rti.com/documentation).

Chapter 3 What's New in 6.0.1

3.1 New platforms

This release adds support for the following platforms:

- Android 9 (Arm v7, Arm v8 64-bit)
- macOS 10.14 (x64)
- Red Hat Enterprise Linux 8 (x64)
- Windows 10 (x86, x64) with Visual Studio® 2019
- Windows Server 2016 (x86, x64) with Visual Studio 2019
- Yocto Project 2.5 (Arm v7)

3.2 Removed platforms

The following platforms are no longer supported:

- macOS 10.11
- Windows 7
- Windows Server 2008 R2

3.3 Updated OpenSSL Version

This release uses OpenSSL 1.1.1d (instead of 1.0.2o).

Chapter 4 Previous Release

4.1 What's New in 6.0.0

4.1.1 New Platforms

This release added support for the following platforms:

Operating System	CPU	Compiler	RTI Architecture Abbreviation
Ubuntu 18.04 LTS	x64	gcc 7.3.0	x64Linux4gcc7.3.0
Wind River Linux 8	Arm v7	gcc 5.2.0	armv7aWRLinux8gcc5.2.0 (custom target platform)

See the *RTI Connext DDS Core Libraries Platform Notes* for details.

4.1.2 New APIs

New APIs are provided to get the library version number of *Secure WAN*:

- NDDS_Transport_WAN_get_library_version()
- NDDS_Transport_WAN_get_build_version_string()

4.1.3 Updated OpenSSL Version

This release uses OpenSSL 1.0.2o (instead of 1.0.2n).

4.2 What's Fixed in 6.0.0

4.2.1 Possible segmentation fault in WAN transport during participant deletion

The WAN transport may have crashed during participant deletion. In particular, this issue was only triggered after logging at least one message from the WAN transport threads.

This problem has been resolved. The WAN transport no longer crashes as a result of this issue.

[RTI Issue ID COREPLG-399]

4.2.2 Changes in hello world dtls example to simplify how static linking is enabled

The hello world dtls example has been updated to simplify how static linking is enabled. Specifically, previous releases of this example required you to define the "USE_STATIC_LINK" variable to link statically. Now this step is no longer required.

[RTI Issue ID COREPLG-430]

4.2.3 Wrong exception printed when using create_function_ptr property

The WAN transport plugin's **create_function_ptr** property was not properly validated. Therefore, the following message was printed when that property was used:

```
[D0064|ENABLE] NDDS_Transport_WAN_plugin_property_from_DDS_property:Unexpected property:
dds.transport.wan_plugin.wan.create_function_ptr. Closest valid property:
dds.transport.wan_plugin.wan.create_function
```

Note that, despite the exception, the property did have effect.

This problem has been resolved: the unexpected log message no longer appears.

[RTI Issue ID COREPLG-451]

Chapter 5 Known Issues

- When communicating over some networks, the *Secure WAN 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 information on peer descriptors, see the *Discovery* chapter in the *RTI Connext DDS Core Libraries User's Manual*.

Chapter 6 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).