RTI Secure WAN Transport

Release Notes

Version 6.1.0

Printed in U.S.A. First printing.
April 2021.

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*.

Note: POSIX®-compliant architectures that end with "FACE_GP" are not supported.

Table 1.1 Supported Platforms

Operating System	Version
Android™ *Available on demand*	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.
Linux®	All Linux platforms in the *RTI Connext DDS Core Libraries Release Notes* for the same version number, except SUSE® Linux Enterprise Server.
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 (x86 and x64) Wind River® Linux 8 (Arm® v7) Yocto Project® 2.5 (Arm v8)

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 6.1.0 is API-compatible with OpenSSL® versions 1.1.1a through 1.1.1k. It is not API-compatible with OpenSSL® 1.1.0 or below. Note that *Secure WAN Transport* 6.1.0 has only been tested by RTI using OpenSSL 1.1.1k. If you need *Secure WAN Transport* 6.1.0 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 6.1.0 and previous releases, see the *Migration Guide* on the RTI Community Portal (https://community.rti.com/documentation).

Chapter 3 What's New in 6.1.0

3.1 New Platforms

This release adds support for the following platforms:

- macOS 10.15 (x64)
- QNX Neutrino 7.0.4 (64-bit Arm v8 and x64)
- Red Hat® Enterprise Linux 7.6 (x64)
- Ubuntu® 18.04 LTS (Arm v7 and v8)
- Ubuntu 20.04 LTS (x64)
- Yocto Project 2.5 (Arm v8) (Custom-supported target platform)

3.2 Removed Platforms

The following platforms are no longer supported:

- Android 5.0, 5.1
- iOS
- macOS 10.12
- SUSE Linux Enterprise Server 11
- Ubuntu 12.04 LTS
- Wind River Linux 7

3.3 Possible Future Deprecation of Secure WAN Transport

RTI may not support *Secure WAN Transport* in future versions of *Connext DDS*. Existing applications that use *Secure WAN Transport* should be updated to take advantage of *RTI Real-Time WAN Transport* as soon as feasible. All new applications should use *Real-Time WAN Transport*.

3.4 Updated OpenSSL Version

This release uses OpenSSL 1.1.1k (instead of 1.1.1d).

3.5 Target OpenSSL Bundles Distributed as .rtipkg Files

Target OpenSSL bundles are now distributed as **.rtipkg** files. Once installed, the OpenSSL files are available in **<installation_folder>/third_party**.

3.6 Changes to OpenSSL Static Library Names

The OpenSSL static library names no longer have a "z" suffix. **libcryptoz** has been renamed to **libcrypto**, and **libsslz** has been renamed to **libssl**. When including the static libraries in a makefile, we recommend including the whole path to the OpenSSL static libraries in order to avoid confusion with the dynamic libraries. Here is an example:

```
gcc -o myApp myApp.o -L$NDDSHOME/lib/$ARCH -lnddstransporttlsz -lnddscz -lnddscorez $RTI_
OPENSSLHOME/$ARCH/release/lib/libssl.a $RTI_OPENSSLHOME/$ARCH/release/lib/libcrypto.a
```

In addition, the Android static library **librtisslsupportz** has been removed. You may use **libcrypto** and **libssl** instead.

Chapter 4 What's Fixed in 6.1.0

4.1 Still reachable memory leaks

After shutting down an application using (D)TLS, memory profilers, such as Valgrind™, may have reported memory leaks categorized as still reachable memory leaks. These leaks were harmless and could not lead to unbounded memory growth. This problem has been fixed.

[RTI Issue ID COREPLG-510]

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*.

- If you load any dynamic libraries, you may see "still reachable" memory leaks in "dlopen" and "dlclose". These leaks are a result of a bug in Valgrind (https://bugs.launchpad.net/ubuntu/+source/valgrind/+bug/1160352) [RTI Issue ID COREPLG-510].