

RTI Secure WAN Transport

Release Notes

Version 6.1.1



© 2022 Real-Time Innovations, Inc.
All rights reserved.
Printed in U.S.A. First printing.
March 2022.

Trademarks

RTI, Real-Time Innovations, Connext, NDDS, the RTI logo, 1RTI and the phrase, “Your Systems. Working as one,” are registered trademarks, trademarks or service marks of Real-Time Innovations, Inc. All other trademarks belong to 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.

This is an independent publication and is neither affiliated with, nor authorized, sponsored, or approved by, Microsoft Corporation.

The security features of this product include software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>). This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Notice

Any deprecations noted in this document serve as notice under the Real-Time Innovations, Inc. Maintenance Policy #4220.

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	1
Chapter 2 Compatibility	3
Chapter 3 What's New in 6.1.1	
3.1 Secure WAN Transport now Deprecated	4
3.2 New Platforms	4
3.3 Third-Party Software Upgrades	4
Chapter 4 What's Fixed in 6.1.1	
4.1 Significant performance regression on Windows systems when using OpenSSL 1.1.1k libraries provided in 6.1.0	5
Chapter 5 Previous Release	
5.1 What's New in 6.1.0	6
5.1.1 New Platforms	6
5.1.2 Removed Platforms	6
5.1.3 Possible Future Deprecation of Secure WAN Transport	7
5.1.4 Updated OpenSSL Version	7
5.1.5 Target OpenSSL Bundles Distributed as .rtipkg Files	7
5.1.6 Changes to OpenSSL Static Library Names	7
5.2 What's Fixed in 6.1.0	7
5.2.1 Still reachable memory leaks	7
Chapter 6 Known Issues	8

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 Connex 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™ <i>Available on demand</i>	All Android platforms listed in the <i>RTI Connex DDS Core Libraries Release Notes</i> for the same version number. Note: RTI WAN Server is not supported.
Linux®	All Linux platforms in the <i>RTI Connex DDS Core Libraries Release Notes</i> for the same version number, except SUSE® Linux Enterprise Server.
macOS®	All macOS platforms listed in the <i>RTI Connex DDS Core Libraries Release Notes</i> for the same version number.
QNX®	All QNX Neutrino® 6.5 - 7.0.4 platforms listed in the <i>RTI Connex DDS Core Libraries Release Notes</i> for the same version number.
Windows®	All Windows platforms listed in the <i>RTI Connex DDS Core Libraries Release Notes</i> 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 Intel x86 and x64 CPUs (available on demand) 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 Connex[®] DDS* with the same version number.

Secure WAN Transport 6.1.1 is API-compatible with OpenSSL[®] versions 1.1.1 through 1.1.1n, but not API-compatible with OpenSSL 1.1.0l or below. Note that *Secure WAN Transport* 6.1.1 has only been tested by RTI using OpenSSL 1.1.1n. If you need *Secure WAN Transport* 6.1.1 to run against older versions of OpenSSL[®], please contact support@rti.com.

If you were using OpenSSL 1.0.1: Because *RTI Connex[®] 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 6.1.1 and previous releases, see the *Migration Guide* on the RTI Community Portal (<https://community.rti.com/documentation>).

Chapter 3 What's New in 6.1.1

3.1 Secure WAN Transport now Deprecated

Secure WAN Transport is deprecated starting with release 6.1.1. Support for *Secure WAN Transport* will be removed in a future release. You should use *RTI Real-Time WAN Transport* instead. See the *RTI Real-Time WAN* part of the [RTI Connex DDS Core Libraries User's Manual](#) for more information.

This deprecation serves as notice under the Real-Time Innovations, Inc. Maintenance Policy #4220.

3.2 New Platforms

This release adds support for these platforms:

- macOS 11 (Big Sur) on Arm v8 CPUs (arm64Darwin20clang12.0)
- macOS 11 (Big Sur) on x64 CPUs: We have validated that the same libraries for macOS 10.x (x64Darwin17clang9.0) can be used on macOS 11 systems.

For information on these platforms, see the *RTI Connex DDS Core Libraries Platform Notes* for this release.

3.3 Third-Party Software Upgrades

This release of *Secure WAN Transport* uses OpenSSL 1.1.1n (the previous release used 1.1.1k).

Chapter 4 What's Fixed in 6.1.1

4.1 Significant performance regression on Windows systems when using OpenSSL 1.1.1k libraries provided in 6.1.0

Previously, OpenSSL was built using compiler flags that enabled the usage of assembly instructions for certain operations on certain operating systems like Windows 64-bit (but not 32-bit).

The OpenSSL 1.1.1k libraries for Windows systems, provided with *Connnext DDS 6.1.0*, were missing those compiler flags. This resulted in degraded performance in *Secure WAN Transport*, which relies on those libraries.

This problem has been fixed, since this release uses OpenSSL 1.1.1n (see [3.3 Third-Party Software Upgrades on page 4](#)).

[RTI Issue ID COREPLG-564]

Chapter 5 Previous Release

5.1 What's New in 6.1.0

5.1.1 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)

5.1.2 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

5.1.3 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*.

5.1.4 Updated OpenSSL Version

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

5.1.5 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.

5.1.6 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 **librttislsupportz** has been removed. You may use **libcrypto** and **libssl** instead.

5.2 What's Fixed in 6.1.0

5.2.1 Still reachable memory leaks

After shutting down an application using (D)TLS, memory profilers, such as ValgrindTM, 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 6 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].