RTI TLS Support

Release Notes

Version 6.1.2

Printed in U.S.A. First printing.
December 2022.

Trademarks

Copy and Use Restrictions

Notice

Any deprecations or removals noted in this document serve as notice under the Real-Time Innovations, Inc. Maintenance Policy #4220 and/or any other agreements by and between RTI and customer regarding maintenance and support of RTI's software.

Deprecated means that the item is still supported in the release, but will be removed in a future release. *Removed* means that the item is discontinued or no longer supported. By specifying that an item is deprecated in a release, RTI hereby provides customer notice that RTI reserves the right after one year from the date of such release and, with or without further notice, to immediately terminate maintenance (including without limitation, providing updates and upgrades) for the item, and no longer support the item, in a future release.

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

1 Supported Platforms

This release of *RTI® TLS Support* 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®	All Android platforms listed in the *RTI Connext® DDS Core Libraries Release Notes* for the same version number.
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, except armv8QNX7.1qcc_gpp8.3.0.
Windows®	All Windows platforms listed in the *RTI Connext DDS Core Libraries Release Notes* for the same version number.

TLS Support 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	CPU	RTI Architecture Abbreviation
Linux	RedHawk™ Linux 8.2.1	x64	x64RedHawk68.2gcc8.3.1
	TI Linux 8.2.0.3	Arm v8	armv8Linux-armgcc9.2.1
	Yocto Project® 2.5	Arm v8	armv8Linux4gcc7.3.0
QNX	QNX Neutrino 6.6	Arm v7	armv7aQNX6.6.0qcc_cpp4.7.3
		x86	i86QNX6.6qcc_cpp4.7.3
	QNX Neutrino 7.0.4	Arm v7	armv7QNX7.0.0qcc_cxx5.4.0 [a]

[a] armv7QNX7.0.0qcc_cxx5.4.0 was tested with QNX Neutrino 7.0.0 kernel.

2 Compatibility

TLS Support is designed for use with the TCP transport that is included with *RTI Connext DDS*. If you choose to use *TLS Support*, it must be installed on top of an existing *TLS Support* installation with the same version number. It can only be used on architectures that support the TCP transport (see the *RTI Connext DDS Core Libraries Platform Notes*).

TLS Support 6.1.2 is API-compatible with OpenSSL® versions 1.1.0 through 1.1.1n, not with versions earlier than OpenSSL 1.1.0. Note that *TLS Support* 6.1.2 has only been tested by RTI using OpenSSL 1.1.1n. If you need *TLS Support* 6.1.2 to run against older versions of OpenSSL, please contact support@rti.com.

TLS Support 6.1.2 uses TLS 1.3. When communicating with *TLS Support* 6.0.0 or below, *TLS Support* 6.1.2 uses TLS 1.1.

If you are upgrading from OpenSSL 1.0.1 to OpenSSL 1.0.2 or above: 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 6.1.2 and previous releases, see the *Migration Guide* on the RTI Community Portal (https://community.rti.com/documentation).

3 What's New in 6.1.2

3.1 New Platforms

This release adds support for these platforms:

OS	Version	CPU	RTI Architecture Abbreviation
Linux	RedHawk Linux 8.2.1 (custom-supported platform)	x64	x64RedHawk8.2gcc8.3.1
	TI Linux 8.2.0.3 (custom-supported platform)	Arm v8	armv8Linux-armgcc9.2.1
	Ubuntu 22.04	Arm v8	armv8Linux4gcc7.3.0
		x64	x64Linux4gcc7.3.0
macOS	macOS 12	x64	x64Darwin17clang9.0
QNX	QNX Neutrino 7.1	x64	x64QNX7.1qcc_cxx8.3.0
		Arm v8	armv8QNX7.1qcc_cxx8.3.0
Windows	Windows 11	x64	x64Win64VS2017

4 What's Fixed in 6.1.2

4.1 Using dh_param_files Caused Memory Leak

Using the property **tls.cipher.dh_param_files** caused a memory leak when deleting the *DomainParticipant*. A memory checking tool, such as Valgrind, would have reported the leak in the OpenSSL function **PEM_read_bio_DHparams**, which is called by the RTI function **RTITLS_tmp_dhparam_callback()**. This problem only affected applications using OpenSSL 1.0.2 or applications communicating with applications using OpenSSL 1.0.2. For example, *TLS Support* 5.3 uses OpenSSL 1.0.2, but version 7.0.0 of *TLS Support* could still communicate with version 5.3, so the leak could also happen in version 7.0.0.

This problem has been fixed; memory will no longer be leaked in this scenario. For example, if *TLS Support* 6.1.2 communicates with an application using OpenSSL 1.0.2, the leak will not occur.

[RTI Issue ID COREPLG-641]

4.2 Memory Leak after Failure to Load String-Based Private Key

If you set the property **tls.identity.private_key** or **tls.identity.rsa_private_key**, and you specified a wrong or missing value for the property **tls.identity.private_key_password** or specified a mal-formed private key, there was a memory leak upon *DomainParticipant* creation failure. A memory checking tool, such as Valgrind, would report the leak in the OpenSSL function **BIO_new_mem_buf()**, which is called by the RTI function **RTITLS_context_init()**. This problem has been fixed. Memory will no longer be leaked in this scenario.

[RTI Issue ID COREPLG-643]

5 Previous Releases

5.1 What's New in 6.1.1

5.1.1 New platform

This release adds support for the following new platform.

Table 1 Added Platforms

Operating System	CPU	Compiler	RTI Architecture Abbreviation
macOS 11	Arm v8	clang 12.0	arm64Darwin20clang12.0

5.1.2 Third-party software upgrade

This release of *TLS Support* uses OpenSSL® 1.1.1n (the previous release used 1.1.1k).

5.2 What's Fixed in 6.1.1

5.2.1 hello_world_tcp example root and intermediate CAs expired too early

In **rti_workspace/examples/connext_dds/c/hello_world_tcp**, the README.txt states:

```
Example certificates for two peers are included in dds_security/cert/tls_rsa01.
```

But the root CA certificate **ca/rsa01RootCaCert.pem**, which was the intended command-line argument for **--tls-cert**, expired only 30 days after it was created. The root CA certificate was therefore unusable and led to communication failure along with the following errors:

```
RTITLS_ConnectionEndpointTLSv4_doHandshake:OpenSSL protocol error:1416F086:SSL
routines:tls_process_server_certificate:certificate verify failed
RTITLS_ConnectionEndpointTLSv4_doHandshake:OpenSSL protocol error:14094415:SSL
routines:ssl3_read_bytes:sslv3 alert certificate expired
```

In the identities folder, the files **rsa01Peer01.pem** and **rsa01Peer02.pem** have intermediate CAs in them, and those intermediate CAs have also expired.

These problems only affected release 6.1.0 and have been fixed by changing all of the certificates in **dds_security/cert/tls_rsa01**. The root CA and intermediate CA certificates now expire in 5 years instead of 30 days.

[RTI Issue ID COREPLG-554]

5.2.2 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 *Connext DDS* 6.1.0, were missing those compiler flags. This resulted in degraded performance in *TLS Support* for the TCP transport, which relies on those libraries.

This problem has been fixed, as this release uses OpenSSL 1.1.1n (see 1.2 Third-Party Software Upgrade on page 1).

[RTI Issue ID COREPLG-565]

5.3 What's New in 6.1.0

5.3.1 Added platforms

This release adds support for these platforms:

- macOS 10.15 (x64) (x64Darwin17clang9.0)
- QNX Neutrino 7.0.4 (Arm v8) (armv8QNX7.0.0qcc_gpp5.4.0, armv8QNX7.0.0qcc_cxx5.4.0)
- QNX Neutrino 7.0.4 (x64) (x64QNX7.0.0qcc_gpp5.4.0, x64QNX7.0.0qcc_cxx5.4.0)
- QNX Neutrino 7.0.4 (Arm v7) (custom supported platform armv7QNX7.0.0qcc_cxx5.4.0)
- Red Hat® Enterprise Linux 7.6 (x64) (x64Linux3gcc4.8.2)
- Ubuntu® 18.04 LTS (Arm v7) (armv7Linux4gcc7.5.0)
- Ubuntu 18.04 LTS (Arm v8) (armv8Linux4gcc7.3.0)
- Ubuntu 20.04 LTS (x64) (x64Linux4gcc7.3.0)
- Yocto Project 2.5 (Arm v8) (custom supported platform armv8Linux4gcc7.3.0)

5.3.2 Removed platforms

These platforms are no longer supported:

- Android™ 5.0, 5.1
- Debian 7 (custom supported platform)
- iOS®
- macOS 10.12
- Ubuntu 12.04 LTS
- Wind River Linux 7

5.3.3 Updated OpenSSL version

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

5.3.4 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**.

5.3.5 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 -lnddstransporttcpz -lnddstlsz -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.

5.4 What's Fixed in 6.1.0

This section describes bugs fixed in 6.1.0. These fixes have been made since 6.0.1 was released.

5.4.1 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]

5.4.2 No way to configure TLS 1.3 ciphers

The property **tls.cipher.cipher_list** applies only to TLS 1.2 communication, which occurs when either *DomainParticipant* is using a *Connext DDS* version older than 6.0.1. When both *DomainParticipants* are using *Connext DDS* 6.0.1 or later, they use TLS 1.3 communication, and the **tls.cipher.cipher_list** property does not apply. There was no way to configure the list of ciphers to be used when using TLS 1.3. This problem has been fixed by introducing a new property, **tls.cipher.ciphersuites**. See the OpenSSL manual page for SSL_CTX_set_ciphersuites for more information on the format of this string.

[RTI Issue ID COREPLG-534]

6 Known Issues

6.1 Possible Valgrind 'still-reachable' Leaks when Loading Dynamic Libraries

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).

This issue affects the *Core Libraries*, *Security Plugins*, *Secure WAN*, and *TLS Support*.

[RTI Issue IDs CORE-9941, SEC-1026, and COREPLG-510]