RTI TLS Support

Release Notes

Version 7.2.0

Trademarks

Copy and Use Restrictions

Third-Party Software

Notices

Deprecations and Removals

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

Chapter 1 Supported Platforms

See the column for *TLS Support* in the table of Supported Platforms for Compiler-Dependent Product, in the RTI Connext Core Libraries Release Notes.

Chapter 2 Compatibility

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

TLS Support 7.2.0 is API-compatible with OpenSSL versions 3.0.0 through 3.0.9, not with versions earlier than OpenSSL 3.0.0. Note that *TLS Support* 7.2.0 has only been tested by RTI using OpenSSL 3.0.9. If you need *TLS Support* 7.2.0 to run against older versions of OpenSSL, please contact support@rti.com.

For instructions on installing the latest version of OpenSSL, see the *TLS Support Installation Guide*.

TLS Support 7.2.0 uses TLS 1.3. When communicating with *TLS Support* 6.0.0 or below, *TLS Support* 7.2.0 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 this and previous releases, see the *Migration Guide* on the RTI Community Portal (https://community.rti.com/documentation).

Chapter 3 What's New in 7.2.0

3.1 Upgraded OpenSSL to version 3.0.9 and removed OpenSSL 1.1.1 support

TLS Support 7.2.0 supports the latest LTS version of OpenSSL (OpenSSL 3.0.9). In this release, *TLS Support* is only available as a set of **nddstls** libraries built against OpenSSL 3.0.9 (which is supported until September, 2026). The support of OpenSSL 1.1.1 has been removed, because it is end-of-life in September, 2023.

See also Chapter 2 Compatibility on page 2.

Chapter 4 What's Fixed in 7.2.0

4.1 *TLS Support* FATAL verbosity not enacted when set for TCP transport

A bug caused FATAL errors not to be logged in *TLS Support* for the TCP transport, even if you set a FATAL verbosity for the transport using the **security_logging_verbosity_bitmap** property. This problem has been fixed.

[RTI Issue ID COREPLG-627]

Chapter 5 Previous Releases

5.1 What's New in 7.1.0

5.1.1 Upgrade OpenSSL to versions 1.1.1t and 3.0.8

The following third-party software used by *TLS Support* has been upgraded:

Third-Party Tool	Old Version	New Version
OpenSSL	1.1.1n	1.1.1t 3.0.8

TLS Support now supports the latest LTS version of OpenSSL (OpenSSL 3.0). In this release, *TLS Support* is available as both a set of **nddstls** libraries built against OpenSSL 1.1.1t (supported until September 2023) and a set of **nddstls** libraries built against OpenSSL 3.0.8 (supported until September 2026).

See Chapter 2 Compatibility on page 2. See also the *Migration Guide* on the RTI Community Portal (https://community.rti.com/documentation) for migration issues related to this upgrade.

5.1.2 TLS Support now included with Connext Secure and Connext Anywhere

In release 7.1.0, *RTI TLS Support* is now included with the purchase of the *Connext Secure* and *Connext Anywhere* bundles. It is still installed separately. See the *RTI TLS Support Installation Guide*.

5.2 What's Fixed in 7.1.0

5.2.1 Using dh_param_files Leaked Memory

Using the property **tls.cipher.dh_param_files** leaked memory 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* 7.1.0 communicates with an application using OpenSSL 1.0.2, the leak will not occur.

[RTI Issue ID COREPLG-641]

5.2.2 Failure to Load a String-Based Private Key Leaked Memory

If you set the property **tls.identity.private_key** or **tls.identity.rsa_private_key**, and you either specified a wrong or missing value for the property **tls.identity.private_key_password** or specified a malformed private key, then memory would be leaked 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]

5.2.3 Fixes Related to Vulnerabilities

5.2.3.1 Potential eavesdropping when using OpenSSL 1.1.1 due to a vulnerability in OpenSSL 1.1.1

TLS Support had a third-party dependency on OpenSSL 1.1.1, which is known to be affected by a number of publicly disclosed vulnerabilities.

These vulnerabilities have been fixed by upgrading OpenSSL to the latest stable version, 1.1.1t. See 5.1 What's New in 7.1.0 on the previous page for more details.

5.2.3.1.1 User Impact without Security

The impact on *Connext* applications of using the previous version was as follows:

- Exploitable by sending trial messages to a DDS Entity.
- The application's confidential data could be decrypted by an attacker.
- CVSS Base Score: 5.9 MEDIUM
- CVSS v3.1 Vector: AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:N/A:N

5.2.3.1.2 User Impact with Security

Same impact as described in "User Impact without Security," above.

[RTI Issue ID COREPLG-689]

5.3 What's Fixed in 7.0.0

5.3.1 Memory Leak when Running out of Memory

If either of the internal functions **RTITLS_ConnectionEndpointFactoryTLSv4_createConnectEndpoint()** or **RTITLS_ConnectionEndpointFactoryTLSv4_createAcceptEndpoint()** ran out of memory, connection creation would fail with a memory leak.

Here is one example set of error messages, along with a valgrind result:

```
NDDS_Transport_TCPv4_Plugin_clientOpenControlConnection_connEA:!create connection endpoint
NDDS_Transport_TCPv4_Plugin_clientOpenControlConnection_connEA:error connecting to peer at
127.0.0.1:36025
NDDS_Transport_TCPv4_Plugin_clientOpenControlConnection_connEA:failed to (re)connect client
control connection
NDDS_Transport_TCPv4_create_sendresource_srEA:failed to open client control connection
==23757== 8,384 (6,280 direct, 2,104 indirect) bytes in 1 blocks are definitely lost in loss
record 128 of 134
==23757==    at 0x4C2FB0F: malloc (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so)
==23757==    by 0x13F366D: CRYPTO_malloc (mem.c:222)
==23757==    by 0x13F36A0: CRYPTO_zalloc (mem.c:230)
==23757==    by 0x1331070: SSL_new (ssl_lib.c:691)
==23757==    by 0xC0FDE9: RTITLS_ConnectionEndpointFactoryTLSv4_createConnectEndpoint
(TLSConnection.c:837)
==23757==    by 0x6266F8: NDDS_Transport_TCPv4_Plugin_clientOpenControlConnection_connEA
(Tcpv4.c:3321)
```

The leak would only happen if memory was already exhausted, so this problem did not lead to unbounded memory growth.

This problem has been fixed. Those two functions will now fail without a memory leak.

[RTI Issue ID COREPLG-589]

Chapter 6 Known Issues

Note: For an updated list of critical known issues, see the Critical Issues List on the RTI Customer Portal at https://support.rti.com.

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*, and *TLS Support*.

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