# RTI TLS Support

## Installation Guide

Version 7.3.0

## Trademarks

RTI, Real-Time Innovations, Connext, Connext Drive, 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 solely under and subject to RTI's standard terms and conditions available at https://www.rti.com/terms and in accordance with your License Acknowledgement Certificate (LAC) and Maintenance and Support Certificate (MSC), except to the extent otherwise accepted in writing by a corporate officer of RTI.

## Third-Party Software

RTI software may contain independent, third-party software or code that are subject to third-party license terms and conditions, including open source license terms and conditions. Copies of applicable third-party licenses and notices are located at community.rti.com/documentation. IT IS YOUR RESPONSIBILITY TO ENSURE THAT YOUR USE OF THIRD-PARTY SOFTWARE COMPLIES WITH THE CORRESPONDING THIRD-PARTY LICENSE TERMS AND CONDITIONS.

## 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 Introduction

*RTI® TLS Support* is an optional product 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 a *Connext* installation with the same version number; it can only be used on architectures that support TCP transport.

To see a simple 'Hello, World' example application that uses TCP and TLS, look in the **hello_world_tcp** example directory for C. (See Chapter 2 Paths Mentioned in Documentation on page 2 for the path to the examples.)

# Chapter 2 Paths Mentioned in Documentation

The documentation refers to:

- **<NDDSHOME>**

  This refers to the installation directory for *RTI*® *Connext*®. The default installation paths are:

  - macOS® systems:  
    **/Applications/rti_connext_dds-7.3.0**
  - Linux systems, non-*root* user:  
    **/home/<*your user name*>/rti_connext_dds-7.3.0**
  - Linux systems, *root* user:  
    **/opt/rti_connext_dds-7.3.0**
  - Windows® systems, user without Administrator privileges:  
    **<*your home directory*>\rti_connext_dds-7.3.0**
  - Windows systems, user with Administrator privileges:  
    **C:\Program Files\rti_connext_dds-7.3.0**

  You may also see **$NDDSHOME** or **%NDDSHOME%**, which refers to an environment variable set to the installation path.

  Wherever you see **<NDDSHOME>** used in a path, replace it with your installation path.

**Note for Windows Users:** When using a command prompt to enter a command that includes the path **C:\Program Files** (or any directory name that has a space), enclose the path in quotation marks. For example:

```
“C:\Program Files\rti_connext_dds-7.3.0\bin\rtiddsgen”
```

Or if you have defined the **NDDSHOME** environment variable:

```
“%NDDSHOME%\bin\rtiddsgen”
```

- ***<path to examples>***

By default, examples are copied into your home directory the first time you run *RTI Launcher* or any script in **<NDDSHOME>/bin**. This document refers to the location of the copied examples as ***<path to examples>***.

Wherever you see ***<path to examples>***, replace it with the appropriate path.

Default path to the examples:

- macOS systems: **/Users/<*your user name*>/rti_workspace/7.3.0/examples**
- Linux systems: **/home/<*your user name*>/rti_workspace/7.3.0/examples**
- Windows systems: **<*your Windows documents folder*>\rti_workspace\7.3.0\examples**

  Where 'your Windows documents folder' depends on your version of Windows. For example, on Windows 10, the folder is **C:\Users\<*your user name*>\Documents**.

Note: You can specify a different location for **rti_workspace**. You can also specify that you do not want the examples copied to the workspace. For details, see *Controlling Location for RTI Workspace and Copying of Examples* in the *RTI Connext Installation Guide*.

# Chapter 3 Download Instructions

*RTI TLS Support*, which is included with some RTI purchases, must be downloaded and installed separately. Download *TLS Support* from the RTI Customer Portal, accessible from https://support.rti.com/.

You will need your username and password to log into the portal; these are included in the letter confirming your purchase or evaluation copy. If you do not have this letter, please contact **license@rti.com**.

*TLS Support* also requires OpenSSL. OpenSSL is available from RTI's Customer Portal, or you may obtain it from another source.

Once you have logged into the portal, select the **Downloads** link, then select the appropriate version of *TLS Support* and OpenSSL for your platform.

For *TLS Support*, download both:

- **rti_tls_support-7.3.0-host-openssl-3.0-<host platform>.rtipkg**
- **rti_tls_support-7.3.0-target-openssl-3.0-<target architecture>.rtipkg**

  The host package includes the compiler-independent *TLS Support* dependencies (documentation, headers, and the libraries used by RTI tools and services) for the host platform.

  The target package contains the *TLS Support* libraries you will link against for your target architecture.

For OpenSSL, download both:

- **openssl-3.0.12-7.3.0-host-<host platform>.rtipkg**
- **openssl-3.0.12-7.3.0-target-<target architecture>.rtipkg**

  The host package includes the OpenSSL distribution files for RTI tools and services.

The target package includes OpenSSL distribution files to link against your application.

**Note:** The OpenSSL target packages for cross-compiled architectures such as QNX do not include the OpenSSL executable.

Where:

- <host platform> depends on your host (**x64Linux** for Linux systems, **darwin** for macOS systems, **x64Win64** for Windows systems).
- <target architecture> names are described in the *RTI Connext Core Libraries Platform Notes*.

If you need help with the download process, contact **support@rti.com**.

# Chapter 4 Installing on Windows Systems

You do not need administrator privileges. All directory locations are meant as examples only; adjust them to suit your site. <NDDSHOME> is described in Chapter 2 Paths Mentioned in Documentation on page 2.

1. Install the *TLS Support* host and target **.rtipkg** files on top of *Connext*.

   There are two ways to install: from *RTI Launcher* or from the command line.

   **To install from *RTI Launcher*:**

   a. Start *RTI Launcher*:

      ```
      cd <NDDSHOME>
      bin\rtilauncher
      ```

   b. From the **Configuration** tab, click on **Install RTI Packages**.

   c. Use the + sign to add the **.rtipkg** files that you want to install.

   d. Click **Install**.

   **To install from the command line:**

   ```
   cd <NDDSHOME>
   bin\rtipkginstall <path to .rtipkg file>
   ```

   The installer will put the TLS libraries in **<NDDSHOME>\lib\<architecture>**.

2. If not already included, add **<NDDSHOME>\lib\<architecture>** to your Path environment variable. For example (enter this on one line):

   ```
   set PATH=<NDDSHOME>\lib\<architecture>;%PATH%
   ```

3. Optionally, install the OpenSSL host package. Use the same process that you used for the **.rtipkg** files above.

You only need this package if your target architecture is a Linux, Windows, or macOS system, *and* you plan to run RTI Infrastructure Services or Tools.

4. Install the OpenSSL target package. (Use the same process that you used for the **.rtipkg** files above.)

   The installer will put the OpenSSL target libraries in **<NDDSHOME>\third_party\openssl-3.0.12\<architecture>**.

5. Add the OpenSSL **bin** directory to your Path environment variable. For example, assuming you want to use the "release" version of the OpenSSL 3.0.12 libraries:

   ```
   set PATH=
   <NDDSHOME>\third_party\openssl-3.0.12\<architecture>\release\bin;%PATH%
   ```

6. To verify your OpenSSL installation, enter:

   ```
   openssl version
   ```

   You should see a response similar to:

   ```
   OpenSSL 3.0.12
   ```

   If you see a version that you didn't expect, your PATH may be pointing with a higher precedence to a different version of OpenSSL. You may need to place the version you just installed first or earlier in your PATH.

   You may run into this OpenSSL warning:

   ```
   WARNING: can't open config file: [default openssl built-in path]\openssl.cnf
   ```

   To resolve this issue, set the environment variable OPENSSL_CONF to the path to the **openssl.cnf** file you are using. For example (enter this all on one line):

   ```
   set OPENSSL_CONF=
   <NDDSHOME>\third_party\openssl-3.0.12\<architecture>\release\ssl\openssl.cnf
   ```

# Chapter 5 Installing on Linux and Other Systems

This chapter applies to all supported operating systems except Windows, which is covered in Chapter 4 Installing on Windows Systems on page 6.

All directory locations are meant as examples only; adjust them to suit your site. <NDDSHOME> is described in Chapter 2 Paths Mentioned in Documentation on page 2.

1. Install the *TLS Support* host and target **.rtipkg** files on top of *Connext*.

   There are two ways to install: from *RTI Launcher* or from the command line.

   **To install from *RTI Launcher*:**

   a. Start *RTI Launcher*:

      ```
      cd <NDDSHOME>
      ./bin/rtilauncher
      ```

   b. From the **Configuration** tab, click on **Install RTI Packages**.

   c. Use the + sign to add the **.rtipkg** files that you want to install.

   d. Click **Install**.

   **To install from the command line:**

   ```
   cd <NDDSHOME>
   ./bin/rtipkginstall <path to .rtipkg file>
   ```

   The installer will put the TLS libraries in **<NDDSHOME>/lib/<architecture>**.

2. If not already included, add **<NDDSHOME>/lib/<architecture>** to your library search path (pointed to by the LD_LIBRARY_PATH environment variable on Linux systems, DYLD_LIBRARY_PATH on macOS systems).

   For example:

```
export LD_LIBRARY_PATH=<NDDSHOME>/lib/<architecture>:${LD_LIBRARY_PATH}
```

3. Optionally, install the OpenSSL host package. Use the same process that you used for the **.rtipkg** files above.

   You only need this package if your target architecture is a Linux, Windows, or macOS system, *and* you plan to run RTI Infrastructure Services or Tools.

4. Install the OpenSSL target package. (Use the same process that you used for the **.rtipkg** files above.)

   The installer will put the OpenSSL target libraries in **<NDDSHOME>/third_party/openssl-3.0.12/<architecture>**.

   **Note:** The OpenSSL target bundles for cross-compiled architectures such as QNX do not include the OpenSSL executable.

5. Add the OpenSSL **lib** directory to your library search path. For example, assuming you want to use the "release" version of the OpenSSL 3.0.12 libraries (enter this on one line):

   ```
   export LD_LIBRARY_PATH=
   <NDDSHOME>/third_party/openssl-3.0.12/<architecture>/release/lib:${LD_LIBRARY_PATH}
   ```

6. **Only on cross-compiled architectures:** The OpenSSL target package for cross-compiled architectures does not include the openssl executable in the **bin** directory. If you require the OpenSSL executable (e.g., to generate identity certificates), you need to download and install the OpenSSL target package for an architecture that is not cross-compiled. Once the package is installed, add it to your PATH as described in the next step.

7. Include the resulting OpenSSL **bin** directory in your PATH. For example, assuming you want to use the "release" version of the OpenSSL 3.0.12 libraries (enter the command all on one line):

   ```
   export PATH=<NDDSHOME>/third_party/openssl-3.0.12/<architecture>/release/bin:${PATH}
   ```

8. To verify your OpenSSL installation, enter:

   ```
   openssl version
   ```

   You should see a response similar to:

   ```
   OpenSSL 3.0.12
   ```

   If you see a version that you didn't expect, your PATH may be pointing with a higher precedence to a different version of OpenSSL. You may need to place the version you just installed first or earlier in your PATH.

You may run into this OpenSSL warning:

```
WARNING: can't open config file: [default openssl built-in path]/openssl.cnf
```

To resolve this issue, set the environment variable OPENSSL_CONF to the path to the **openssl.cnf** file you are using. For example, if you are using OpenSSL 3.0.12 (enter this all on one line):

```
export OPENSSL_CONF=
<NDDSHOME>/third_party/openssl-3.0.12/<architecture>/release/ssl/openssl.cnf
```

# Chapter 6 Special Backup of RTI Libraries

The following information applies if you are installing a patch release (a release with four digits, such as 7.3.0.1) into an existing installation directory. It does not apply if you are installing a patch release into a new directory. See the instructions that come with your patch release to know whether it's okay to install into an existing directory or you should install into a new one. You should almost never install a three-digit release into an existing installation directory.

## 6.1 Overwriting Libraries

When installing a new RTI package that overwrites a given library in the **<NDDSHOME>/lib/<architecture>** directory, the installer will copy the library from the **<NDDSHOME>/lib/<architecture>** directory to the **<NDDSHOME>/lib/<architecture>/<current_installed_version>** directory. This copy will serve as a backup.

For example, if you install patch version 7.3.0.1 to the RTI core libraries for x64Win64VS2017, your overwritten core 7.3.0 libraries will be copied into the following directory before the 7.3.0.1 libraries are installed: **<NDDSHOME>/lib/x64Win64VS2017/7.3.0**.

If you install another patch later, before the 7.3.0.1 libraries are overwritten, then the 7.3.0.1 libraries will be copied into **<NDDSHOME>/lib/x64Win64VS2017/7.3.0.1**.

## 6.2 Installing Add-on Products when Using RTI Tools

RTI Tools commonly rely on the backup directory, so it is important to make sure its contents are correct, especially when you are using add-on products, such as *RTI Security Plugins*. To ensure the backup contents are correct, do the following:

> Before installing a new patch (e.g., 7.3.0.1) that contains an add-on product, such as *Security Plugins*, always install the base version (e.g., **rti_security_plugins-7.3.0-host-<arch>.rtipkg**) of the add-on product, if you haven't already, before installing the patch release

version (e.g., **rti_security_plugins-7.3.0.1-host-<arch>.rtipkg**).

Installing the base version first ensures that when RTI Tools (such as *RTI Admin Console*, *RTI Monitor*, and *RTI System Designer*) are not part of the patch release (they usually are not), they keep using the base libraries they are built with instead of the patched libraries. You will receive an error otherwise.

(The RTI tool will look for the base libraries in the backup directory, such as **<NDDSHOME>/resource/app/lib/x64Win64VS2017/7.3.0**. But that backup directory either will not exist or will not contain all of the necessary base libraries if you didn't install the base version packages first. The tool will then look for the library in the patched directory. But if the patched library uses a function that is not implemented in the core library in the backup directory, an error occurs.)