RTI Security Plugins

Installation Guide

Version 6.1.1

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

Trademarks

Copy and Use Restrictions

Securing a distributed, embedded system is an exercise in user risk management. RTI expressly disclaims all security guarantees and/or warranties based on the names of its products, including Connext DDS Secure, RTI Security Plugins, and RTI Security Plugins SDK. Visit https://www.rti.com/terms/ for complete product terms and an exclusive list of product warranties.

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 Download Instructions

> The *Connext DDS* LM bundle includes the *Security Plugins* and OpenSSL®. If you have the LM version of *Connext DDS* (with "lm" in its package name), you can skip to the next chapter.

Log into the RTI Customer Portal, https://support.rti.com/. You will need your username and password, which are included in the letter confirming your purchase. If you do not have this letter, please contact license@rti.com If you need help with the download process, contact support@rti.com.

Once you have logged into the portal, select the **Downloads** link. Which files to download depends on your host, target, and which crypto libraries you will be using, OpenSSL or wolfSSL®. See details below.

1.1 Downloading the Security Plugins for OpenSSL

From the portal's **Downloads** page, select the appropriate version of the *Security Plugins* and OpenSSL for your platform. (You may also obtain OpenSSL from another source.)

1. For the *Security Plugins* for OpenSSL, download both:
   - **rti_security_plugins-6.1.1-host-<*host platform*>.rtipkg**

     This includes the compiler-independent *Security Plugins* dependencies (documentation, headers, and the libraries used by RTI tools and services) for the host platform.
   - **rti_security_plugins-6.1.1-target-<*target architecture*>.rtipkg**

     This contains the *Security Plugins* libraries you will link against for your target architecture.

2. For OpenSSL, download both:
   - **openssl-1.1.1n-6.1.1-host-<*host platform*>.rtipkg**

     This includes the OpenSSL distribution files for RTI tools and services
   - **openssl-1.1.1n-6.1.1-target-<*target architecture*>.rtipkg**

     This includes OpenSSL distribution files to link against your application.

<*host platform*> names depend on your host (**x64Linux** for Linux systems, **darwin** for macOS systems, **x64Win64** for Windows systems).

<*target architecture*> names are described in the *RTI Connext DDS Core Libraries Platform Notes*.

1.2 Downloading the Security Plugins for wolfSSL

From the portal's **Downloads** page, select the appropriate version of the *Security Plugins* for wolfSSL. Only select target architectures can be used with wolfSSL; these are noted in the Compatibility section of the *Security Plugins Release Notes*.

1. For the *Security Plugins* for wolfSSL, download both:
   - **rti_security_plugins-6.1.1-host-wolfssl-4.7-<*host platform*>.rtipkg**

     This includes the compiler-independent *Security Plugins* dependencies (documentation, headers, and the libraries used by RTI tools and services) for the host platform.
   - **rti_security_plugins-6.1.1-target-wolfssl-4.7-<*target architecture*>.rtipkg**

     This contains the *Security Plugins* libraries you will link against for your target architecture.

2. For wolfSSL:

   RTI does not distribute wolfSSL as a package bundle. You should get a commercial version of wolfSSL 4.7 and follow the instructions in 3.3 Installing a Crypto Library on page 7.

<*host platform*> names depend on your host (**x64Linux** for Linux systems, **darwin** for macOS systems, **x64Win64** for Windows systems).

<*target architecture*> names are described in the *RTI Connext DDS Core Libraries Platform Notes*.

Chapter 2 Paths Mentioned in Documentation

The documentation refers to:

- **<NDDSHOME>**

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

  - macOS® systems:
    /Applications/rti_connext_dds-6.1.1
  - Linux systems, non-*root* user:
    /home/<*your user name*>/rti_connext_dds-6.1.1
  - Linux systems, *root* user:
    /opt/rti_connext_dds-6.1.1
  - Windows® systems, user without Administrator privileges:
    <*your home directory*>\rti_connext_dds-6.1.1
  - Windows systems, user with Administrator privileges:
    C:\Program Files\rti_connext_dds-6.1.1

  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-6.1.1\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/6.1.1/examples**
  - Linux systems: **/home/<*your user name*>/rti_workspace/6.1.1/examples**
  - Windows systems: ***<your Windows documents folder>*\rti_workspace\6.1.1\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 DDS Installation Guide*.

Chapter 3 Installation Instructions

You do not need administrator privileges. All directory locations are meant as examples only; adjust them to suit your site.

Follow the steps in either:

3.1 Installing a License-Managed (LM) Version

The license-managed (LM) version of *Connext DDS* comes with OpenSSL and will be automatically installed. wolfSSL is not supported with the LM version.

1. Install the *Connext DDS* license-managed ("lm") bundle as described in the *RTI Connext DDS Installation Guide*.

   The "lm" bundle includes *Security Plugins* and OpenSSL. The installer provides a pre-built version of OpenSSL 1.1.1n. If you want to build your own version of OpenSSL 1.1.1n, you can find the source code here: https://www.openssl.org/source/.

   After installation, the *Security Plugins* header files and libraries will be in ***<install dir>*/include/ndds/security** and ***<install dir>*/lib/*<target architecture>***, respectively. OpenSSL will be under ***<install dir>*/third_party**.
2. Add OpenSSL's **/bin** directory to your PATH.

   For example, assuming you want to use the *release* version of the OpenSSL libraries, enter the following command (all on one line). Adjust the path to match your installation directory and use your own architecture string.

On Linux and macOS systems:

```
> export PATH=
  <install dir>/third_party/openssl-1.1.1n/<architecture>/release/bin:${PATH}
```

If linking dynamically, also add OpenSSL's **/lib** directory in your **LD_LIBRARY_PATH**. For example:

```
> export LD_LIBRARY_PATH=
  <installdir>/third_party/openssl-1.1.1n/<architecture>/release/lib:$LD_ LIBRARY_PATH
```

On Windows systems:

```
> set PATH=
  <install dir>\third_party\openssl-1.1.1n\<architecture>\release\bin;%PATH%
```

3. To verify your installation, enter:

```
> openssl version
```

You should see a response similar to:

```
OpenSSL 1.1.1n
```

4. Your *Security Plugins* distribution requires a license file. See Chapter 4 License Management on page 12.

3.2 Installing a Regular (non-LM) Version

3.2.1 Security Plugins for OpenSSL

1. Install the *Connext DDS* host and target bundles as described in the *RTI Connext DDS Installation Guide*.
2. Install the *Security Plugins* host and target packages:
   - **rti_security_plugins-6.1.1-host-<*host platform*>.rtipkg**
   - **rti_security_plugins-6.1.1-target-<*target architecture*>.rtipkg**

   <*host platform*> depends on your host (such as **x64Linux** on Linux systems, **darwin** on macOS systems, **x64Win64** on Windows systems).

   <*target architecture*> is one of the supported platforms, see the *RTI Security Plugins Release Notes*.

   After installation, the *Security Plugins* header files and libraries will be in ***<install dir>*/include/ndds/security** and ***<install dir>*/lib/*<target architecture>***, respectively.
3. Install OpenSSL as described in 3.3.1 Installing OpenSSL on the next page.

3.2.2 Security Plugins for wolfSSL

The *Security Plugins* for wolfSSL are only supported on select target platforms. The Compatibility section of the *RTI Security Plugins Release Notes* lists which platforms are compatible with wolfSSL.

1. Install the *Connext DDS* host and target bundles as described in the *RTI Connext DDS Installation Guide*.
2. Install the *Security Plugins* host and target packages that are compatible with wolfSSL:
   - **rti_security_plugins-6.1.1-wolfssl-4.7-host-<*host platform*>.rtipkg**

     Note: You only need this host package if your target architecture is for a Linux, Windows, or macOS system, *and* you plan to run RTI infrastructure services or tools.
   - **rti_security_plugins-6.1.1-wolfssl-**4.7**-target-<*target architecture*>.rtipkg**

   <*host platform*> depends on your host (**x64Linux** on Linux systems, **darwin** on macOS systems, **x64Win64** on Windows systems).

   <*target architecture*> is one of the supported platforms that supports wolfSSL, see the *RTI Security Plugins Release Notes*.

   After installation, the *Security Plugins* header files and libraries will be in **<*install dir*>/include/ndds/security** and **<*install dir*>/lib/<*target architecture*>**, respectively.

1. Install wolfSSL as described in 3.3 Installing a Crypto Library below.

3.3 Installing a Crypto Library

3.3.1 Installing OpenSSL

> If you have the license-managed (LM) version of *Connext DDS* (with "lm" in the package file name): OpenSSL is installed automatically with the LM bundle. The following instructions are only for regular installations.

RTI provides:

- An OpenSSL host package, which enables OpenSSL for RTI's applications such as *RTI Admin Console*, *RTI Routing Service*, *rtiddsspy*, etc.
- An OpenSSL target package, which provides OpenSSL libraries that can be used to secure your applications.

3.3.1.1 Linux and macOS Systems

1. Make sure you've installed host and target *Security Plugins* packages as described in 3.2.1 Security Plugins for OpenSSL on the previous page.
2. Install an OpenSSL host package from RTI: **openssl-1.1.1n-host-<host platform>.rtipkg**. (The <host platform> for **x64Linux** for Linux systems, or **darwin** for macOS systems.) Use the same process that you used for the **.rtipkg** files in the previous step.

3. Install an OpenSSL target package from RTI: **openssl-1.1.1n-6.1.1-target-<target architecture>.rtipkg**. (Use the same process that you used for the **.rtipkg** files in the previous step.)
4. Include the resulting OpenSSL **bin** directory in your **PATH**. For example, assuming you want to use the "release" version of the OpenSSL libraries (enter the command all on one line):

```
export PATH=
<NDDSHOME>/third_party/openssl-1.1.1n/<architecture>/release/bin:${PATH}
```

5. If you will be using the dynamic libraries, include the resulting OpenSSL **lib** directory in your LD_LIBRARY_PATH (on Linux systems) or DYLD_LIBRARY_PATH (on macOS systems). For example, assuming you want to use the *release* version of the OpenSSL libraries (enter the command all on one line):

```
export LD_LIBRARY_PATH=
<NDDSHOME>/third_party/openssl-1.1.1n/<architecture>/release/lib:$LD_LIBRARY_PATH
```

6. To verify your installation, enter:

```
openssl version
```

   You should see a response similar to:

```
OpenSSL 1.1.1n
```

   If you get a version other than OpenSSL 1.1.1n, your PATH may be pointing with a higher precedence to a different version of OpenSSL. You may need to place version 1.1.1n first or earlier in your PATH.

   Note: When running the **openssl version** command, 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):

```
export OPENSSL_CONF=
<NDDSHOME>/third_party/openssl-1.1.1n/<architecture>/release/ssl/openssl.cnf
```

3.3.1.2 Windows Systems

1. Make sure you've installed host and target *Security Plugins* packages as described in 3.2.1 Security Plugins for OpenSSL on page 6.
2. Install an OpenSSL host package from RTI: **openssl-1.1.1n-host-x64Win64.rtipkg.** Use the same process that you used for the **.rtipkg** files in the previous step.
3. Install an OpenSSL target package from RTI: **openssl-1.1.1n-6.1.1-target-<target architecture>.rtipkg**. (Use the same process that you used for the **.rtipkg** file in the previous step.)

4. Add the resulting OpenSSL **bin** directory to your **Path** environment variable. For example (enter the command all on one line):

```
set PATH=
<NDDSHOME>\third_party\openssl-1.1.1n\<architecture>\release\bin;%PATH%
```

5. If you will be using the dynamic libraries, add the resulting OpenSSL **lib** directory to your **Path**. For example, assuming you want to use the *release* version of the OpenSSL libraries (enter the command all on one line):

```
set PATH=
<NDDSHOME>\third_party\openssl-1.1.1n\<architecture>\release\lib;%PATH%
```

6. To verify your installation, open a command prompt and enter:

```
openssl version
```

   You should see a response similar to:

```
OpenSSL 1.1.1n
```

   If you get a version other than OpenSSL 1.1.1n, your PATH may be pointing with a higher precedence to a different version of OpenSSL. You may need to place version 1.1.1n first or earlier in your path.

   Note: When running the above command, 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-1.1.1n\<architecture>\release\ssl\openssl.cnf
```

3.3.2 Building wolfSSL

wolfSSL is only for use with specific architectures noted in the *RTI Security Plugins Release Notes*. RTI does not distribute wolfSSL. You should get a commercial version of wolfSSL 4.7.

In a location of your choice, build wolfSSL for your target architecture. Read the the chapter on "Building" in the wolfSSL User Manual.

It is important that you build wolfSSL with the following flags:

- --enable-smime
- --enable-opensslall
- --enable-opensslextra
- --enable-crl

- --enable-certgen
- --enable-des3
- --enable-reproducible-build
- --enable-aesni
- -DWOLFSSL_PSS_SALT_LEN_DISCOVER
- --enable-harden
- --enable-static

You will need the resulting installation directory when installing wolfSSL in the next section.

We refer to the wolfSSL installation directory as the folder created after building wolfSSL. This folder should contain **bin/**, **include/**, **lib/**, and **share/** directories. You can configure it when building wolfSSL by adding the **--prefix** and **--exec-prefix** flags during the **make install** step.

3.3.3 Installing wolfSSL

After you've built wolfSSL for your target architecture:

1. Make sure you've installed the host and target *Security Plugins* packages as described in 3.2 Installing a Regular (non-LM) Version on page 6.
2. In your **<NDDSHOME>/third_party** directory, create **wolfssl-4.7.0/<target architecture>/release/**. Copy your wolfSSL installation directory under the **release/** folder.

   (This assumes that you want to use the *release* version of the wolfSSL libraries, if you want to use the *debug* version of the libraries, use **<NDDSHOME>/third_party/wolfssl-4.7.0/<target architecture>/debug/** instead.)

   You will end up with: **<NDDSHOME>/third_party/wolfssl-4.7.0/<target architecture>/[release|debug]/**.
3. (This step isn't necessary for a QNX target, because the tools and services are supported natively on QNX systems.)

   If your *target* architecture is on a Linux, macOS, or Windows system and you want to use RTI Tools and Infrastructure Services: you also need to build the wolfSSL library compiled for your *host* architecture. To do so, repeat the steps in 3.3 Installing a Crypto Library on page 7 and create a new wolfSSL installation directory with the library compiled for your *host* architecture.

   Host architecture names are: **aix**, **darwin**, **x64Linux**, and **x64Win64**.

   Once you have wolfSSL compiled for your host architecture, copy the dynamic library files (*.so) to the **<NDDSHOME>/resource/app/lib/<host architecture>/** directory. The dynamic library files

are in the **lib/** directory of your wolfSSL installation directory.

You must copy both the release and debug versions, including symbolic links.

4. Include the wolfSSL **bin/** directory in your PATH.

   For example, assuming you want to use the "release" version of the wolfSSL libraries (enter the command all on one line):

   ```
   export PATH=
   <NDDSHOME>/third_party/wolfssl-4.7.0/<architecture>/release/bin:${PATH}
   ```

   If you will be using the dynamic libraries, include the wolfSSL **lib/** directory in your library search path (LD_LIBRARY_PATH on Linux systems, DYLD_LIBRARY_PATH on macOS systems, or Path on Windows systems). For example, assuming you want to use the release version of the wolfSSL libraries (enter the command all on one line):

   ```
   export LD_LIBRARY_PATH=
   <NDDSHOME>/third_party/wolfssl-4.7.0/<architecture>/release/lib:$LD_LIBRARY_PATH
   ```

5. To verify your installation, enter:

   ```
   wolfssl-config  --version
   ```

   You should see a response similar to:

   ```
   4.7.0
   ```

   If you get a version other than wolfSSL 4.7.0, your PATH may be pointing with a higher precedence to a different version of wolfSSL. You may need to place version 4.7.0 first or earlier in your PATH.

Chapter 4 License Management

There's a distinction between a license ***file*** and a ***license***. When you buy an RTI product, like the Professional package, you are ***licensed*** to use it.

Tools like *RTI Admin Console*, which are included in the Professional package, additionally require a license ***file*** in order to run. A license ***file*** is never required to deploy your system in production. LM (license-managed) bundles are not for production and require a license ***file***.

If your *Connext DDS* distribution requires a license file, you will receive one from RTI via email.

This section describes how to manage a license ***file***. If you have more than one license file from RTI, you can concatenate them into one file. A single license file can be used to run on any architecture and is not node-locked. You are not required to run a license server.

4.1 Installing the License File

Save the license file in any location of your choice; the locations checked by the plugin are listed below. You can also specify the location of your license file in *RTI Launcher*'s **Configuration** tab. Then *Launcher* can copy the license file to the installation directory or to the user workspace.

Each time your application starts, it will look for the license file in the following locations until it finds a valid license. (The properties are in the PropertyQosPolicy of the *DomainParticipant*.)

1. A property called **com.rti.serv.secure.license_string**. The value for this property can be set to the content of a license file. (This may be necessary if a file system is not supported on your platform.)
2. A property called **dds.license.license_string**. (Only if you have a license-managed, or "lm," version of *Connext DDS Professional*.)

   The above two **license_string** properties can be set to the content of a license file. (This may be necessary if a file system is not supported on your platform.) You can set the property either in source code or in an XML file.

If the content of the license file is in XML, special characters for XML need to be escaped in the license string. Special characters include: quotation marks (") (replace with "), apostrophes (') (replace with '), greater-than (>) (replace with >), less-than (<) (replace with <), and ampersands (&) (replace with &).

Example XML file:

```
<domain_participant_qos>
    <property>
        <value>
            <element>
                <name>dds.license.license_string</name>
                <value>contents of license file</value>
            </element>
        </value>
    </property>
</domain_participant_qos>
```

3. A property called **com.rti.serv.secure.license_file**.
4. A property called **dds.license.license_file**. (Only if you have a license-managed, or "lm," version of *Connext DDS Professional*.)

   The above two **license_file** properties can be set to the location (full path and filename) of a license file. (This may be necessary if a default license location is not feasible and environment variables are not supported.) You can set the property either in source code or in an XML file.

   Example XML to set **dds.license.license_file**:

```
<domain_participant_qos>
    <property>
        <value>
            <element>
                <name>dds.license.license_file</name>
                <value>path to license file</value>
            </element>
        </value>
    </property>
</domain_participant_qos>
```

5. In the location specified in the environment variable RTI_LICENSE_FILE, which you may set to point to the full path of the license file, including the filename.

   Note: When you run any of the scripts in the **<NDDSHOME>/bin** directory, this automatically sets the RTI_LICENSE_FILE environment variable (if it isn't already set) prior to calling the executable. It looks for the license file in two places: your **rti_workspace** directory and the installation directory (NDDSHOME). (See Chapter 2 Paths Mentioned in Documentation on page 3.)
6. If you are running any of the tools/services as executables via **NDDSHOME/bin/<*executable script*>** or through *Launcher*:

   a. In your **rti_workspace/<*version*>** directory, in a file called **rti_license.dat**.
   b. In your **rti_workspace** directory, in a file called **rti_license.dat**.
   c. In **<NDDSHOME>** (the *Connext DDS* installation directory), in a file called **rti_license.dat**.

7. If you are running your own application linked with *Connext DDS* libraries:
   a. In your current working directory, in a file called **rti_license.dat**.
   b. In **<NDDSHOME>** (the *Connext DDS* installation directory), in a file called **rti_license.dat**.

As *Connext DDS* attempts to locate and read your license file, you may (depending on the terms of the license) see a message with details about your license.

If the license file cannot be found or the license has expired, your application may be unable to initialize, depending on the terms of the license. If that is the case, your application's call to **DomainParticipantFactory.create_participant()** will return null, preventing communication.

If you have any problems with your license file, please email support@rti.com.

4.2 Adding or Removing License Management

If your license file changes—for example, you receive a new license for a longer term than your original license—you do not need to reinstall.

However, if you switch from a license-managed distribution of *Connext DDS* to one of the same version that does not require license management, or vice versa, RTI recommends that you first uninstall your original distribution before installing your new distribution. Doing so will prevent you from inadvertently using a mixture of libraries from multiple installations.

Chapter 5 Next Steps

See the *RTI Security Plugins Getting Started Guide* and *User's Manual* for further information on setting up and using the *Security Plugins*. The *Getting Started Guide* introduces important concepts and includes hands-on exercises. In the *User's Manual*, make sure to read these chapters in particular:

- Libraries Required for Using RTI Security Plugins
- Restrictions when Using RTI Security Plugins

For descriptions and examples of the security configuration in this release, please consult the **hello_security** examples under the **rti_workspace/<*version*>/examples/connext_dds/[c, c++, java, cs]** directory.

The *Security Plugins* documentation and examples may be updated online between releases. Please see the RTI Community website (https://community.rti.com) for the most up-to-date documentation.