RTI Security Plugins

Getting Started Guide

Version 6.0.1

Printed in U.S.A. First printing.
November 2019.

Trademarks

Copy and Use Restrictions

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® Security Plugins is a robust set of security capabilities, including authentication, encryption, access control and logging. Secure multicast support enables efficient and scalable distribution of data to many subscribers. Performance is also optimized by fine-grain control over the level of security applied to each data flow, such as whether encryption or just data integrity is required.

This release of *Security Plugins* includes partial support for the DDS Security specification from the Object Management Group (OMG)[1]. This support allows *DomainParticipants* to authenticate and authorize each other before initializing communication, and then encode and decode the communication traffic to achieve confidentiality, message authentication, and data integrity.

Specifically, these features are supported:

- Authentication can be done as part of the *RTI Connext® DDS* discovery process to ensure that DomainParticipants validate each other's identity.
- Access Control permissions checking can be done as part of the *Connext DDS* discovery process to ensure that *DomainParticipants*, *DataWriters*, and *DataReaders* have the appropriate permissions to exist and match with each other. Domain governance can now be done during entity creation to ensure the right security attributes are applied to the right *DomainParticipants*, *DataWriters*, and *DataReaders*.
- Cryptographic operations can be done as part of *Connext DDS* steady-state communication to ensure confidentiality, message authentication, and data integrity.
- Logging operations can be done using the Logging Plugin. There are options to print the log messages using NDDS_Config_Logger or an output file, distribute the log messages over a DDS topic, and control the verbosity level of the log messages.
- Data tagging can be done using the DataTagQosPolicy, and data tags can now be allowed or denied using the Permissions Document.

---

[1] http://www.omg.org/spec/DDS-SECURITY/1.1/

The above features are supported in the RTI core middleware in the C, C++, Java, and .NET programming languages.

The following DDS Security features are *not* supported:

- Revocation of identities and permissions
- Instance-level permissions checking

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.

To use *Security Plugins*, you will need to create private keys, identity certificates, governance, and permission files, as well as signed versions for use in secure authenticated, authorized, and/or encrypted communications.

If you are new to the world of internet security, see this link:

- https://en.wikipedia.org/wiki/Public-key_cryptography

Fundamentally, if you want to deploy a secure system, your organization will need to have an in-house security expert. Just using *Security Plugins* is not sufficient. It is a tool that can build secure systems, but you do have to use it (configure it) to meet your requirements. If used incorrectly, systems deployed with *Security Plugins* may not meet the security requirements of a project.

The *Security Plugins* bundle includes a set of builtin plugins that implement those defined by the DDS Security specification. It is also possible to implement new custom plugins by using the *Security Plugins SDK* bundle (for more information, please contact **support@rti.com**).

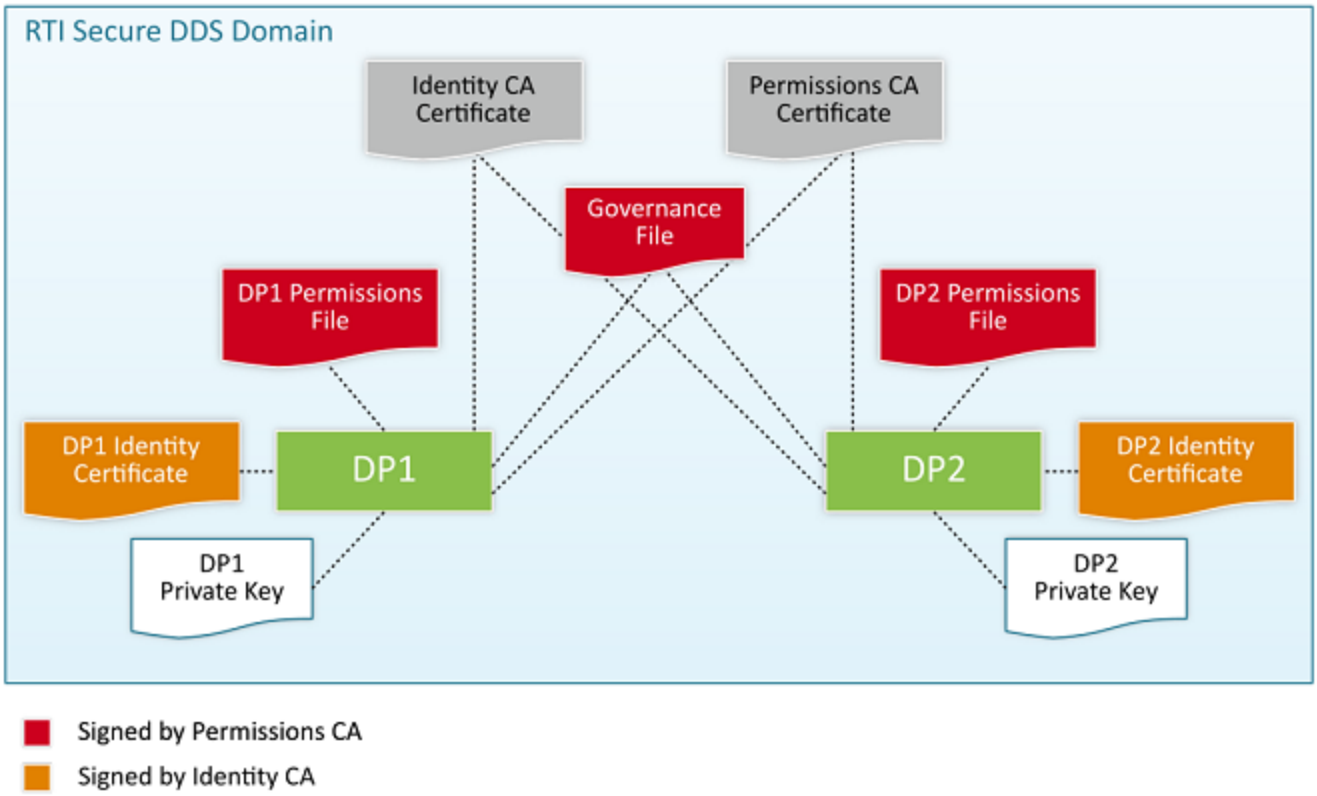
You should know that *Security Plugins* use the same technology as most of the world's eCommerce, so if you have ever purchased something on the internet, the same technology protecting your purchase is used by *Security Plugins* to protect data exchange.

As an end user, you need to have the following files, which an application using *Security Plugins* needs to communicate in a secure DDS domain:

- **Keys.** Each participant has a Private Key and Identity Certificate pair that is used in the authentication process.
- **Shared CA** has signed participant public keys. Participants must also have a copy of the CA certificate (also known as Identity Certificate Authority Certificate).
- **Permissions File** specifies what domains/partitions the *DomainParticipant* can join, what topics it can read/write, and what tags are associate with the readers/writers.
- **Domain Governance** specifies which domains should be secured and how.

Permissions CA has a signed participant permission file, as well as the domain governance document. Participants must have a copy of the permissions CA certificate (also known as Permissions Authority Certificate).

Figure 1.1: Artifacts required for RTI Security Plugins



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-6.0.1
  - UNIX-based systems, non-*root* user:
    /home/<*your user name*>/rti_connext_dds-6.0.1
  - UNIX-based systems, *root* user:
    /opt/rti_connext_dds-6.0.1
  - Windows® systems, user without Administrator privileges:
    <*your home directory*>\rti_connext_dds-6.0.1
  - Windows systems, user with Administrator privileges:
    C:\Program Files\rti_connext_dds-6.0.1

  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-6.0.1\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/6.0.1/examples**
  - UNIX-based systems: **/home/<*your user name*>/rti_workspace/6.0.1/examples**
  - Windows systems: ***<your Windows documents folder>*\rti_workspace\6.0.1\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 Download Instructions

Download *Security Plugins* from the RTI Support Portal, accessible from https://support.rti.com/.

Security Plugins also requires OpenSSL®, which is available from RTI's Support Portal, or you may obtain it from another source.

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**.

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

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

- *Security Plugins* can be downloaded in the following packages:

  Non-Evaluation:

  - **rti_security_plugins-<*version*>-host-<*host platform*>.rtipkg**,
    which 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-<*version*>-target-<*target architecture*>.rtipkg**,
    which contains the Security Plugins libraries you will link against.

  Evaluation:

  - **rti_security_plugins-<*version*>-eval-<*target architecture*>.rtipkg**,
    which includes the compiler-independent Security Plugins dependencies (documentation, headers, and the libraries used by RTI tools and services) for the host platform and the Security Plugins evaluation libraries you will link against for your target platform.

- OpenSSL:
    - OpenSSL distribution files for RTI tools and services follow the naming convention: **openssl-<*version*>-host-<*host platform*>.rtipkg**.
    - OpenSSL distribution files to link against your application follow the naming convention: **openssl-<*version*>-target-<*target architecture*>.tar.gz** (or **.zip** on Windows systems).

  For the currently supported OpenSSL version number, see the *RTI Security Plugins Release Notes*. Architecture names are described in the *RTI Connext DDS Core Libraries Platform Notes*. For example:
    - Bundle with distribution files for RTI tools and services: **openssl-1.1.1d-host-x64Win64.rtipkg.**
    - Bundle with distribution files to link against your application: **openssl-1.1.1d-target-x64Win64VS2013.zip.**

Chapter 4 Installation Instructions

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

These instructions assume you are installing *Security Plugins* 6.0.1 and OpenSSL 1.1.1d. See the *RTI Security Plugins Release Notes* for the currently supported versions.

4.1 Installing an Evaluation Version

1. In the evaluation version, the *Security Plugins* and OpenSSL are installed automatically when you install the *Connext DDS* host bundle (this is described in the *RTI Connext DDS Installation Guide*). You do not need to install a target bundle.

   After installation, the *Security Plugins* header files and libraries will be under **include/ndds/security** and **lib/*<target architecture>***, respectively.
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 command all on one line, adjust the path to match your installation directory, and use your own architecture string):

   On UNIX-based systems:

   ```
   > setenv PATH
     <installdir>/third_party/openssl-1.1.1d/<architecture>/release/bin:${PATH}
   ```

   On Windows systems:

   ```
   > set PATH=
     <installdir>\third_party\openssl-1.1.1d\<architecture>\release\bin;%PATH%
   ```
3. On UNIX-based systems: If linking dynamically, add OpenSSL's **/lib** directory in your **LD_LIBRARY_PATH**.

   For example, assuming you want to use the *release* version of the OpenSSL libraries:

```
> setenv LD_LIBRARY_PATH
  <installdir>/third_party/openssl-1.1.1d/<architecture>/release/lib:$LD_ LIBRARY_PATH
```

4. To verify your installation, enter:

```
> openssl version
```

You should see a response similar to:

```
OpenSSL 1.1.1d
```

5. Your *Security Plugins* distribution may require a license. See Chapter 6 License Management on page 16.

4.2 Installing a Non-Evaluation Version

4.2.1 UNIX-Based Systems

1. Install the *Connext DDS* host and target bundles on top of each other, as described in the *RTI Connext DDS Installation Guide*.
2. Install the *Security Plugins* host and target packages to enable security for your applications. The security header files and libraries will be under **include/ndds/security** and **lib/<*target architecture*>**, respectively:
   - **rti_security_plugins-6.0.1-host-<*host platform*>.rtipkg**
   - **rti_security_plugins-6.0.1-target-<*target architecture*>.rtipkg**

   (Where <*host platform*> is **i86Linux**, **x64Linux**, or **darwin** and <*target architecture*> is one of the supported platforms, see the *RTI Security Plugins Release Notes*).
3. If you want to enable security for RTI tools and services, install an OpenSSL host package from RTI:
   - **openssl-1.1.1d-host-<*host platform*>.rtipkg**
4. Install the OpenSSL target package by following 4.3 Installing OpenSSL on the next page.

This completes the installation process.

4.2.2 Windows Systems

1. Install the *Connext DDS* host and target bundles on top of each other, as described in the *RTI Connext DDS Installation Guide*.
2. Install the *Security Plugins* host and target packages to enable security for your applications. The security header files and libraries will be under **include/ndds/security** and **lib/<*target architecture*>**, respectively:

- **rti_security_plugins-6.0.1-host-<*host platform*>.rtipkg**
- **rti_security_plugins-6.0.1-target-<*target architecture*>.rtipkg**

(Where <*host platform*> is **i86Win32** or **x64Win64**, and <*target architecture*> is one of the supported platforms, see the *RTI Security Plugins Release Notes*).

3. If you want to enable security for RTI tools and services, install an OpenSSL host package from RTI:
   - **openssl-1.1.1d-host-<*host platform*>.rtipkg**
4. Install the OpenSSL target package by following 4.3 Installing OpenSSL below.

This completes the installation process.

4.3 Installing OpenSSL

4.3.1 UNIX-Based Systems

1. Make sure you have installed *Security Plugins* packages as described in 4.1 Installing an Evaluation Version on page 8 or 4.2 Installing a Non-Evaluation Version on the previous page.
2. Install an OpenSSL target package from RTI: **openssl-1.1.1d-target-<*target architecture*>.tar.gz**.
   a. Make sure you have GNU's version of the tar utility, **gtar** (which handles long file names), and GNU's version of the unzip utility, **gunzip**.
   b. Move the downloaded OpenSSL distribution file to a directory of your choice, such as **/local/rti**, and change to that directory:

   ```
   $ cd /local/rti
   ```

   c. Use **gunzip** to uncompress the OpenSSL file. (This is not the same as the OpenSSL host package in the previous step.) For example (your filename may be different):

   ```
   $ gunzip openssl-1.1.1d-target-armv7aQNX6.6.0qcc_cpp4.7.3.tar.gz
   ```

   d. Use **gtar** to extract the distribution from the uncompressed file. For example:

   ```
   $ gtar xvf openssl-1.1.1d-target-armv7aQNX6.6.0qcc_cpp4.7.3.tar
   ```

   This will extract files into **/local/rti/openssl-1.1.1d**.
   e. Include the resulting **/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):

   ```
   $ setenv PATH
     /local/rti/openssl-1.1.1d/armv7aQNX6.6.0qcc_cpp4.7.3/release/bin:${PATH}
   ```

   f. If linking dynamically, include the resulting **/lib** directory in your LD_LIBRARY_PATH. For example, assuming you want to use the "release" version of the OpenSSL libraries (enter the command all on one line):

```
$ setenv LD_LIBRARY_PATH
  /local/rti/openssl-1.1.1d/armv7aQNX6.6.0qcc_cpp4.7.3/release/lib:$LD_LIBRARY_PATH
```

g. To verify your installation, enter:

```
$ openssl version
```

You should see a response similar to:

```
OpenSSL 1.1.1d
```

If you get a different version than OpenSSL 1.1.1d, your PATH may be pointing with a higher precedence to a different version of OpenSSL. You may need to place version 1.1.1d 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 environmental variable OPENSSL_CONF with the path to the **openssl.cnf** file you are using. For example:

```
$ setenv OPENSSL_CONF /local/rti/openssl-1.1.1d/armv7aQNX6.6.0qcc_
cpp4.7.3/release/ssl/openssl.cnf
```

4.3.2 Windows Systems

1. Install an OpenSSL host package from RTI: o**penssl-1.1.1d-host-<*host platform*>.rtipkg**.
2. Install an OpenSSL target package from RTI: **openssl-1.1.1d-target-<*target architecture*>.zip**.
   a. Right-click the distribution file and extract the contents in a directory of your choice.
   b. Add the resulting **bin** directory to your **Path** environment variable:

      c:\rti\openssl-1.1.1d\<*target architecture*>\release\bin

      (If you need help with this process, please see *RTI Connext DDS Core Libraries Getting Started Guide*.)
   c. To verify your installation, open a command prompt and enter:

```
> openssl version
```

You should see a response similar to:

```
OpenSSL 1.1.1d
```

If you get a different version than OpenSSL 1.1.1d, your PATH may be pointing with a higher precedence to a different version of OpenSSL. You may need to place version 1.1.1d 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 environmental variable OPENSSL_CONF with the path to the openssl.cnf file you are using. For example:

```
> set OPENSSL_CONF=c:\rti\openssl-1.1.1d\<target
architecture>\release\ssl\openssl.cnf
```

Chapter 5 Libraries Required for Using RTI Security Plugins

To use the *RTI Security Plugins*, link against the additional libraries in one of the following tables, depending on your platform. Select the files appropriate for your chosen library format.

Table 5.1 Additional Libraries for Using RTI Security Plugins on Android Systems

Library Format	RTI Security Plugins Libraries [a]	OpenSSL Libraries [b]
Dynamic Release	libnddssecurity.so	librtisslsupport.so
Dynamic Debug	libnddssecurityd.so	librtisslsupport.so
Static Release	libnddssecurityz.a	librtisslsupportz.a
Static Debug	libnddssecurityzd.a	librtisslsupportz.a

[a] These libraries are in <NDDSHOME>/lib/<architecture>.

[b] These libraries are in <openssl install dir>/<architecture>/<debug or release dir>/lib.

Table 5.2 Additional Libraries for Using RTI Security Plugins on iOS Systems

Library Format	RTI Security Plugins Libraries [a]	OpenSSL Libraries [b]
Static Release	libnddssecurityz.a	libsslz.a libcryptoz.a
Static Debug	libnddssecurityzd.a	libsslz.a libcryptoz.a

[a] These libraries are in <NDDSHOME>/lib/<architecture>.

[b] These libraries are in <openssl install dir>/<architecture>/<debug or release dir>/lib.

Table 5.3 Additional Libraries for Using RTI Security Plugins on UNIX-Based Systems

Library Format	RTI Security Plugins Libraries [a]	OpenSSL Libraries [b]
Dynamic Release	libnddssecurity.so	libssl.so libcrypto.so
Dynamic Debug	libnddssecurityd.so	libssl.so libcrypto.so
Static Release	libnddssecurityz.a	libsslz.a libcryptoz.a
Static Debug	libnddssecurityzd.a	libsslz.a libcryptoz.a

[a] These libraries are in <NDDSHOME>/lib/<architecture>.

[b] These libraries are in <openssl install dir>/<architecture>/<debug or release dir>/lib.

Table 5.4 Additional Libraries for Using RTI Security Plugins on macOS Systems

Library Format	RTI Security Plugins Libraries [a]	OpenSSL Libraries [b]
Dynamic Release	libnddssecurity.dylib	libssl.dylib libcrypto.dylib
Dynamic Debug	libnddssecurityd.dylib	libssl.dylib libcrypto.dylib
Static Release	libnddssecurityz.a	libsslz.a libcryptoz.a
Static Debug	libnddssecurityzd.a	libsslz.a libcryptoz.a

[a]These libraries are in <NDDSHOME>/lib/<architecture>.

[b]These libraries are in <openssl install dir>/<architecture>/<debug or release dir>/lib.

Table 5.5 Additional Libraries for Using RTI Security Plugins on QNX Systems

Library Format	RTI Security Plugins Libraries [a]	OpenSSL Libraries [b]
Dynamic Release	libnddssecurity.so	libssl.so libcrypto.so
Dynamic Debug	libnddssecurityd.so	libssl.so libcrypto.so
Static Release	libnddssecurityz.a	libsslz.a libcryptoz.a
Static Debug	libnddssecurityzd.a	libsslz.a libcryptoz.a

[a]These libraries are in <NDDSHOME>/lib/<architecture>.

[b]These libraries are in <openssl install dir>/<architecture>/<debug or release dir>/lib.

Table 5.6 Additional Libraries for Using RTI Security Plugins on Windows Systems

Library Format	RTI Security Plugins Libraries [a]	OpenSSL Libraries [b]
Dynamic Release	nddssecurity.lib	ssleay32.lib libeay32.lib
Dynamic Debug	nddssecurityd.lib	ssleay32.lib libeay32.lib
Static Release	nddssecurityz.lib	ssleay32z.lib libeay32z.lib
Static Debug	nddssecurityzd.lib	ssleay32z.lib libeay32z.lib

[a]These libraries are in <NDDSHOME>/lib/<architecture>.
[b]These libraries are in <openssl install dir>\<architecture>\<debug, release, static_debug, or static_release dir>\lib.

Chapter 6 License Management

Most package types (Professional, Secure, Basic, and Evaluation) require a license file in order to run.

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

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.

6.1 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 an evaluation 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 "), apo-

strophes (') (replace with '), greater-than (>) (replace with >), less-than (<) (replace with <), and ampersands (&) (replace with &).

Example XML file:

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

3. A property called **com.rti.serv.secure.license_file**.
4. A property called **dds.license.license_file**. (Only if you have an evaluation 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**:

```
<participant_qos>
    <property>
        <value>
            <element>
                <name>dds.license.license_file</name>
                <value>path to license file</value>
            </element>
        </value>
    </property>
</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 4.)
6. If you are running any of the tools/services as executables via **NDDSHOME/bin/<e***xecutable 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**.

6.2 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 7 Next Steps

Refer to the *Security Plugins User's Manual* for further information on setting up and using *Security Plugins*. In particular, see the chapter "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.

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