

RTI Connex

Installation Guide

Version 7.3.0



Trademarks

RTI, Real-Time Innovations, Connex, Connex 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 Installing RTI Connex	
1.1 Installing a Regular Version	1
1.1.1 Installing on a Linux System	3
1.1.2 Installing on a Windows System	4
1.1.3 Installing on a macOS System	6
1.2 Installing an Evaluation or LM Version	7
1.2.1 Installing an Evaluation or LM Version on a Linux System	7
1.2.2 Installing an Evaluation or LM Version on a Windows System	8
1.2.3 Installing an Evaluation or LM Version on a macOS System	9
1.2.4 Telemetry Data Collected by Evaluation Version	10
1.3 After Installation	10
Chapter 2 Paths Mentioned in Documentation	13
Chapter 3 License Management	
3.1 Installing the License File	15
3.2 Adding or Removing License Management	17
3.3 Activation Library	18
Chapter 4 Controlling Location for RTI Workspace and Copying of Examples	21
Chapter 5 Special Backup of RTI Libraries	
5.1 Overwriting Libraries	23
5.2 Installing Add-on Products when Using RTI Tools	23
Chapter 6 Uninstalling Connex	25

Chapter 1 Installing RTI Connex

RTI® *Connex*® is delivered as static and dynamic libraries with development tools, examples, and documentation that will help you use the libraries.

If you are installing a purchased version of *Connex*, see [1.1 Installing a Regular Version below](#). If you are evaluating *Connex* or using it for research purposes, see [1.2 Installing an Evaluation or LM Version on page 7](#). Note that all evaluation and LM versions require a license key file to run. All use of both regular and LM versions is subject to RTI's Software License Agreement #4000. The evaluation version is subject to RTI's Software Evaluation License Agreement #4045.

1.1 Installing a Regular Version

Connex distributions are provided in two types of bundles: *hosts* and *targets*.

- A *host* bundle contains non-target-specific files, such as documentation, header files, tools, and infrastructure services executables. The host bundle is provided in a **.run**, **.exe**, or **.dmg** file that will run an installer.

Host bundles are named:

- **rti_connex_dds-<version>-<package_type>-host-<host-platform>.<extension>**

The `<package_type>` is usually **pro**.

Note: The legacy **core** package type does not include all functionality. Please contact your local RTI sales representative or email sales@rti.com for more information.

The `<host_platform>` depends on your development platform: **x64Linux** for Linux® systems, **x64Win64** for Windows® systems, or **x64Darwin** or **arm64Darwin** for macOS® systems.

The `<extension>` is **run** for Linux systems, **exe** for Windows systems, or **dmg** for macOS systems.

- A *target* bundle contains libraries and any applications that have target-specific dependencies. A target bundle is an “RTI Package” (*.rtipkg) file that must be installed on top of your host bundle.

Target bundles are named:

- **rti_connex_dds-<version>-<package_type>-target-<target_architecture>.rtipkg**

The <target_architecture> depends on your target machine (where you will deploy your completed application). These strings are listed in the [RTI Connex Core Libraries Platform Notes](#); examples are **x64Win64VS2017** and **x64Linux4gcc7.3.0**.

Note: You may have more than one target bundle if you are developing for multiple targets.

Note: If you are developing C# applications, you need to download NuGet packages as well. For detailed instructions, see [Installing Connex](#) in the [RTI Connex Getting Started Guide for C#](#).

If you are installing a patch release (for example, a 7.3.0.x release), read [Chapter 5 Special Backup of RTI Libraries on page 23](#).

There are two ways to install a *Connex* host bundle:

- using an installer, which will prompt you for information such as where to install, or
- using a script, which allows you to run in "unattended mode" so you will not be prompted for information while it is installing.

Both ways are described below.

Running from a Script (Unattended Mode)

When running in unattended mode, the installer will use default values for its parameters unless you specify them at the command line (see [Table 1.1 Command-Line Options when Installing from a Script](#)).

Table 1.1 Command-Line Options when Installing from a Script

Parameter	Values	Meaning	Default
--mode	win32 gtk xwindows osxqt text [not available on Windows] unattended [all platforms] [not available for "eval" or "lm" package]	Should the installer display a graphical or a text-based UI? Should the installer ask for user feedback?	GUI available on your platform
--unattendedmodeui	none [not available on Windows] minimal [not available on Windows] minimalWithDialogs [required on Windows if using --mode unattended]	If you chose --mode unattended, how much graphical feedback should the installer display?	none
--prefix	directory to install into	The installation directory the product will be placed into. Note that the installer will always place the product in a directory named rti_connex_tdds-7.3.0. See Chapter 2 Paths Mentioned in Documentation on page 13	Platform-dependent Windows Example: c:\program files\rti_connex_tdds-7.3.0
--disable_copy_examples	true false	Should the installer disable the copying of examples into rti_workspace during installation (true) or not (false)?	false

1.1.1 Installing on a Linux System

In these instructions, we assume your host bundle is named **rti_connex_tdds-7.3.0-pro-host-x64Linux.run** and your target bundle is named **rti_connex_tdds-pro-target-7.3.0-x64Linux4gcc7.3.0.rtipkg**. Your filenames will be different depending on your package type and architecture.

1. Download the host and target bundles to a location of your choice; for example, **/home/<your user name>/Downloads**.
2. Install the host bundle. As described above, you can use an installer or a script:
 - Using the installer:

Run the host bundle installer (the **.run** file). For example:

```
/home/<your user name>/Downloads/rti_connex_t_7.3.0-dds-pro-host-x64Linux.run
```

Follow the on-screen prompts. You will be asked if you want to use the default installation directory (**/home/<your user name>**) or specify a different path. We refer to the install-

ation directory as `<NDDSHOME>`, see [Chapter 2 Paths Mentioned in Documentation on page 13](#).

- Using a script:

Run from a command prompt:

```
rti_connex_dds-7.3.0-<package_type>-host-<host_platform>.run
--mode unattended --prefix <installation path>
```

The `--prefix` option allows you to specify the installation directory; if you omit this, the installer will install into a default location.

For example (enter on one line):

```
rti_connex_dds-7.3.0-pro-host-x64Linux.run
--mode unattended --prefix /home/user/rti_connex_dds-7.3.0
```

3. Install the target bundle(s). There are two ways to do this:

- From *RTI Launcher*:

Start *RTI Launcher* (`<NDDSHOME>/bin/rtilauncher`), select the **Configuration** tab, and click on **Install RTI Packages**. Add the location of your target file. Click **Install**.

- From the command line:

Open a command prompt and change to the `rti_connex_dds-7.3.0/bin` directory. Use `<NDDSHOME>/bin/rtipkinstall` to install the target bundle. For example (enter on one line):

```
rtipkinstall
/home/<your user name>/Downloads/rti_connex_dds-7.3.0-pro-target-
x64Linux4gcc7.3.0.rtipkg
```

4. Read [Chapter 3 License Management on page 15](#).
5. See [1.3 After Installation on page 10](#).

1.1.2 Installing on a Windows System

In these instructions, we assume your host bundle is named `rti_connex_dds-7.3.0-pro-host-x64Win64.exe` and your target bundle is named `rti_connex_dds-7.3.0-pro-target-x64Win64VS2017.rtipkg`. Your filenames will be different depending on your package type and architecture.

Depending on your version of Windows and where you want to install these files, your user account may or may not require administrator privileges.

1. Download the host and target bundles to a location of your choice; for example, `C:\Users\<your user name>\Downloads`.

2. Install the host bundle. As described above, you can use an installer or a script:

- Using the installer:

From your file explorer, open (double-click) the host bundle installer (the **.exe** file).

Follow the on-screen prompts. You will be asked if you want to use the default installation directory (**C:\Program Files**) or specify a different path.

- Using a script:

Run from a command prompt:

```
rtd_connext_dds-<version>-<package_type>-host-<host_platform>.exe
--mode unattended --unattendedmodeui minimalWithDialogs --prefix <installation
path>
```

The **--prefix** option allows you to specify the installation directory. If you do not specify a prefix, the installer will install into a default location.

For example (enter on one line):

```
rtd_connext_dds-7.3.0-pro-host-x64Win64.exe --mode unattended
--unattendedmodeui minimalWithDialogs --prefix c:\program files\rtd_connext_dds-
7.3.0
```

When running in unattended mode on a Windows system, you must specify **--unattendedmodeui minimalWithDialogs** or the installation will fail.

3. Install the target bundle(s). There are two ways to do this:

- From *RTI Launcher*:

Start *RTI Launcher* (<NDDSHOME>\bin\rtilauncher or from the Windows **Start** menu), select the **Configuration** tab, and click on **Install RTI Packages**. Add the location of your package file. Click **Install**.

- From the command line:

Open a command prompt and change to the **rtd_connext_dds-7.3.0\bin** directory. Use the **rtipkginstall** batch file to install the target bundle. For example (enter on one line):

```
rtipkginstall C:\Users\<your user name>\Downloads\rtd_connext_dds-7.3.0-pro-
target-x64Win64VS2017.rtipkg
```

4. Read [Chapter 3 License Management on page 15](#).

5. See [1.3 After Installation on page 10](#).

1.1.3 Installing on a macOS System

In these instructions, we assume your host bundle is named **rti_connex_dds-7.3.0-pro-host-arm64Darwin.dmg** and your target bundle is named **rti_connex_dds-7.3.0-pro-target-arm64Darwin20clang12.0.rtipkg**. Your filenames may be different depending on your package type and architecture.

1. Download the host and target bundles to a location of your choice; for example, **/Users/<your user name>/Downloads**.
2. Double-click on the host bundle **.dmg** file. You should see a screen appear.
3. Double-click on the icon in the screen to run the installer application.
4. Follow the on-screen prompts. You will be asked if you want to use the default installation directory (**/Applications/rti_connex_dds-7.3.0**) or specify a different path. Note that the installer will always place the product inside a folder named **rti_connex_dds-7.3.0**. We refer to the installation directory as **<NDDSHOME>**, see [Chapter 2 Paths Mentioned in Documentation on page 13](#).

Note: If you do not install *Connex* in the default installation directory (**/Applications/rti_connex_dds-7.3.0**), the *Launcher* icon might not display in the macOS Launchpad (see the [Launchpad User Guide](#)). You can still access the *Launcher* application in the **rti_connex_dds-7.3.0** directory.

5. Install the target bundle(s). There are two ways to do this:

- From *RTI Launcher*:

Start *RTI Launcher* (**<NDDSHOME>/RTI Launcher**), select the Configuration tab, and click on **Install RTI Packages**. Add the location of your target file. Click **Install**.

- From the command line:

Open a command prompt and change to the **rti_connex_dds-7.3.0/bin** directory. Use **<NDDSHOME>/bin/rtipkginstall** to install the target bundle. For example (enter on one line):

```
rtipkginstall /Users/<your user name>/Downloads/rti_connex_dds-7.3.0-pro-target-arm64Darwin20clang12.0.rtipkg
```

6. Read [Chapter 3 License Management on page 15](#).
7. See [1.3 After Installation on page 10](#).

1.2 Installing an Evaluation or LM Version

In addition to *Connex Professional*, the evaluation ("eval") and LM ("lm") packages include [RTI Real-Time WAN Transport](#), [RTI Cloud Discovery Service](#), [RTI Security Plugins](#), and OpenSSL® version 3.0.12. (The installer provides a prebuilt version of OpenSSL 3.0.12. If you wish to build your own version of OpenSSL 3.0.12, you can find the source code here: <https://www.openssl.org/source/>.)

Evaluation and LM distributions usually combine the *host* and *target* in a single bundle, named after the *target*. For example, `rti_connex_dds-7.3.0-eval-x64Win64VS2017.exe` combines the `x64Win64` host with the `x64Win64VS2017` target in a single evaluation bundle; the same is true for the `rti_connex_dds-7.3.0-lm-x64Win64VS2017.exe` LM bundle. The target architecture depends on your target machine where you will deploy your completed application. These strings are listed in the [RTI Connex Core Libraries Platform Notes](#); examples are `x64Win64VS2017`, `x64Linux4gcc7.3.0`, and `arm64Darwin20clang12.0`.

Some evaluation and LM distributions require you to install two bundles:

- A combined host/target bundle (`.run`, `.exe`, or `.dmg`), such as `rti_connex_dds-7.3.0-lm-x64Linux4gcc7.3.0.run` or `rti_connex_dds-7.3.0-eval-x64Linux4gcc7.3.0.run`.
- A separate LM target package (`.rtipkg`) for the specific target architecture, such as `rti_connex_dds-7.3.0-lm-target-armv8Linux4gcc7.3.0.rtipkg`.

For example, suppose your target machine uses Ubuntu 18.04 LTS on an Arm v8 CPU. RTI does not currently provide a combined evaluation or LM host/target bundle for this architecture. So first you would install an evaluation/LM bundle for any Linux architecture, then install the LM target package for your Ubuntu 18.04 LTS target on an Arm v8 CPU. The instructions below explain how.

Note: Evaluation and LM packages cannot be installed in "unattended mode."

1.2.1 Installing an Evaluation or LM Version on a Linux System

1. Download the evaluation or LM bundle (for example: `rti_connex_dds-7.3.0-eval-x64Linux4gcc7.3.0.run` or `rti_connex_dds-7.3.0-lm-x64Linux4gcc7.3.0.run`) to a location of your choice; for example, `/home/<your user name>/Downloads`.
2. Run the eval/lm bundle installer (the `.run` file). For example:

```
/home/<your user name>/Downloads/rti_connex_dds-7.3.0-eval-x64Linux4gcc7.3.0.run
```

Follow the on-screen prompts. You will be asked if you want to use the default installation directory (`/home/<your user name>`) or specify a different path. We refer to the installation directory as `<NDDSHOME>`, see [Chapter 2 Paths Mentioned in Documentation on page 13](#).

3. If your target architecture does not have an Arm or x86 CPU, skip this step.

For Linux targets with an Arm or x86 CPU: you also need to install an LM target package.

This package will have "**lm-target-**" in the filename. You can install it on top of any evaluation or LM bundle for a Linux architecture. That is, the **.run** file and the **.rtipkg** file do not have to use the same Linux architecture string. For example, you can install **rti_connex_dds-7.3.0-lm-target-armv8Linux4gcc7.3.0.rtipkg** on top of **rti_connex_dds-7.3.0-eval-x64Linux4gcc7.3.0.run**.

There are two ways to install the LM target package:

- From *RTI Launcher*:

Start *RTI Launcher* (<NDDSHOME>/bin/rtilauncher), select the **Configuration** tab, and click on **Install RTI Packages**. Add the location of your target file. Click **Install**.

- From the command line:

Open a command prompt and change to the **rti_connex_dds-7.3.0/bin** directory. Use <NDDSHOME>/bin/rtipkginstall to install the target bundle. For example (enter on one line):

```
rtipkginstall
/home/<your user name>/Downloads/rti_connex_dds-7.3.0-lm-target-
armv8Linux4gcc7.3.0.rtipkg
```

4. Read [Chapter 3 License Management on page 15](#).
5. See [1.3 After Installation on page 10](#).

1.2.2 Installing an Evaluation or LM Version on a Windows System

Depending on your version of Windows and where you want to install these files, your user account may or may not require administrator privileges.

1. Download the evaluation or LM bundle (for example: **rti_connex_dds-7.3.0-eval-x64Win64VS2017.exe** or **rti_connex_dds-7.3.0-lm-x64Win64VS2017.exe**) to a location of your choice; for example, **C:\Users\<your user name>\Downloads**.
2. Install the eval/lm bundle:

From your file explorer, open (double-click) the bundle installer (the **.exe** file).

Follow the on-screen prompts. You will be asked if you want to use the default installation directory (**C:\Program Files**) or specify a different path.

3. If your target architecture does not have an Arm CPU, skip this step.

For Windows targets with an Arm CPU: you also need to install an LM target package.

This package will have “**lm-target-arm**” in the filename and the architecture will start with “**arm**”. You can install it on top of any evaluation or LM bundle for a Windows architecture. That is, the **.exe** file and the **.rtipkg** file do not have to use the same Windows architecture string. For example, you can install **rti_connex_t_dds-7.2.0-lm-target-arm64Win64VS2022.rtipkg** on top of **rti_connex_t_dds-7.2.0-eval-x64Win64VS2017.exe**.

There are two ways to install the LM target package:

- From *RTI Launcher*:

Start *RTI Launcher* (<NDDSHOME>/bin/rtilauncher), select the **Configuration** tab, and click on **Install RTI Packages**. Add the location of your target file. Click **Install**.

- From the command line:

Open a command prompt and change to the **rti_connex_t_dds-7.3.0/bin** directory. Use <NDDSHOME>/bin/rtipkginstall to install the target bundle. For example (enter on one line):

```
rtipkginstall.bat
  C:\User\\Downloads\rti_connex_t_dds-7.2.0-lm-target-
arm64Win64VS2022.rtipkg
```

4. Read [Chapter 3 License Management on page 15](#).
5. See [1.3 After Installation on the next page](#).

1.2.3 Installing an Evaluation or LM Version on a macOS System

1. Download the evaluation or LM bundle (for example: **rti_connex_t_dds-7.3.0-lm-arm64Darwin20clang12.0.dmg**) to a location of your choice; for example, **/Users/<your user name>/Downloads**.
2. Double-click on the host bundle **.dmg** file. You should see a screen appear.
3. Double-click on the icon in the screen to run the installer application.
4. Follow the on-screen prompts. You will be asked if you want to use the default installation directory (**/Applications/rti_connex_t_dds-7.3.0** or specify a different path. Note that the installer will always place the product inside a folder named **rti_connex_t_dds-7.3.0**. We refer to the installation directory as <NDDSHOME>, see [Chapter 2 Paths Mentioned in Documentation on page 13](#).

Note: If you do not install *Connex_t* in the default installation directory (**/Applications/rti_connex_t_dds-7.3.0**), the *Launcher* icon might not display in the macOS Launchpad (see the

[Launchpad User Guide](#)). You can still access the *Launcher* application in the `rti_connex_dds-7.3.0` directory.

5. Read [Chapter 3 License Management on page 15](#).
6. See [1.3 After Installation below](#).

1.2.4 Telemetry Data Collected by Evaluation Version

After installing the *Connex*, you will see a `TELEMETRY_DATA.json` file, which contains general product usage information without any personal data or personal identifiers. If usage data sharing is enabled, this is the only file that is shared with RTI, for "eval" bundles only.

Note: Although you will see the `TELEMETRY_DATA.json` file in all *Connex* installations, usage data sharing occurs only with the evaluation bundle (bundles with "eval" in their name). See information about the ["Preferences Dialog," in the Configuration Tab section of the RTI Launcher User's Manual](#) to learn how to enable/disable usage data sharing in the evaluation bundle.

`TELEMETRY_DATA.json` is in the following locations:

- macOS systems: `/Users/<your user name>/rti_workspace/7.3.0/.tdata/`
- Linux systems: `/home/<your user name>/rti_workspace/7.3.0/.tdata/`
- Windows systems: `<your Windows documents folder>\rti_workspace\7.3.0\tdata\`

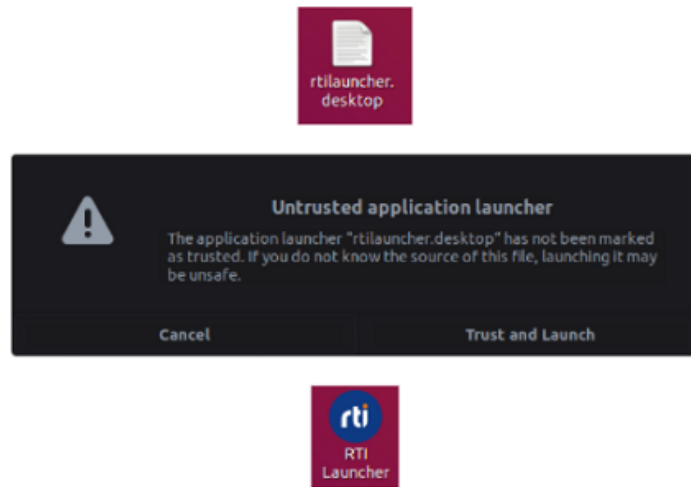
Where 'your Windows documents folder' depends on your version of Windows. For example, on Windows 10, the folder is `C:\Users\<your user name>\Documents`.

1.3 After Installation

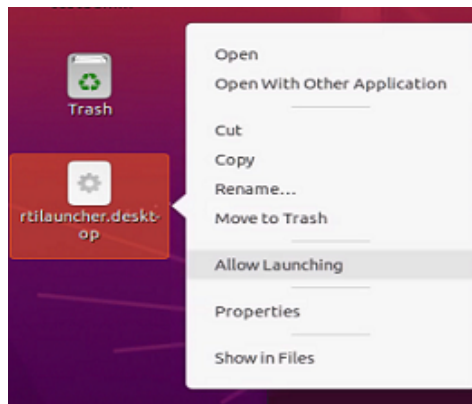
After installation:

- From *Launcher*, right-click each icon to access documentation about that component. See the *RTI Launcher User's Manual* (in `<NDDSHOME>/doc/manuals/connex_dds_professional/tools/launcher`).
- See the *RTI Connex Getting Started Guide* (in `<NDDSHOME>/doc/manuals/connex_dds_professional/getting_started_guide`).

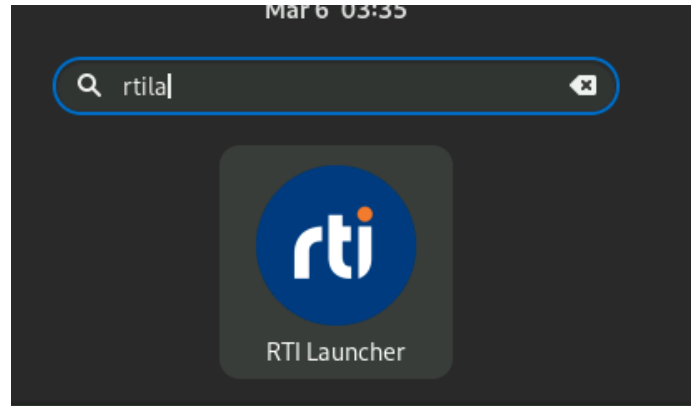
Note: On some Linux operating systems, executing the desktop shortcut for *Launcher* might report that the application has not been marked as trusted. This is a harmful warning that will prevent you from executing *Launcher* using the desktop shortcut. To remove this warning, click "Trust and Launch":



This will mark the desktop shortcut as a trusted application, allowing you to run the application. If the warning doesn't appear and the desktop shortcut still doesn't run, right-click the shortcut and click "Allow Launching":



On some Linux operating systems, the desktop shortcut might not execute and will need to be run through the terminal or in the Activities panel. In some newer versions of GNOME, the "Allow Launching" option is missing from the context menu, preventing the desktop shortcut from being run. *Launcher* can still be run from the Activities panel:



Or you can run *Launcher* through the terminal by entering the following command:

```
gtk-launch rtilauncher.desktop
```


Chapter 2 Paths Mentioned in Documentation

The documentation refers to:

- **<NDDSHOME>**

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

- macOS® systems:
/Applications/rti_connex_dds-7.3.0
- Linux systems, *non-root* user:
/home/<your user name>/rti_connex_dds-7.3.0
- Linux systems, *root* user:
/opt/rti_connex_dds-7.3.0
- Windows® systems, user without Administrator privileges:
<your home directory>\rti_connex_dds-7.3.0
- Windows systems, user with Administrator privileges:
C:\Program Files\rti_connex_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 [Chapter 4 Controlling Location for RTI Workspace and Copying of Examples on page 21](#).

Chapter 3 License Management

There's a distinction between a license (or "activation") *file* or *string*, and a *license*. When you buy an RTI product, such as *RTI Connex Professional*, you are *licensed* to use it.

Within a package, some components may have to be activated using a license file or string, while others may not. For example, when you install the *RTI Connex Professional* package, the Core Libraries (which you are licensed to use) do not require a license file or string. However, tools like *RTI Admin Console* (which you are also licensed to use) require a license file.

Some components that do not require a license file in certain packages may require a license file in other packages. For example, the Core Libraries require a license file in the evaluation ("eval") and LM ("lm") packages. If a component in your *Connex* distribution requires a license file, you will receive one from RTI.

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.

In the rest of this section, the term *license file* is used interchangeably with *activation file*. We also use the term *file* generically to refer to both an actual file and the string with the contents of the file.

3.1 Installing the License File

You can install a license file using the six different mechanisms below, in order of priority. The first two mechanisms only activate the *Connex* Core Libraries in the LM or evaluation bundles. The other mechanisms apply to all the *Connex* components that require a license file, including the *Connex* Core Libraries in the LM and evaluation bundles.

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.

In all components (such as *Admin Console*) except for the *Connex* Core Libraries, each time the component starts, it will look for the license file in the locations listed below until it finds a valid license.

For the *Connex* Core Libraries in the LM or evaluation bundles, your application's call to **DomainParticipantFactory.create_participant()** will return null, preventing communication, if it cannot find a valid license.

1. In the PropertyQosPolicy of the *DomainParticipant*, there may be a property called **dds.license.license_string**. (Only if you have an "eval" or "lm" version of *Connex*.) 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.) You can set the property either in source code or in an XML file. An XML file example is shown below:

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

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 **&**).

2. In the PropertyQosPolicy of the *DomainParticipant*, there may be a property called **dds.license.license_file**. (Only if you have an "eval" or "lm" version of *Connex*.) The value for this property 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>
```

```
</property>
</domain_participant_qos>
```

3. 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 13.](#))

4. If you are running any of the tools/services as executables via `NDDSHOME/bin/<executable script>` or through *Launcher*:
 - In your `rti_workspace/<version>` directory, in a file called **rti_license.dat**.
 - In your `rti_workspace` directory, in a file called **rti_license.dat**.
 - In `<NDDSHOME>` (the *Connex*t installation directory), in a file called **rti_license.dat**.
5. If you are running your own application linked with *Connex*t libraries:
 - In your current working directory, in a file called **rti_license.dat**.
 - In `<NDDSHOME>` (the *Connex*t installation directory), in a file called **rti_license.dat**.
6. You can also provide a license string by creating a shared library called **rti_activation**, which implements the activation API. See [3.3 Activation Library on the next page.](#)

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

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

3.2 Adding or Removing License Management

If you are using a package type that requires a license file and 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 *Connex*t.

However, if you switch from an evaluation ("eval") or LM ("lm") distribution of *Connex*t to a regular distribution, 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.

3.3 Activation Library

Providing a license string by copying a license file into the filesystem where the *Connex*t applications run may not be feasible. For example, you may have your own license server from which you want to obtain all the licenses required by your applications, including the *Connex*t licenses.

To help support these use cases, *Connex*t allows you to provide a license string by implementing the C activation API in a shared library called **rti_activation** (e.g., **librti_activation.so**, **rti_activation.dll**). This activation mechanism is available on all platforms but VxWorks.

The C activation API is provided in the header file **ndds/dl_driver/dl_driver_activation_user.h**.

You must implement the following three APIs:

```
/**
 * @brief Get the activation string.
 *
 * @pre buffer can be NULL if *size is 0.
 * @pre size cannot be NULL.
 * @pre exception cannot be NULL.
 *
 * @param buffer Out. Buffer where the activation string will be copied.
 * @param size In/Out. In: Size of the buffer. Out: Size of the activation
 * string. The size includes the null terminator.
 * @param user_data In. User data returned by RTI_ActivationLibrary_initialize().
 * @param exception Out. Exception object. If there is no error, code will be
 * set to RTI_ACTIVATION_LIBRARY_EXCEPTION_OK. If there is an error, code
 * will be set to any of the predefined errors
 * (RTI_ACTIVATION_LIBRARY_EXCEPTION_ERROR or
 * RTI_ACTIVATION_LIBRARY_EXCEPTION_INSUFFICIENT_SPACE) or a user-defined
 * error code. The exception message will contain a description of the error.
 *
 * If the input buffer is not big enough to hold the activation string, the
 * function will return RTI_ACTIVATION_LIBRARY_EXCEPTION_INSUFFICIENT_SPACE
 * and *size will be set to the required size of the activation string.
 *
 * This function can be called in parallel by multiple threads.
 *
 * @return 0 if there is an error, 1 otherwise.
 */
DLDRIVERDllExport
int RTI_ActivationLibrary_get_activation_string(
    char *buffer,
    unsigned int *size,
    void *user_data,
    struct RTI_ActivationLibraryException *exception);

/**
 * @brief Initialize the activation library.
 *
 * This function will be called once after loading the activation shared
 * library.
```

```

*
* @param user_data Out. User data. The library can initialize *user_data to
* NULL or to a pointer to a library-defined object.
* @param error_contact_info Out. Who-to-contact message. The library can use
* this message to change the displayed who-to-contact message in case of an
* exception.
* @param exception Out. Exception object. If there is no error, code will be
* set to RTI_ACTIVATION_LIBRARY_EXCEPTION_OK. If there is an error, code
* will be set to any of the predefined errors or a user-defined error code.
*
* @return 0 if there is an error, 1 otherwise.
*/
DLDRIVERD11Export
int RTI_ActivationLibrary_initialize(
    void **user_data,
    char error_contact_info[RTI_ACTIVATION_LIBRARY_CONTACT_INFO_MESSAGE_MAX_SIZE],
    struct RTI_ActivationLibraryException *exception);

/**
* @brief Finalize the activation library.
*
* This function will be called once after unloading the activation shared
* library.
*
* @param user_data In. User data.
* @param exception Out. Exception object. If there is no error, code will be
* set to RTI_ACTIVATION_LIBRARY_EXCEPTION_OK. If there is an error, code
* will be set to any of the predefined errors or a user-defined error code.
*
* @return 0 if there is an error, 1 otherwise.
*/
DLDRIVERD11Export
int RTI_ActivationLibrary_finalize(
    void *user_data,
    struct RTI_ActivationLibraryException *exception);

```

The loading of the **rti_activation** library will be done transparently by *Connex*t if available in the environment, but only as the last mechanism for activation. If any other mechanism is provided (such as a valid license file), this library will not have any effect.

Once loaded, the **RTI_ActivationLibrary_initialize** API will only be called once per process. This provides an opportunity to initialize some state required to obtain the activation string. This state is returned in the **userData** output parameter.

The **RTI_ActivationLibrary_get_activation_string** API may be called one or more times per process whenever a valid activation string is needed. For example, for the *Connex*t Core Libraries distributed with the LM or evaluation bundles, the API is called every time a *DomainParticipant* is created.

Optionally, before your application finishes, you might want to unload the Activation Library as part of your cleanup process. For this purpose, *Connex*t provides the **NDDS_LM_unload_activation_library ()** API, which will internally both call **RTI_ActivationLibrary_finalize** and unload the **rti_activation** shared library. This function is available for the C, Traditional C++, and Modern C++ API's:

C:

```
DDS_Boolean NDDS_LM_unload_activation_library();
```

Traditional C++:

```
class NDDSLM {  
    static DDS_Boolean unload_activation_library();  
};
```

Modern C++:

```
namespace rti { namespace util { namespace activation {  
    bool unload_activation_library();  
};
```


Chapter 4 Controlling Location for RTI Workspace and Copying of Examples

By default, the `rti_workspace` directory is created here:

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

Here's how to specify a different location for `rti_workspace` and/or disable copying the examples into the workspace.

Linux and macOS systems:

- To configure the behavior for *all* users, edit `<NDDSHOME>/resource/scripts/rticommon_config.sh`.
- To configure the behavior for the *current* user (takes precedence), create `$HOME/.rti/rticommon_config.sh` as follows:

```
# Customize RTI Workspace Directory and Copy Behavior
#####
# copy_examples=false
# workspace_dir=$HOME/rti_workspace
```

Uncomment the line(s) that you want to use (remove the #).

Windows systems:

- To configure the behavior for *all* users, edit `<NDDSHOME>\resource\scripts\rticommon_config.bat`.

- To configure the behavior for the *current* user (takes precedence), create **<your home directory>\rti\rticommon_config.bat** as follows:

```
@REM Customize RTI Workspace Directory and Copy Behavior
@REM #####
@REM set copyExamples=false
@REM set "workspaceDir=C:\Users\%USERNAME%\Documents\rti_workspace"
```

Uncomment the line(s) that you want to use (delete "@REM" at the start of the line).

Chapter 5 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.

5.1 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`.

5.2 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>.r-tipkg`) 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>/re-source/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.)

Chapter 6 Uninstalling Connex

To uninstall *Connex*, run the uninstaller application.

- On Windows platforms:

If available from the Start Menu, run the uninstaller by clicking on the **Uninstall RTI Connex <package>** option. Note that this is not available from the menu on Windows 8 systems. Alternatively, use Add/Remove Programs from the Windows Control Panel to uninstall *Connex*, or run the application from the <NDDSHOME>\uninstall directory.

- On other platforms:

Run the uninstaller from the <NDDSHOME>/uninstall directory.

Note: The uninstaller does not delete the **rti_workspace** directory. To delete all RTI files, delete the **rti_workspace** directory in your home directory. (See [Chapter 2 Paths Mentioned in Documentation on page 13.](#))