RTI Launcher

User's Manual

Version 5.2.3

rti
Your systems. Working as one.

Trademarks

Real-Time Innovations, RTI, and Connext are trademarks or registered trademarks of Real-Time Innovations, Inc. All other trademarks used in this document are the property of 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 under and subject to the RTI software license agreement. The software may be used or copied only under the terms of the license agreement.

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 Launcher is a graphical application that allows you to install, configure, and run *RTI Connext™ DDS* components. It automatically detects which RTI components are installed and enables their launch buttons, providing a convenient graphical interface to the configuration and run-time parameters.

You can use *Launcher* to install new components or target platforms into the current *Connext DDS* installation, see what is already installed, navigate to the documentation and web-based resources, create new source-code projects (via its integration with *rtiddsgen*), and run all the *Connext DDS* tools and command-line utilities.

Launcher is intended to be your "window" and "management console" for all your *Connext DDS* products. Keep it handy and use it as the "launch" interface whenever you need to start any RTI tool or service.

In addition, you can customize *Launcher* to run any program installed in your system, such as third-party tools that work with *Connext DDS*, or commonly used editors. This can be easily done by changing the XML configuration file in the **resource/app/app_support/launcher/panels** subdirectory of your *Connext DDS* installation directory.

Chapter 2   Paths and Environment Variables Mentioned in Documentation

The documentation refers to:

❑ *Connext DDS* **Installation Directory, <NDDSHOME>**

These both refer to the installation directory for *Connext DDS*.

The default installation paths are:

- Mac OS X systems:

    /Applications/rti_connext_dds-*version*

- UNIX-based systems, non-*root* user:

    /home/*your user name***/rti_connext_dds-*version***

- UNIX-based systems, *root* user:

    /opt/rti_connext_dds-*version*

- Windows systems, user without Administrator privileges:

    <your home directory>\rti_connext_dds-*version***

- Windows systems, user with Administrator privileges:

    C:\Program Files\rti_connext_dds-*version* (for 64-bits machines) or
    C:\Program Files (x86)\rti_connext_dds-*version* (for 32-bit machines)

Wherever you see "*Connext DDS* Installation Directory" or <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-version\bin\rtiddsgen"
```

or if you have defined the NDDSHOME environment variable:

```
"%NDDSHOME%\bin\rtiddsgen"
```

❑ RTI Workspace directory, **rti_workspace**

The RTI Workspace is where all configuration files for the applications and example files are located. All configuration files and examples are copied here the first time you run *RTI Launcher* or any script in **<NDDSHOME>/bin**. The default path to the RTI Workspace directory is:

- Mac OS X systems:

  /Users/*your user name***/rti_workspace**

- UNIX-based systems:

  /home/*your user name***/rti_workspace**

- Windows systems:

  *your Windows documents folder***\rti_workspace**

  Note: '*your Windows documents folder*' depends on your version of Windows.
  For example, on Windows 7, the folder is **C:\Users\your user name\Documents**;
  on Windows Server 2003, the folder is **C:\Documents and Settings\your user name\Documents**.

You can specify a different location for the **rti_workspace** directory. See the *RTI Connext DDS Core Libraries Getting Started Guide* for instructions.

❏ **<path to examples>**

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

By default, the examples are copied here:

- Mac OS X systems:

  /Users/*your user name***/rti_workspace/***version***/examples**

- UNIX-based systems:

  /home/*your user name***/rti_workspace/***version***/examples**

- Windows systems:

  *your Windows documents folder***\rti_workspace***version***\examples**

  Note: '*your Windows documents folder*' is described above.

You can specify that you do not want the examples copied to the workspace. See the *RTI Connext DDS Core Libraries Getting Started Guide* for instructions.

❏ NDDSHOME environment variable

This variable appears in makefiles and project files and will be set to the *Connext DDS* installation path. If you use applications/commands from *RTI Launcher*, this variable will be automatically set for you. For details, see the *RTI Connext DDS Core Libraries Getting Started Guide.*

Chapter 3   Using RTI Launcher

3.1   Starting Launcher

To start *Launcher*:

❏ **On a Mac OS X system**:

If you installed *Connext DDS* in the **/Applications** directory, the LaunchPad utility should show RTI Launcher in the *Connext DDS* folder (**rti_connext_dds-*version***). Click on it to open *Launcher*.

❏ **On a UNIX-based system**:

You must have a graphical environment (such as X11, GNOME or KDE) running.

- Open a command prompt and navigate to the *Connext DDS* Installation Directory. Run the script **bin/rtilauncher**.

- Alternatively, use your browser to navigate to your *Connext DDS* Installation Directory's **bin** directory and double-click **rtilauncher.**

❏ **On a Windows system**:

There are three ways to run RTI *Launcher* on a Windows system:

- In the Start menu under **All Programs**, find the entry called *RTI Connext DDS <version>*. Underneath there you will see **RTI Launcher *<version>***. Select it to start *Launcher*.

- Use Windows Explorer to navigate to your *Connext DDS* Installation Directory's **bin** directory. For example, **C:\Program Files\rti_connext_dds-*version*\bin.** Double-click **rtilauncher.bat**.

- From a command prompt, navigate to your *Connext DDS* Installation Directory/ bin and double-click **rtilauncher.bat**.

3.2   License Management

Launcher itself does not require a license file. However, some of the *Connext DDS* components may require a license file. *Launcher* will aid you in making sure a license file can be found by

these applications when started from *Launcher. Launcher* will look for the license file named **rti_license.dat** in three locations in the order specified below:

❏ The path indicated by the **RTI_LICENSE_FILE** environment variable. The file does *not* need to be called **rti_license.dat** to use this method.

❏ The **RTI user home**. This is the **rti_workspace/***version* directory in the user's home folder. The specific location depends on the operating system.

❏ The *Connext DDS* **Installation directory**. This is top-level directory where you installed the *Connext DDS* bundle.

For example, if a license-managed version of *Connext DDS* is installed in **C:\Program Files\rti_connext_dds-***version* on a Windows system and there is no RTI_LICENSE_FILE environment variable set, *Launcher* will first look for **C:\Users***your user name***\Documents\rti_workspace***version***\rti_license.dat** *(RTI user home)*. Then, if not found, it will look for **C:\Program Files\ rti_connext_dds-***version***\rti_license.dat.**

3.2.1 Setting the License File

Some of *Launcher*'s tools may be license-managed. In this case, if *Launcher* has no knowledge of a valid license file, those tools will be grayed out and won't work until a valid license is provided. A warning message will be shown to indicate that the license file can not be found.

When you have a valid license, there are a few ways to tell *Launcher* about the license file:

❏ Use the *Launcher* Installation tab's **Choose** button. This will open a file chooser dialog that will let you navigate to the license file. After choosing the file, *Launcher* will prompt you with the following dialog:
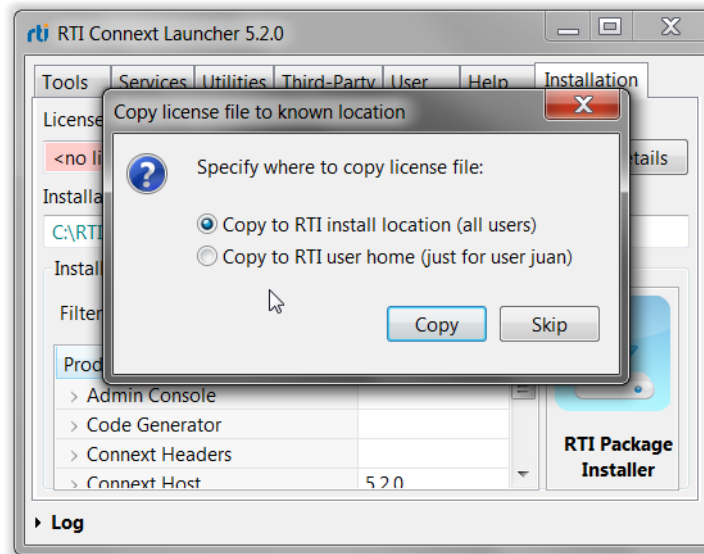
You have three choices:

- **Copy to RTI install location (all users):** The chosen license file will be copied to the *Connext DDS* Installation Directory, with the name **rti_license.dat**. *Launcher* will use this license file not only for the current execution and the current user, but for subsequent executions by any user in the system with access to the *Connext DDS* installation. Note that if you have installed into a system directory, selecting this option may require system privileges.

- **Copy to RTI user home (just for user <user>)**: The chosen license file will be copied to the current user's **rti_workspace/<*version*>** directory. The license file will thus only be visible for the current user in subsequent executions of *Launcher*.

- **Skip**: The chosen license file will be applied by *Launcher* only for the current execution of the tool. It will not be automatically detected by the tool in subsequent executions.

❏ Copy the file manually to one of the aforementioned locations where *Launcher* will search by default, and name the file **rti_license.dat**. As mentioned before, if the RTI license file is copied to the RTI user home, it will only be visible for this user in subsequent executions of the tool. When the file is copied to the RTI install location directly, it will be visible for all users with access to that directory.

The file may be copied both when *Launcher* has already been started or when it is offline. If the license file is copied while *Launcher* is executing, *Launcher* will detect the change automatically and apply the license to the UI.

❏ Set the **RTI_LICENSE_FILE** environment variable with the path to the license file. When setting this variable, not only *Launcher* but all the RTI applications will use the license file environment variable when executing.

Notice that when copied to one of the known locations, all RTI applications will also detect the license file.

After successfully setting a license file with *Launcher*, all the tools that were disabled because of the license will become available and usable.

3.3 Launcher's Tabs

Launcher has a Graphical User Interface with tabs for different application categories. The contents of the tabs are XML-defined. If a tab contains no entries (buttons), or none of the applications in a tab are supported on your platform, that tab will not appear in *Launcher*.

There are five tabs that are always visible to the user:

❏ Tools Tab (Section 3.3.1)

❏ Services Tab (Section 3.3.2)

❏ Utilities Tab (Section 3.3.3)

❏ Installation Tab (Section 3.3.4)

❏ Help Tab (Section 3.3.5)

Launcher also includes other tabs, which may or may not be visible.

❏ Labs Tab (Section 3.3.6)

❏ Third-Party Tab (Section 3.3.7)

❏ User Tab (Section 3.3.8)

To start one of the applications in a tab, click on the application's button. Some applications will launch directly, such as *RTI Admin Console.* Other applications, like the ones in the Services or Utilities tab, will pop-up an application-specific dialog. This dialog allows you to configure the settings to run the application and then execute it.

Right- clicking (or Command-Click on Mac OS) a button will open a context menu with additional information and options:

The context menu may vary for every tool, but in general, the options are:

❑ **View command history:** *Launcher* keeps track of all the successful command executions in the current *Launcher* session. When you click this menu item, it will open up the application's Command History dialog, which shows the command that was executed, when it was launched, and the working directory from where it was launched:
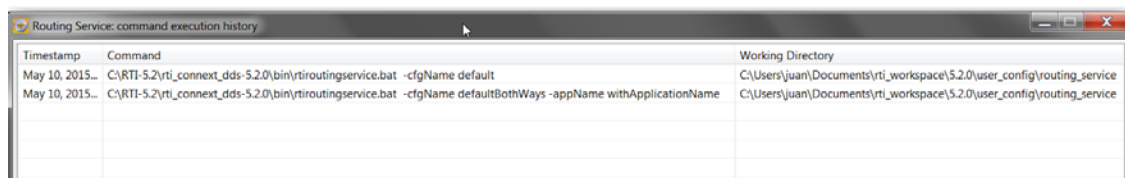
❑ **Command Help:** Some applications provide usage notes regarding calling parameters (services, utilities, etc.). For those applications, this menu will show the available options.
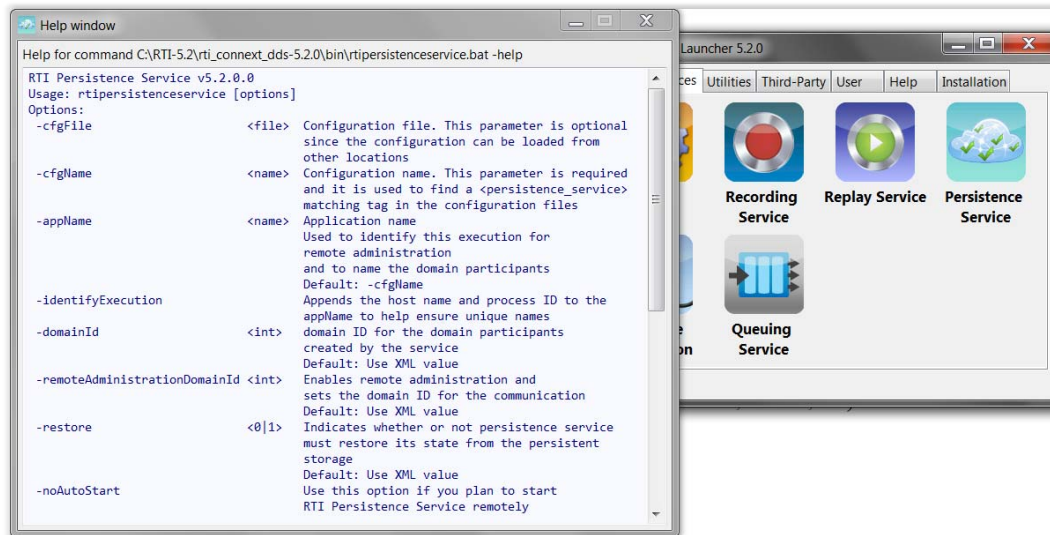
For example:

❏ **Document links:** Every application provides its own set of documentation or information links; these are shown in the context menu. They may be local PDF files, online PDF files, online HTML links, etc.

If a tool is not installed or cannot be used (due to an expired license, for instance), its button will appear grayed out and will not work.

Launcher automatically detects the presence of the applications in various ways. In the majority of cases, *Launcher* looks for installed applications in the *Connext DDS* Installation Directory's **bin** directory. For other applications, it looks for in the Windows Registry. On Linux systems, Wireshark is found by using the UNIX *which* command.

Not all tools are available on all platforms. For example, the RTI Spreadsheet Add-in for Microsoft Excel is only available on Windows systems.

3.3.1 Tools Tab

The tools tab contains applications that can be run directly, without requiring any additional application-specific setup. The Tools tab includes the following applications:

❏ **Admin Console** provides a centralized tool for administering, supervising, and visualizing your distributed system. It provides control over RTI services such as *RTI Routing Service* and *RTI Recording Service*, offers a centralized way to receive and view log messages, detects configuration errors including type and QoS mismatches, and even data visualization capabilities: you can subscribe to the topics in the system to see the data being published in real time.

❏ **Monitor** provides a graphical view to the internal configuration and operation of *Connext DDS* applications, including QoS and detailed statistics on connections, traffic, errors, and resource usage.

❏ **Recording Console** is a graphical front-end for the Record and Replay services. It can record real-time data without having prior knowledge of the data-types or topics in the system; it can also replay that data back into the DDS cloud.

❏ **Shapes Demo** demonstrates *Connext DDS* capabilities by publishing and subscribing to colored moving shapes.

❏ **Wireshark** captures RTPS packets, shows their contents, and analyzes network usage.

❏ **Spreadsheet Add-in for Microsoft Excel®** integrates *Connext DDS* with Microsoft Excel. You can use Excel cell functions to subscribe to topic data and then use the data in real-time for formulas, graphs and other Excel features. You can also publish data from a spreadsheet into the DDS cloud.

3.3.2 Services Tab

The Services tab provides buttons and dialogs to run RTI infrastructure services that require application-specific setup. The Services tab includes the following applications:

❏ **Routing Service** integrates disparate and geographically dispersed systems. It scales *Connext DDS* applications across domains, LANs and WANs, including firewall and NAT traversal. It also supports *Connext DDS*-to-*Connext DDS* bridging by allowing you to make transformations in the data along the way.

❏ **Recording Service** records real-time data without having prior knowledge of the data-types or topics in the system. (This service is also available by using Recording Console in the Tools tab.)

❏ **Replay Service** replays recorded data by injecting it back into DDS. You can change data rates and QoS settings. (This service is also available by using Recording Console in the Tools tab.)

❏ **Persistence Service** saves data from *Connext DDS* publishing applications to memory or permanent storage, so it can be delivered to subscribing applications that join the system at a later time—even if the publishing application has already terminated. Data can be persisted to files or an external database.

❏ **Database Integration Service** provides bidirectional integration between *Connext DDS* and a database (Oracle Database, Oracle TimesTen In-Memory Database, or MySQL).

❏ **Queuing Service** provides a queue (exactly-once) communication pattern in which each sample will be delivered to exactly one QueueConsumer. If there are no QueueConsumers available when the sample is sent, the sample will be stored until a QueueConsumer is available to process the sample. This pattern is very useful in load-balancing applications.

The configuration dialog is similar for all the services. For example, the configuration dialog for *Routing Service* looks like this:
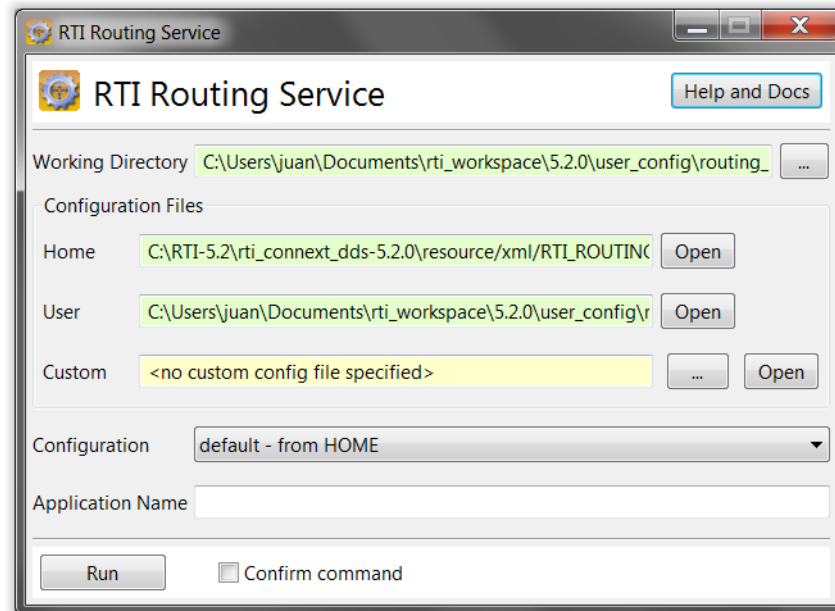

These dialogs also include validation for wrong parameters or paths. Text fields are highlighted with different colors depending on their validation status. When a text field is highlighted with a color: green indicates that the field has a valid entry; yellow indicates it is missing but not required; red means it contains invalid values.

When you are done configuring the service, click on the **Run** button to run the service. If the dialog has some invalid entries, the **Run** button will be disabled and a warning icon will appear next to it.
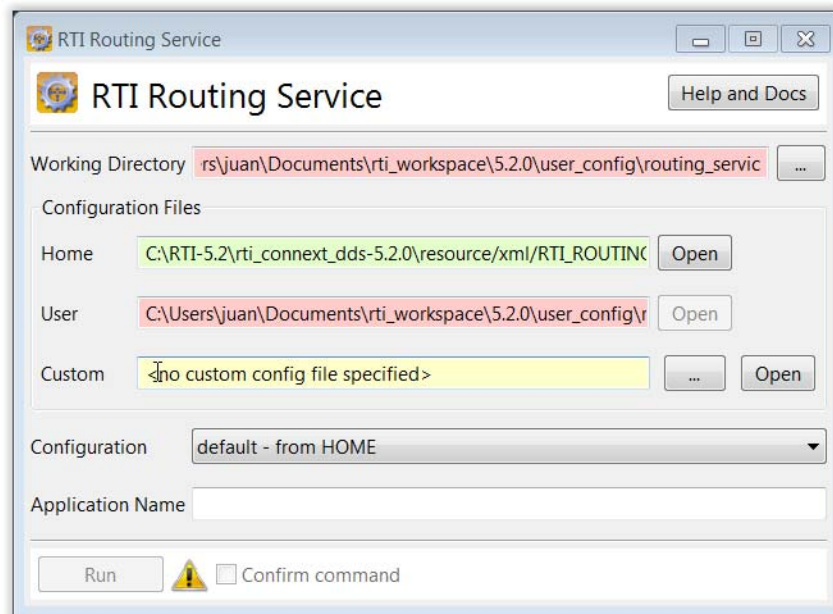
The warning icon's pop-up will show you the why the dialog is invalid.

If you want to preview the command that will be used for execution, check the **Confirm command** checkbox. If the checkbox is checked when clicking on **Run**, a pop-up dialog will show the command and command-line parameters that will be used and the working directory from which the command will be run:
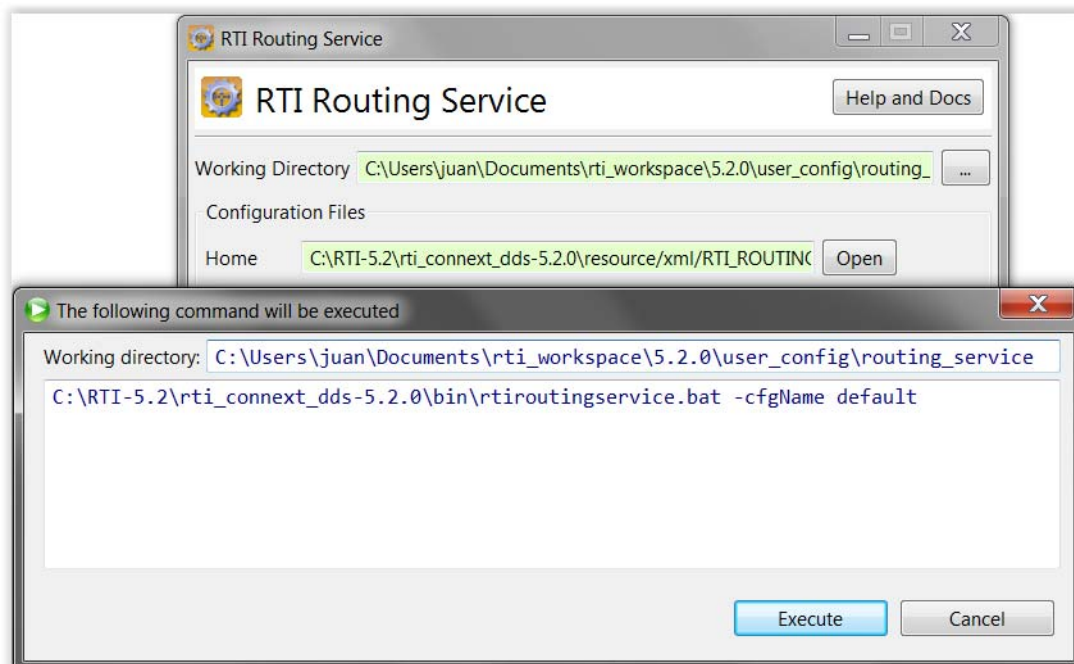
If the execution is successful, this command will be stored in the Command History of the service. See Section 3.3 for more about viewing command history.

3.3.3 Utilities Tab

The Utilities tab contains helper applications provided with *Connext DDS* to assist you when testing, designing, or inspecting your DDS system. It also provides a Package Installer utility, which makes it easy to install target platform libraries, or additional components not included in the original bundle.
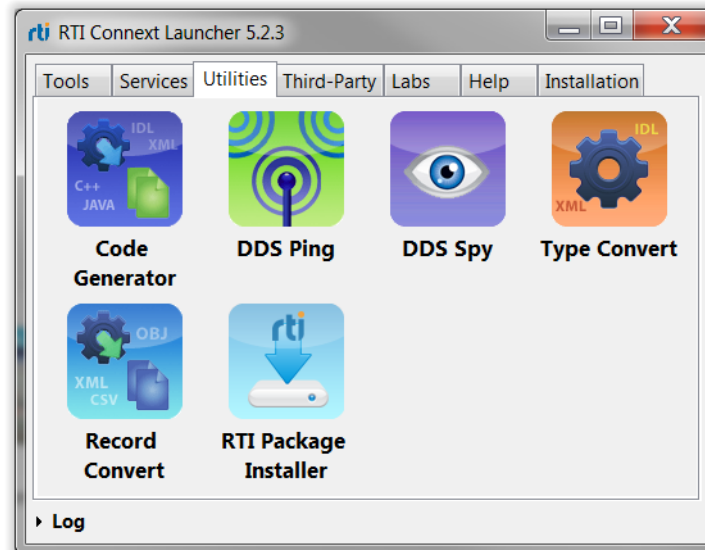
❑ **Code Generator**, also known as *rtiddsgen,* generates the code to serialize and deserialize application data on the wire from an IDL (or XML) description of the data-type. It also generates example code and makefiles/project files so you can quickly build a working application.

❑ **DDS Ping**, also known as *rtiddsping*, publishes (and subscribes to) DDS ping messages to test system connectivity.

❑ **DDS Spy**, also known as *rtiddsspy*, can subscribe to any DDS Topic and print out all the data being published.

❑ **Type Convert** converts data-type formats between IDL and XML.

❑ **Record Convert** converts database files recorded by Recorder into readable, commonly accepted formats such as CSV, XML, and HTML.

❑ **RTI Package Installer** installs additional components and libraries into an existing *Connext DDS* installation. This utility accepts RTIPKG (**.rtipkg** extension) files as its inputs.

When you click on a utility button, a configuration dialog will pop up. Every utility has a different dialog, customized for its own needs. For example, the Code Generator dialog looks like this:
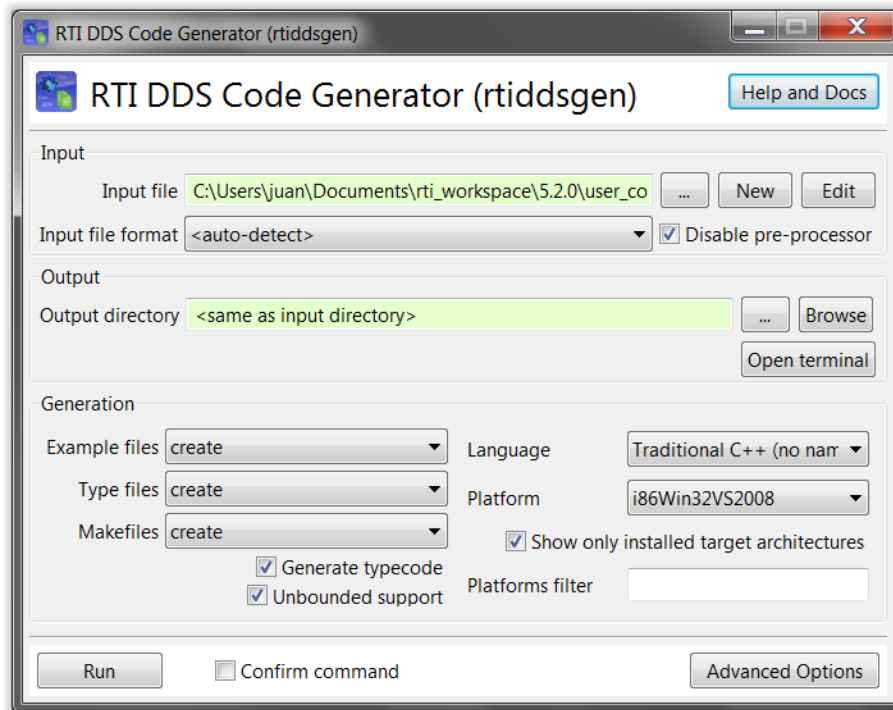

3.3.4 Installation Tab

The Installation tab provides general information regarding the current *Connext DDS* installation. This tab:

❑ Displays the current installation directory for *Connext DDS*.

❑ Displays and allows changes to the license file. License may not be necessary depending on the bundle installed. *Launcher* detects the need for a license file on a per-product basis.

❑ Displays a table with the installed products and libraries, and their installed versions. This table can be filtered by using a 'Filter' text field. The table can also be fully expanded/collapsed by using the two buttons next to the Filter field.

❏ Provides another way to access the RTI Package Installer to install additional components and libraries into an existing *Connext DDS* installation.
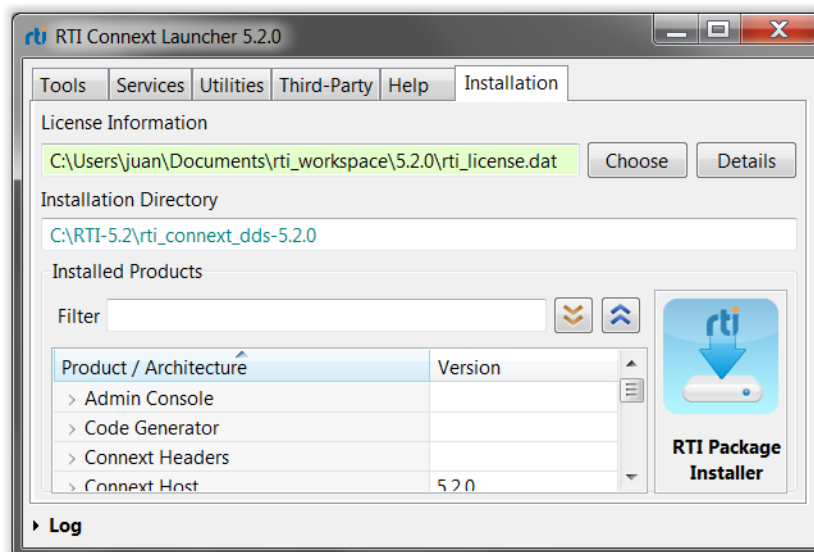
3.3.5 Help Tab

This tab provides relevant online and local (installed) links to documentation, information, e.g. the RTI Support Portal, the RTI Community web-page, etc.
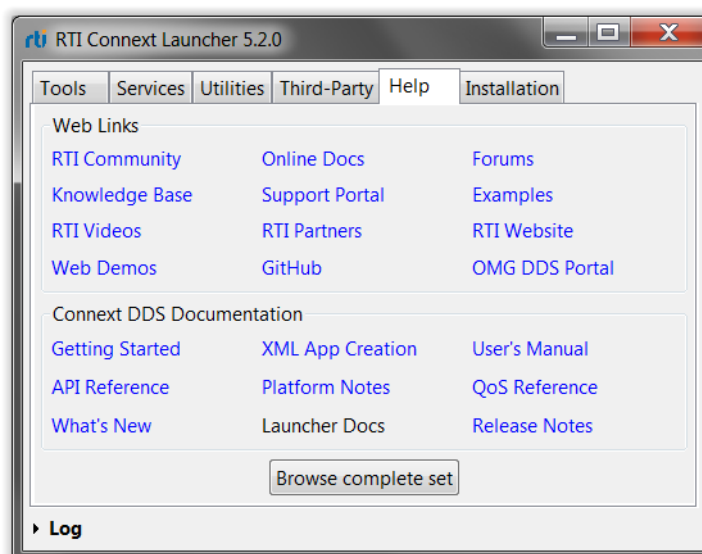
The **Browse complete set** button opens the root of the documentation folder in the *Connext DDS* Installation Directory.

3.3.6 Labs Tab

This tab is reserved for RTI applications that are experimental. It may be empty in some releases and thus not shown.

The XML file that defines this tab is in the RTI workspace in the user's home folder: **rti_workspace/***version***/user_config/launcher/panels/labs.xml**.

You can find experimental applications on our Community Portal (https://community.rti.com). Visit it and download the new applications. Most of them will also provide XML snippets to be added to *Launcher* configuration file that will allow you to run the new application as other *Connext DDS* ones.

3.3.7 Third-Party Tab

This tab is reserved for applications from RTI's partners. For information about RTI's strategic partners, see the RTI Partners website: https://www.rti.com/partners.

Products downloaded and installed by using the Third-Party tab in RTI *Launcher* are not part of RTI and you should contact the third-party provider for more information about them.
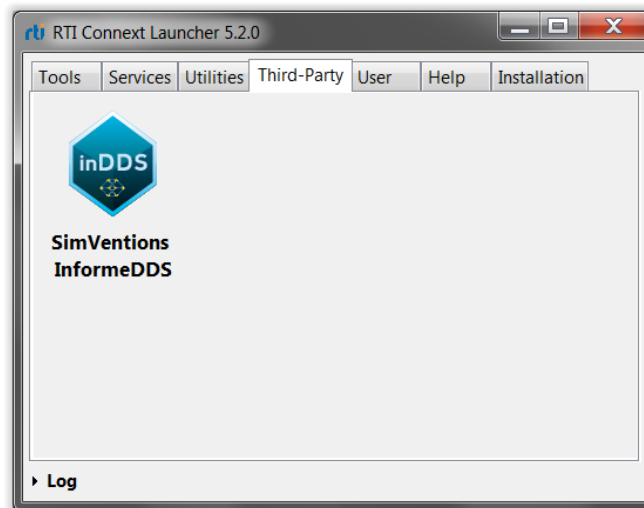
3.3.8 User Tab

The User tab is reserved for customizing *Launcher*. Its purpose is to allow users to integrate their preferred applications in *Launcher*. The User tab is not visible the first time Launcher is run, because it's empty by default, and Launcher doesn't show tabs that are empty.
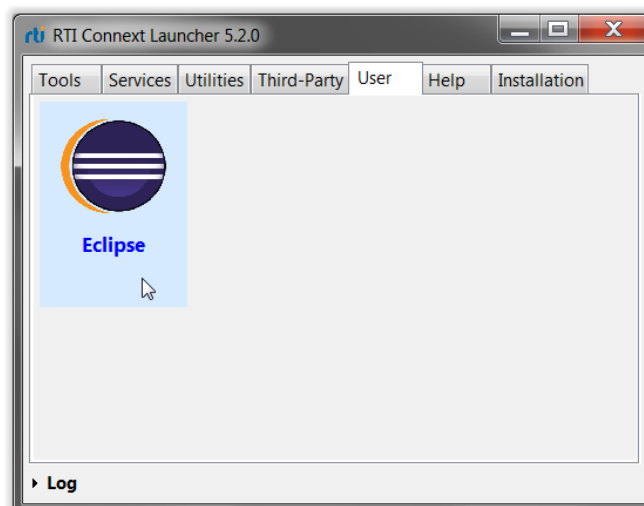
The XML file that defines this tab is in the RTI workspace in the user's home folder: **rti_workspace/*version*/user_config/launcher/panels/user.xml.**

You can modify this XML file to add shortcuts for any application in the system. See Chapter 4: Configuring Launcher for more details on how to do this.

3.4   Installing Target Libraries and RTI Components

Launcher uses the RTI Package Installer to install new libraries and components. The Package Installer accepts files in RTIPKG format (***.rtipkg**). RTI's target libraries and add-on components are available in this RTIPKG format when downloaded from the RTI download portal.

The RTI Package Installer executable is in the *Connext DDS* Installation Directory, in the **bin** directory. It can also be executed from the command-line prompt.

In *Launcher,* the RTI Package Installer can be found in both the Utilities tab and the Installation tab. When you click it, a dialog will pop up, like this:
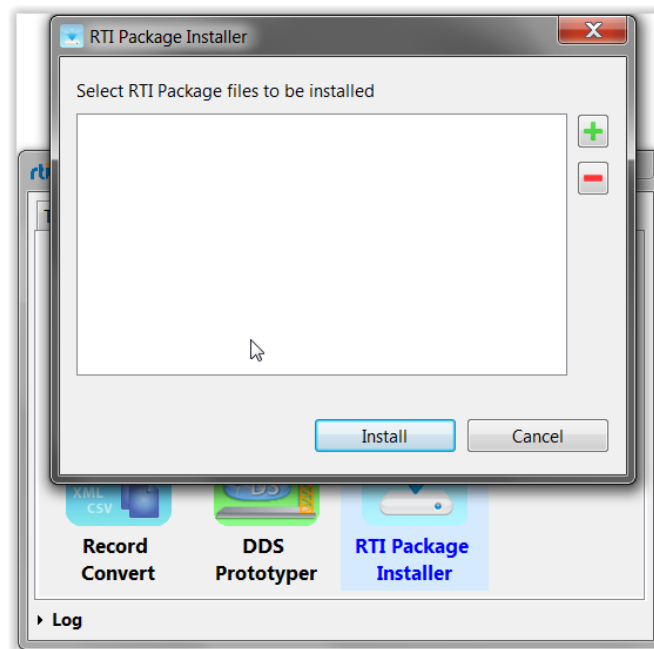
At this point, select the RTIPKG files to be installed. If you click on the '+' button, a file chooser dialog will appear, allowing you to navigate to the file(s) you want to install. You can also drag

and drop RTIPKG files into this window. Once the files are selected, click the Install button. The install commands are launched sequentially. A progress dialog will appear:
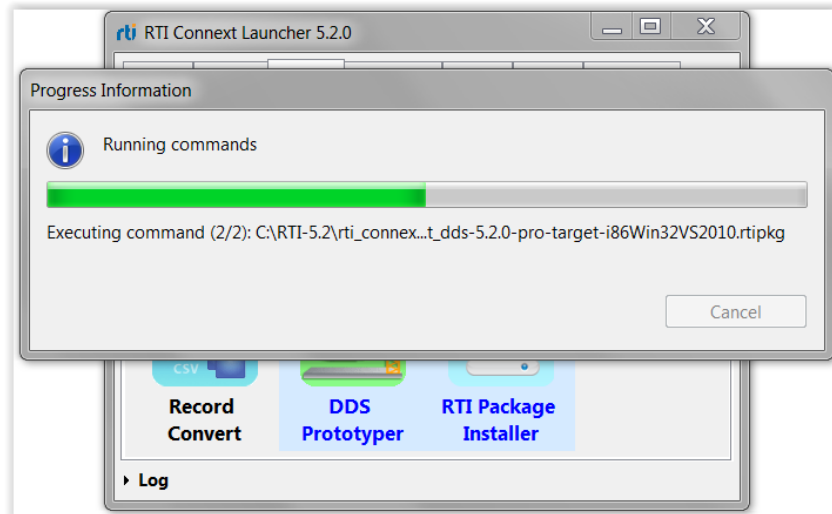
3.5   Troubleshooting with Launcher

Launcher provides two debugging or troubleshooting mechanisms:

❏ **Log view**: *Launcher* produces log messages when it performs an action, an error occurs, etc. You can access these log messages from any of the tabs by clicking on the expandable 'Log' view. When clicked, the view will expand and show all the messages produced up to that point:
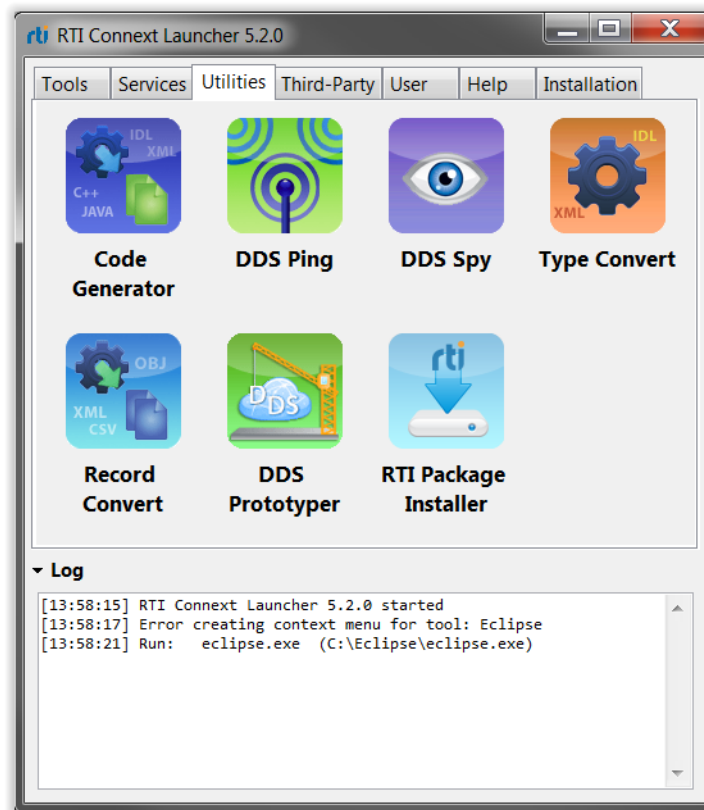
❏ **Log files**: *Launcher* logs output and errors to files in a special directory. This directory can be found in the user home's RTI workspace directory: **rti_workspace/<version>/ user_config/launcher/logs**. *Launcher*'s execution log files are named with the following pattern: **launcher-log-<timestamp>.txt.** Every execution of *Launcher* creates a different log file.

For applications not launched in command-prompt windows—like the applications in the Tools tab—*Launcher* also logs their output into text files for application troubleshooting. These files are stored in the same place as the *Launcher* logs (**rti_workspace/***version***/user_config/launcher/ logs**). They are named following this pattern: *log-<executable name>-<timestamp>.out*.

Chapter 4   Configuring Launcher

4.1   Launcher Tabs and Their Associated XML Files

Launcher's tabs are defined using XML. There are two kinds of XML configuration files for *Launcher*: predefined ones and user-customizable ones.

You should not modify the predefined XML files. These XML files are in the *Connext DDS* Installation Directory, in **<NDDSHOME>/resource/app/app_support/launcher/panels**. These files configure the following tabs:

- ❏ Tools tab: **tools.xml**
- ❏ Services tab: **services.xml**
- ❏ Utilities tab: **utilities.xml**
- ❏ Third-Party tab: **third-party.xml**

There are other *Launcher* configuration files that are intended to be customized by users. You will find these files in your RTI workspace[1]: **rti_workspace/<*version*>/user_config/launcher/panels** directory. You can use these files to configure the following tabs:

- ❏ Labs tab: **labs.xml**
- ❏ User tab: **user.xml**

4.2   How Launcher Finds Images

*Launcher* looks for the image files to be shown for the applications in the following places:

- ❏ In the *Connext DDS* Installation Directory, in **<NDDSHOME>/resource/app/app_support/launcher/icons**. This folder is used for RTI-related applications and applications by RTI's partners (shown in the Third-Party tab).

- ❏ In your RTI workspace directory, under **rti_workspace/*version*/user_config/launcher/icons**. This folder may not exist after installing and first running the tool. To add user-related images, you have to create this folder and add your icon image files there.

- ❏ Absolute paths. Launcher can detect files described as an absolute path to the image.

---

1. See Chapter 2: Paths and Environment Variables Mentioned in Documentation

4.3   Example: Adding a Button to Launcher's User Panel

The User tab is not visible the first time Launcher is run, because it's empty by default, and Launcher doesn't show tabs that are empty. This tab is intended to be configured by the user.

This section shows how to add a button to the User tab's XML file so that appears in *Launcher*. We'll use the Eclipse development environment as an example:
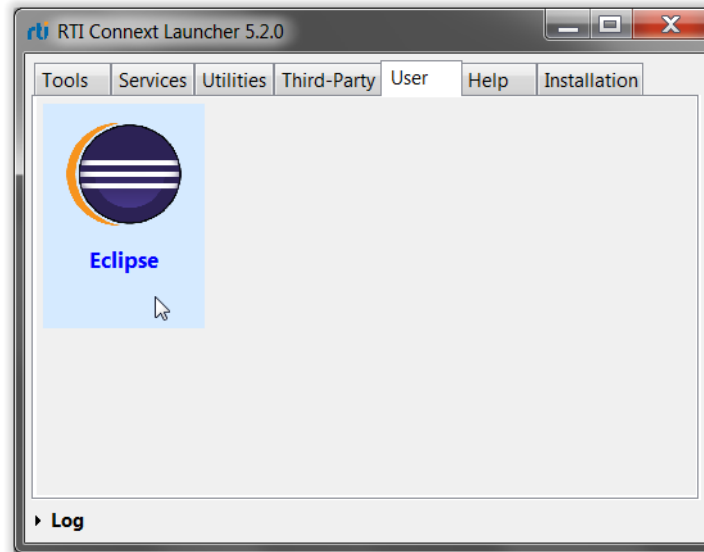
The User tab is defined in the user.xml file found in the Launcher configuration folder in the user home's RTI workspace. When this file is opened for the first time, it will contain the following:

```
<launcher xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="launcher_panel.xsd">
    <!-- Add custom menu items here -->

</launcher>
```

All the tab/panel configuration files start with the XML tag <launcher>. Let's add a button entry for running Eclipse to the user panel definition. Buttons are defined in a tag called <button>:

```
<launcher xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="launcher_panel.xsd">
    <!-- Add custom menu items here -->

    <button name="eclipse">
        <icon>eclipse256.png</icon>
        <tooltip>Eclipse:
            Eclipse Development Environment.
        </tooltip>
        <legend>Eclipse</legend>
        <toolexec>C:\Eclipse\eclipse.exe</toolexec>
        <popupmenu>
        </popupmenu>
    </button>
</launcher>
```

This is how the different tags are interpreted by *Launcher*:

❏ **'name' attribute**: Internal name of the button. No specific rule for it.

❏ **icon**: Path or name of the icon file that will be used to show the application in the panel. Preferred image file formats for Launcher are PNG and GIF (Launcher may not be able to correctly render the file if other format is provided, e.g. vectorial). It can contain an absolute path to the file, or it can refer to the name of the file, in which case the image file will be looked for in one of the two well-known image locations (see section 1.1 in this chapter). In our case, we copied the eclipse 256 logo file (eclipse256.png) into *rti_workspace/<version>/user_config/launcher/icons*. If the icons sub-directory doesn't exist, it can just be created normally.

❏ **tool-tip**: Required field. It is used to define a title for the application and a brief description. This information will be shown to the user as a balloon tool-tip when hovering over the icon. The format of this field must be: **{Title}:{Description}** (the ':' is necessary; *Launcher* may fail to create the panel if not found).

❏ **legend**: Title for the tool that will be shown right below the icon.

❏ **toolexec**: Path to the executable to run when the user clicks on the application icon.

❏ **popupmenu**: Required field, needs to be present, but it may be empty, like in our example. This tag is used to describe entries in the context menu that will be shown when the user right-clicks on the icon.

Once the **user.xml** file has been modified and saved, restart or start *Launcher* for the changes to take effect. You should be able to see the User tab now, and in it there will be a launch button for the Eclipse Development Environment.
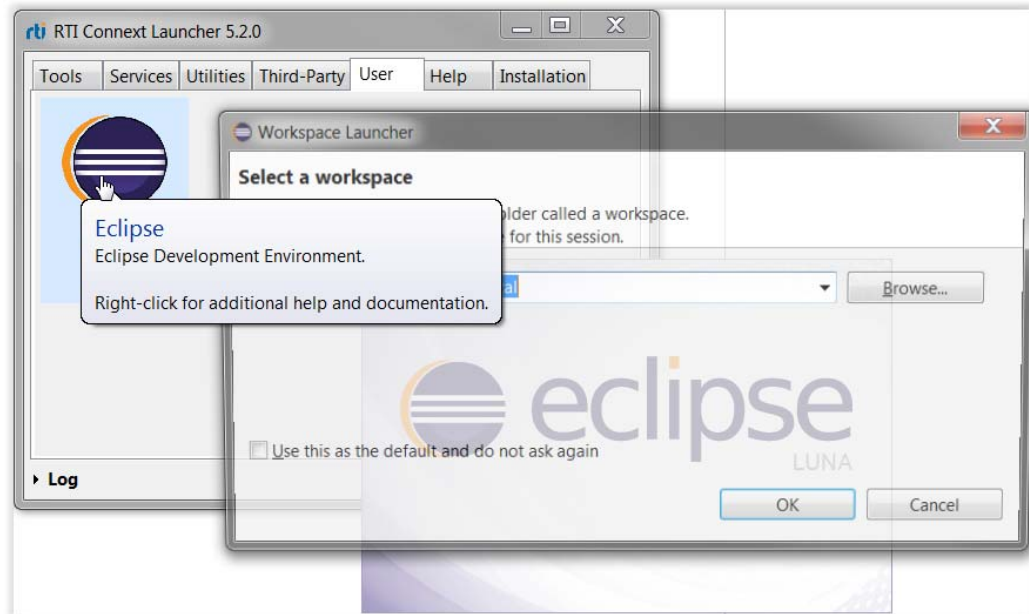
Launcher's XML format for panels is described in the XSD file **launcher_panel.xsd**. This file is located in the same directory as the **user.xml** and **labs.xml** files (**rti_workspace/*version*/user_config/launcher/panels**). This file provides complete reference for defining buttons for *Launcher* in XML.