HCL Software

HCL DevOps Model RealTime

DevOps Model RealTime 12.0

Installation Instructions

Table of Contents

1 Introduction

This document guides you through the installation of DevOps Model RealTime. It covers three different installation scenarios:

- An interactively performed local installation in section 2
- A command line based installation suitable for scripting in section 3
- An installation based on the Eclipse Installer in section 4

To follow the instructions in this document and successfully install Model RealTime 12.0 you will need the following:

- An installation of **Eclipse IDE for C/C++ Developers** for Eclipse 2023-06. This can be downloaded from https://www.eclipse.org/downloads/packages/release/2023-06/r. If you use the Eclipse installer it will be downloaded automatically.
- Access to a repository containing the Model RealTime update site. This can either be a locally downloaded zip file or the address of a shared repository that has been preconfigured for your usage.

  You can download the Model RealTime update site zip file from the HCL FlexNet Operations web site.
- A valid license. This can either be provided by means of a license server that contains Model RealTime licenses, or it can be a license key or file (permanent or trial) which you have obtained from HCL.
- An installation of a Java development kit (JDK) 64 bit.

  NOTE: Model RealTime 12.0 requires a Java 17 JDK, both when installing and running it. We recommend to use the Open JDK or Oracle JDK (see the System Requirements for the exact recommended versions).

  By default Eclipse 2023-06 distribution includes Open JDK version 17, but if you want to change it you could specify another JDK to use by editing the file `eclipse.ini` (found in the Eclipse installation folder). Specify the path to your JDK installation using the `-vm` argument. For example:

  ```
  -vm
  D:/jdk-17.0.5/bin
  ```

2 Installing from the Eclipse workbench

Here we describe how to install Model RealTime as an update site on top of Eclipse 2023.06 (4.28) using the Eclipse graphical user interface. The sections that follow will cover how to install it from the command line or using the Eclipse installer.

NOTE: Some of the screen shots below were generated on the Windows platform for an older version of Model RealTime, and will therefore differ slightly when installing the latest version of Model RealTime 12.0 and when installing on Linux.

2.1 Install

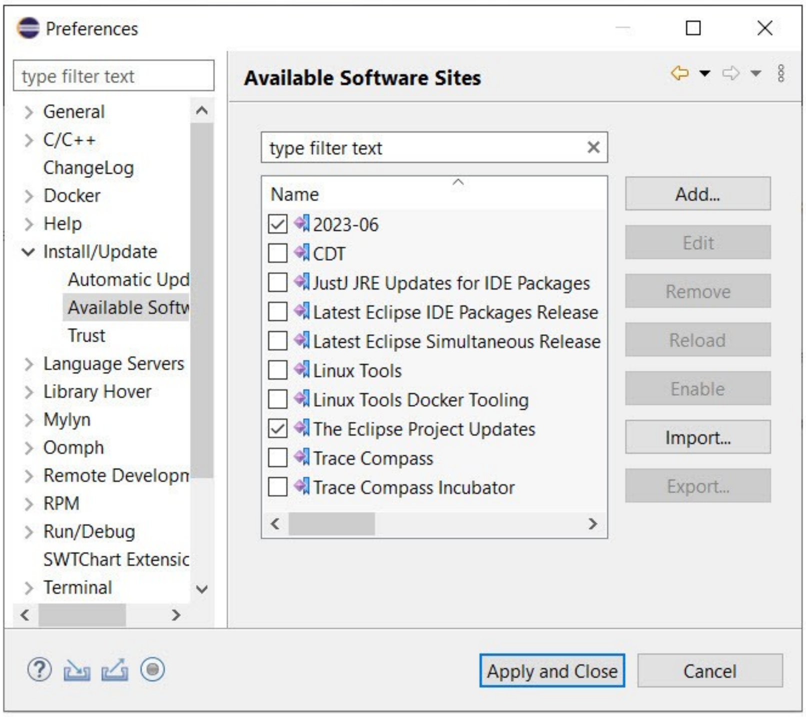
NOTE: You must have sufficient administration rights so that the OS will allow you to write the files to the installation directory of Eclipse for installing all features of Model RealTime.

If you don't have rights to write the files to the installation directory of Eclipse follow the instructions in the section on 5.1.Read-Only Installation.

NOTE: Make sure the Eclipse installation path does not contain any space characters. There are components in Model RealTime which do not work correctly if they are located in a path that contains a space.

NOTE: Model RealTime can be installed on top of many Eclipse distributions, but we recommend to use “Eclipse IDE for C/C++ Developers”. Installing it on other distributions may take a very long time since many dependent plugins need to be downloaded and installed.

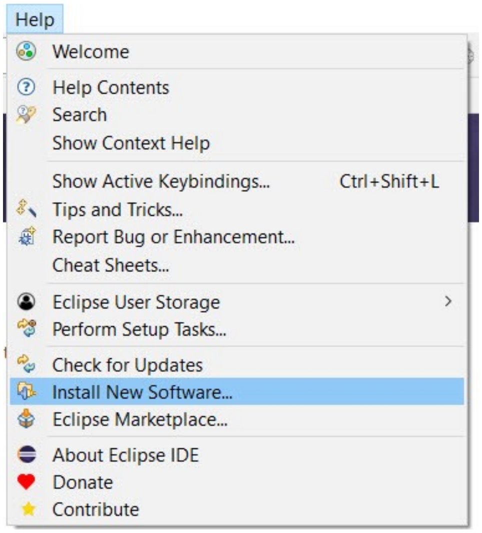
1. Start Eclipse.
2. Open **Window - Preferences** and go to the **Install/Update - Available Software Sites** preference page. Make sure to uncheck all the software sites except the following: **2023-06**, **The Eclipse Project Updates**.



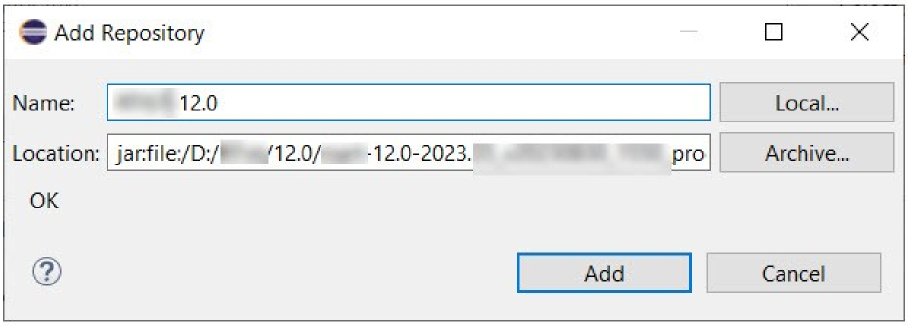
☑ 2023-06
□ CDT
□ JustJ JRE Updates for IDE Packages
□ Latest Eclipse IDE Packages Release
□ Latest Eclipse Simultaneous Release
□ Linux Tools
□ Linux Tools Docker Tooling
☑ The Eclipse Project Updates
□ Trace Compass
□ Trace Compass Incubator

NOTE: It is important to uncheck items like **CDT, Latest Eclipse Simultaneous Release, Latest Eclipse IDE Packages Release** and others that can trigger Eclipse components upgrade, because such upgrades might cause errors during Model RealTime installation.

3. Choose the **Install New Software...** command from the **Help** menu.



4. Press the **Add...** button and enter the location of the Model RealTime update site. If the update site is located on a server, you can directly type its URL in the **Location** field. If instead you have it as a ZIP file in the file system, press the **Archive...** button and select it. If the ZIP file already was unpacked to a folder, you should instead use the **Local...** button and select the folder. You can give a meaningful name to the repository in case you want to install from it again in the future. For example:


Add Repository

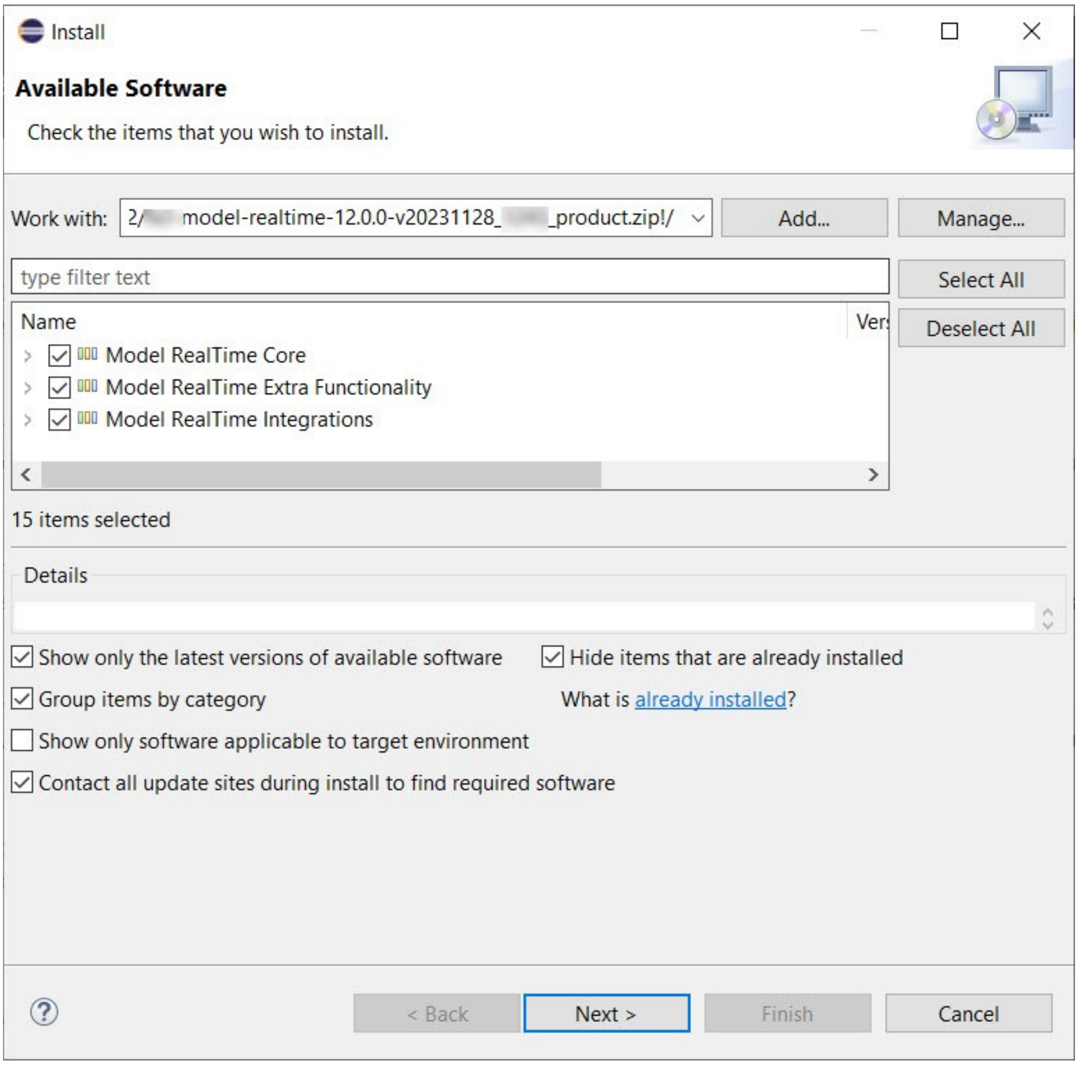
Name: [redacted] 12.0

Location: jar:file:/D:/[redacted]/12.0/[redacted]-12.0-2023.[redacted]pro

OK

Note that the Model RealTime ZIP file that you should install from ends with "product". There is also another ZIP file in what you download that contains an offline version of the documentation.

5. Select the added repository in the **Work with** drop down list. A list of Model RealTime installation categories will be shown in the list below. It will look something like this (which categories that are available depends on the exact version of Model RealTime):



Install

Available Software

Check the items that you wish to install.

Work with: 2/[redacted] model-realtime-12.0.0-v20231128_[redacted]_product.zip!/

type filter text

Name Ver

☑ Model RealTime Core

☑ Model RealTime Extra Functionality

☑ Model RealTime Integrations

15 items selected

Details

☑ Show only the latest versions of available software

☑ Hide items that are already installed

☑ Group items by category

What is already installed?

□ Show only software applicable to target environment

☑ Contact all update sites during install to find required software

Under each installation category you can see the features it contains. You can select individual features to install, but often you will select only the top-level categories to install all features they contain.

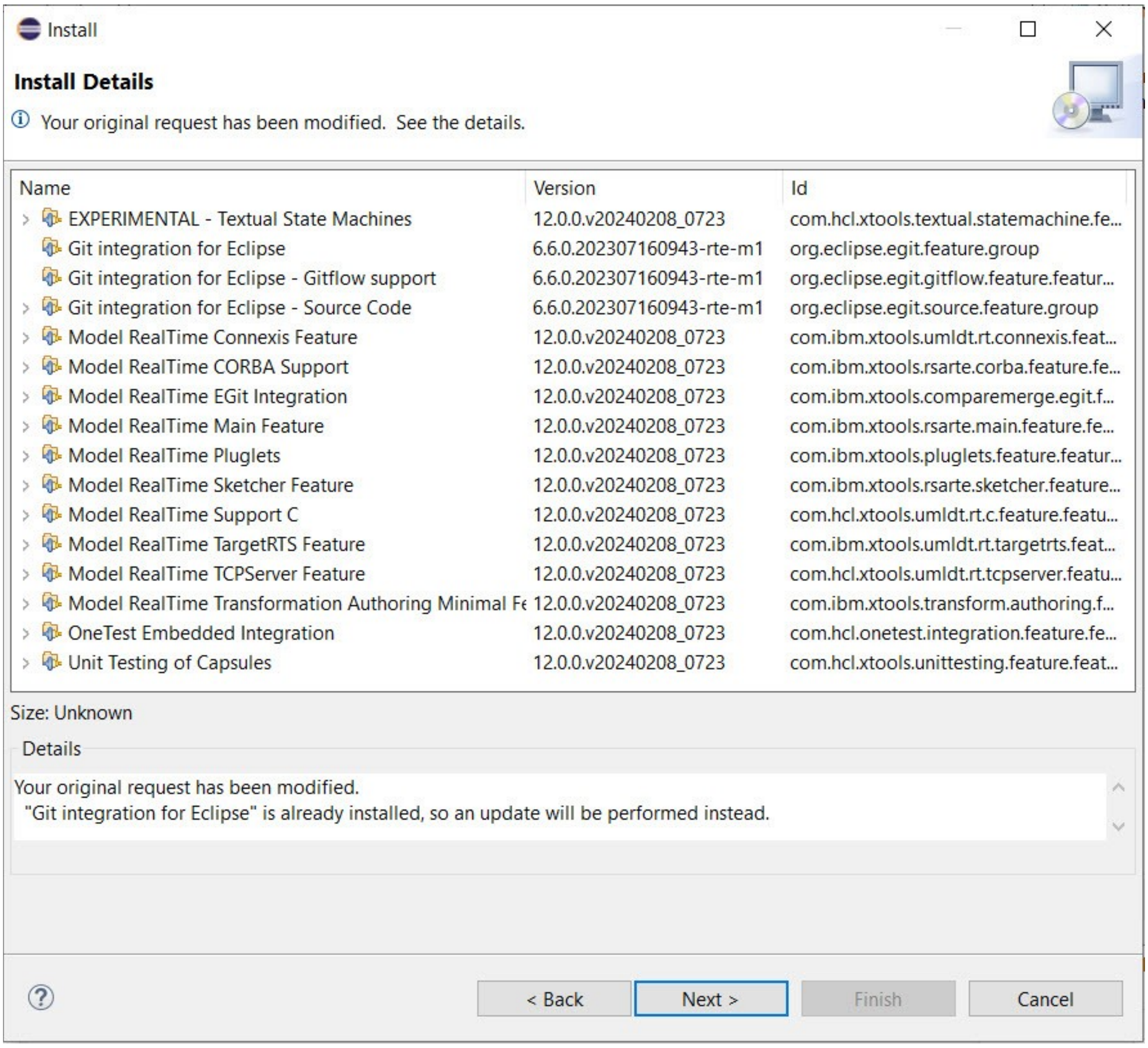
The core Model RealTime functionality is contained in the **Model RealTime Core** category. All other categories contain optional additional functionality which you may or may not choose to install depending on your needs.

6. Select **Model RealTime Core** and any of the other optional categories you want to install. Press **Next**.

- **NOTE**: If you get errors at this step double check that you run with sufficient priviliges to be able to make updates to the system and files. (On Windows you may need to start Eclipse using the “Run As Administrator” command and on Linux with root rights). Also make sure that you use a Java 17 JVM.

- **NOTE:** Depending on what Eclipse distribution you use, some Eclipse plugins which Model RealTime requires may be missing. Such plugins must be downloaded during the installation. To allow this to happen automatically you must select the checkbox **Contact all update sites during install to find required software**. As explained above, Eclipse 2023-06 (https://www.eclipse.org/downloads/packages/release/2023-06/r) must be in the list of available software sites (*Preferences – Install/Update – Available Software Sites*). If your machine does not have internet access, please follow the instructions given in the chapter Installing without Internet Access.

- **NOTE**: If you install the Model RealTime EGit Integration (under the category Model RealTime Integrations) you also must install EGit (Git Integration for Eclipse). Model RealTime has its own version of EGit which is extended compared to the one that is part of Eclipse itself. At a minimum you must install the component called "Git integration for Eclipse".
- **NOTE:** If you choose to install the Model RealTime EGit Integration into an Eclipse installation that already contains a version of EGit that is different from the version included in the Model RealTime EGit Integration, you will be informed with a page similar to the one shown below that an update of EGit feature will be performed. This will automatically uninstall the existing EGit feature, before installing the Model RealTime EGit Integration.



Install Details

Your original request has been modified. See the details.

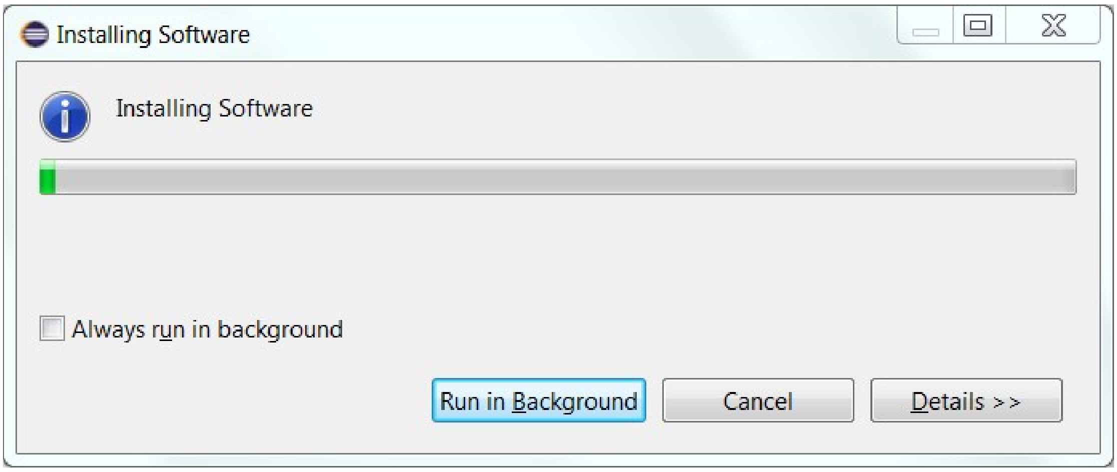
Name	Version	Id
EXPERIMENTAL - Textual State Machines	12.0.0.v20240208_0723	com.hcl.xtools.textual.statemachine.fe...
Git integration for Eclipse	6.6.0.202307160943-rte-m1	org.eclipse.egit.feature.group
Git integration for Eclipse - Gitflow support	6.6.0.202307160943-rte-m1	org.eclipse.egit.gitflow.feature.featur...
Git integration for Eclipse - Source Code	6.6.0.202307160943-rte-m1	org.eclipse.egit.source.feature.group
Model RealTime Connexis Feature	12.0.0.v20240208_0723	com.ibm.xtools.umldt.rt.connexis.feat...
Model RealTime CORBA Support	12.0.0.v20240208_0723	com.ibm.xtools.rsarte.corba.feature.fe...
Model RealTime EGit Integration	12.0.0.v20240208_0723	com.ibm.xtools.comparemerge.egit.f...
Model RealTime Main Feature	12.0.0.v20240208_0723	com.ibm.xtools.rsarte.main.feature.fe...
Model RealTime Pluglets	12.0.0.v20240208_0723	com.ibm.xtools.pluglets.feature.featur...
Model RealTime Sketcher Feature	12.0.0.v20240208_0723	com.ibm.xtools.rsarte.sketcher.feature...
Model RealTime Support C	12.0.0.v20240208_0723	com.hcl.xtools.umldt.rt.c.feature.featu...
Model RealTime TargetRTS Feature	12.0.0.v20240208_0723	com.ibm.xtools.umldt.rt.targetrts.feat...
Model RealTime TCPServer Feature	12.0.0.v20240208_0723	com.hcl.xtools.umldt.rt.tcpserver.featu...
Model RealTime Transformation Authoring Minimal Fe	12.0.0.v20240208_0723	com.ibm.xtools.transform.authoring.f...
OneTest Embedded Integration	12.0.0.v20240208_0723	com.hcl.onetest.integration.feature.fe...
Unit Testing of Capsules	12.0.0.v20240208_0723	com.hcl.xtools.unittesting.feature.feat...

Size: Unknown

Details

Your original request has been modified.
"Git integration for Eclipse" is already installed, so an update will be performed instead.

7. If you already had an older version of Model RealTime 12.0 installed, you will get an **Install Details** page with a message that “Your original request has been modified”. This is just information that Eclipse detected an old version of Model RealTime which it will update. See chapter 2.3 for more information. Just press **Next** to proceed.
8. A final **Review Licenses** page will now be shown. If you accept the license agreements, click the appropriate button and then **Finish**.
9. The installation will now start. This can take several minutes so please be patient.

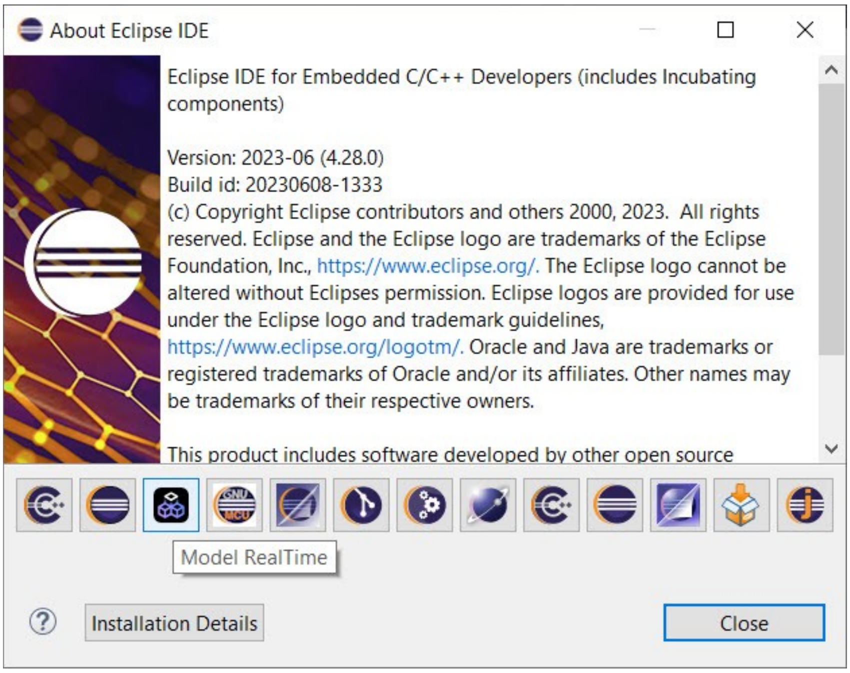


Installing Software

Installing Software

☐ Always run in background

10. After the installation is complete, you will be asked to restart Eclipse. Answer **Yes**. You are not encouraged to try to use Model RealTime without first restarting Eclipse.
    - **NOTE:** The first start-up of Model RealTime can take a long time, especially if you installed it on top of another Eclipse distribution than “Eclipse IDE for C/C++ Developers”. The reason is that all downloaded plugins must be reconciled with already existing Eclipse plugins, and this can take a significant amount of time. Subsequent start-up times will be much faster.
11. When Eclipse has started again you can confirm that Model RealTime was successfully installed. Open the **About** dialog and locate the Model RealTime icon:



About Eclipse IDE

Eclipse IDE for Embedded C/C++ Developers (includes Incubating components)

Version: 2023-06 (4.28.0)
Build id: 20230608-1333
(c) Copyright Eclipse contributors and others 2000, 2023. All rights reserved. Eclipse and the Eclipse logo are trademarks of the Eclipse Foundation, Inc., https://www.eclipse.org/. The Eclipse logo cannot be altered without Eclipses permission. Eclipse logos are provided for use under the Eclipse logo and trademark guidelines, https://www.eclipse.org/logotm/. Oracle and Java are trademarks or registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

This product includes software developed by other open source

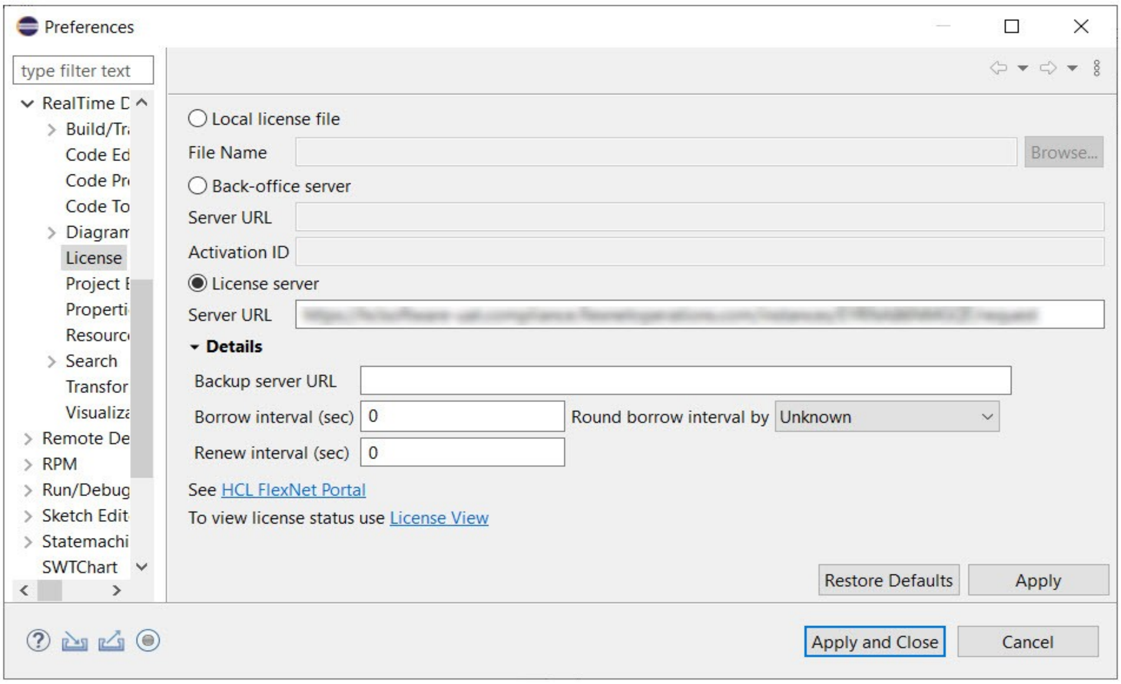
12. Click the button to view the installed version of Model RealTime and release.

    You can also find the installed Model RealTime features if you press the **Installation Details** button in the About dialog, and then click the **Installed Software** tab.

2.2 License Setup

Model RealTime can start without having a license, but you will not be able to do much before you have applied a valid license.

Use the preference page *RealTime Development – License* to apply a license.



○ Local license file

File Name

○ Back-office server

Server URL

Activation ID

● License server

Server URL [redacted]

Details

Backup server URL

Borrow interval (sec) 0 Round borrow interval by Unknown

Renew interval (sec) 0

See HCL FlexNet Portal

To view license status use License View

There are three ways to specify the license:

- **Local license file**
  If you have a local license file (usually with the extension .bin) you can specify it here.
- **Back-office server**
  If your organization has a "back-office" server at FlexNet Operations, specify the URL to the FlexNet Operations server (https://hclsoftware.flexnetoperations.com/flexnet/deviceservices). You must then also specify the Activation ID that describes your license entitlement.
- **License server**
  This could either be the URL of a locally set-up license server, or a FlexNet Operations cloud license server. The URL of the latter is https://hclsoftware.compliance.flexnetoperations.com/instances/<instID>/request where <instID> is the server instance ID.

  Note that if the license server uses a self-signed SSL-certificate, it needs to be exported to a .cert file and then imported into the truststore of the JVM running Model RealTime. Hence, it would be able to acquire the license from the server.

Under **Details**, you can specify the detailed settings for how Model RealTime should acquire and return its licenses when you use a license server. There is a field called the **Backup server URL**, which lets you specify a server link as a backup when there is an issue in getting the license from the main server. To configure the backup license server, contact the license administrator or refer to the Local license server administration guide for configuration instructions.

By default, Model RealTime attempts to acquire a license as soon as you use any tool feature that requires a license, for example when opening a model or running the model compiler. The acquired license will then be kept until the end of the day. This scheme is appropriate for a scenario when there are enough many licenses available in the license server to allow each user to have their own license which they acquire once for each day they work with Model RealTime.

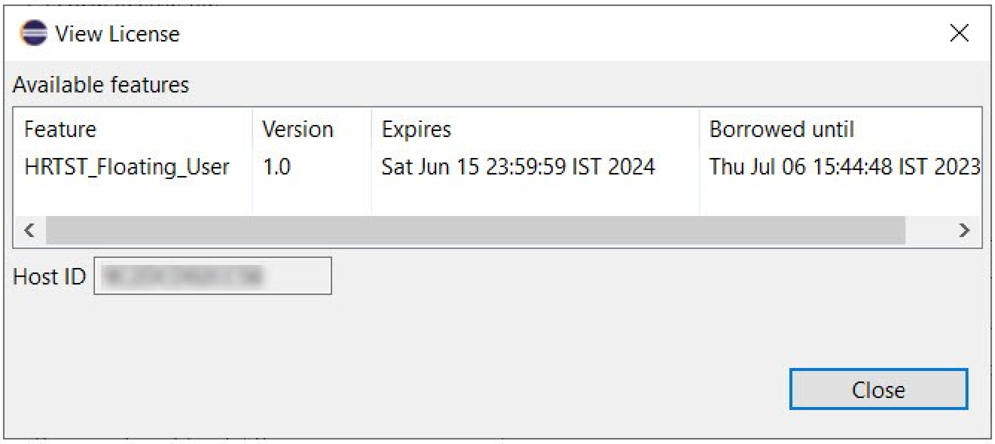
The time during which the license is kept by Model RealTime is called the **Borrow interval** (the license is "borrowed" from the license server for some time). Together with the **Round borrow interval by** it decides when it is time to return the license to the license server, so that someone else can use it. For the scenario when multiple users need to use the same license on different occasions during the same day you can set a custom borrow interval. Specify the number of seconds that the acquired license should be borrowed from the server, and set an appropriate time unit, such as Second, Minute or Hour, to round the borrow internal by. When the borrow interval has passed, at the start of the next time unit, the license will be returned to the license server, and it can then be used by someone else.

To avoid that the license is returned while you are currently working in Model RealTime and still need the license, you can specify a **Renew interval**. This is also specified in seconds and is a "heartbeat" interval at which the current license will be renewed. Therefore the renew interval should be shorter than the borrow interval. When a renew interval has been set the license is periodically renewed until you close Model RealTime. Hence you avoid the risk of losing the license while still working with the tool.

When communicating with the license server, Model RealTime (and model compiler command line builds) will by default use 5 seconds timeout for connection and 10 seconds timeout for to get a response for a license request. These timeouts could be adjusted e.g. in slow connection environments by modifying the corresponding java properties: com.hcl.rtist.license.connectTimeout and com.hcl.rtist.license.readTimeout. To change the values of these properties the following lines should be added in the end of `eclipse.ini` file

```
-Dcom.hcl.rtist.license.connectTimeout=20
-Dcom.hcl.rtist.license.readTimeout=300
```

Click the **License View** hyperlink to view the current license status.



Available features

Feature	Version	Expires	Borrowed until
HRTST_Floating_User	1.0	Sat Jun 15 23:59:59 IST 2024	Thu Jul 06 15:44:48 IST 2023

Host ID [redacted]

You can also open this dialog by means of the command **Help – View License**…

NOTE: To manage your Model RealTime licenses (and licenses for other HCL products) go to https://hclsoftware.flexnetoperations.com/flexnet/operationsportal/logon.do. You can click the hyperlink HCL FlexNet portal on the preference page to open this web page.

2.3 Update

If your Eclipse already contains a version of Model RealTime, you can update it to a later version. The steps you follow then are the same as described above. The only difference is that you will get a message telling you that some or all of the selected Model RealTime features are already installed, and that an update will be performed.

It is normally not necessary to uninstall the old version of Model RealTime before installing the new version.

If you are installing from a repository that is being updated with new releases of Model RealTime you can also update your local installation using the Eclipse command **Check for Updates** in the **Help** menu.

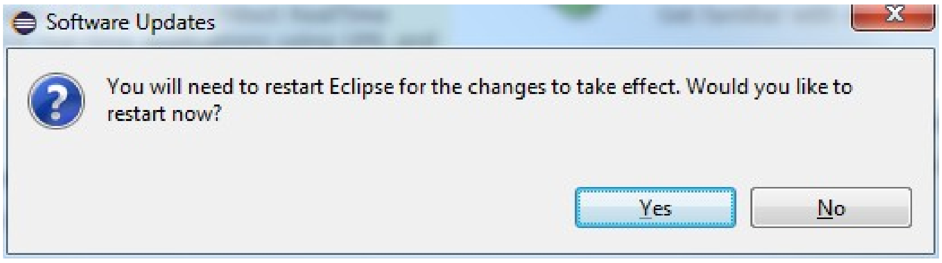
2.4 Uninstall

You can uninstall Model RealTime from the **About** dialog. Press the **Installation Details** button and then go to the **Installed Software** tab.

1. Select all lines starting with “Model RealTime”.

- **NOTE:** In case Model RealTime was installed from the command line (as described in the next chapter), you may see products grouped by several Model RealTime categories. In this case just select all top Model RealTime categories.

2. Press the **Uninstall...** button and then **Finish**.
3. After Model RealTime has been uninstalled, you will be asked to restart Eclipse. Accept it by pressing **Yes**.



Software Updates

You will need to restart Eclipse for the changes to take effect. Would you like to restart now?

3 Installing from the Command Line

The easiest way to install Model RealTime from the command-line is to use the "Install Product" script which you can find in the Utilities section on the Model RealTime Info Center web site:

Install Product

If you want to write your own script, or directly install with commands on the command-line, you will use the P2 director application which is part of Eclipse. There are two ways to invoke this application:

1. **Using Java**
   ```
   java -jar <eclipse_install_plugin_dir>/
   org.eclipse.equinox.launcher_1.6.400.v20210924-0641.jar -application org.e-
   clipse.equinox.p2.director
   ```
   For example, on Windows it may look like this (exact version number of the JAR file may differ):
   ```
   java -jar C:\eclipseCpp\eclipse\plugins\
   org.eclipse.equinox.launcher_1.6.400.v20210924-0641.jar -application org.e-
   clipse.equinox.p2.director
   ```
2. **Using Eclipse**
   ```
   eclipse -nosplash -application org.eclipse.equinox.p2.director
   ```
   On Windows, use `eclipsec.exe` instead.

In the examples below we use `<p2_director>` to refer to one of the above ways to launch the P2 director application.

NOTE: The same prerequisites regarding running with sufficient privileges apply when using the P2 director application for installing Model RealTime, as when installing from the Eclipse UI.

NOTE: When installing or uninstalling components into an Eclipse installation it's recommended not to have that Eclipse running.

The Model RealTime update site is a mandatory argument in all invocations of the P2 director application. You specify it using the `-repository` command-line argument. For example, on Windows it can look like this:

```
<p2_director> -repository jar:file:d:\downloads\hcl-model-realtime-12.0.0-
v20231211_1631_product.zip!/
```

Depending on what action you want to perform, you will use additional arguments to the P2 director application, as described in the sections below.

3.1 Install

Make sure to use an installation of Eclipse that has no Model RealTime installed. If Model RealTime is already installed you should uninstall it (see chapter 3.2) before installing a new Model RealTime version.

Also make sure that the Eclipse installation that you will install Model RealTime into is configured to not automatically update the Eclipse version. Launch Eclipse, open the **Window - Preferences - Install/Update - Available Software Sites** preference page, and make sure the **Latest Eclipse Simultaneous Release** checkbox is not checked. Then close Eclipse again before installing Model RealTime into it.[1]

Use the following command to install all Model RealTime categories.

[1] *This step can also be done from the command-line. See the section called "Disabling Latest Eclipse Release" in the "Install Product" script linked above.*

NOTE: In Model RealTime version 12.0 **category identifiers begin with version qualifier**. For example, if the product zip file name is `hcl-model-realtime-12.0.0-`**`v20231211_1631`**`_product.zip` then the category identifiers for Model RealTime will start with **`v20231211_1631`**.

```
<p2_director> -repository <model_rt_update_site> -installIU “v20231211_1631.model-realtime.core, v20231211_1631.model-realtime.extra, v20231211_1631.model-realtime.integrations, org.eclipse.egit.feature.group” -r http://download.eclipse.org/releases/2023-06
```

The category identifiers are given after the `-installIU` argument. See chapter 3.3 for how to extract the identifiers for available categories from the update site.

NOTE: The feature `org.eclipse.egit.feature.group` mentioned above refers to EGit. Normally a version of EGit is already present in Eclipse before you install Model RealTime, but if you choose to install the Model RealTime EGit Integration you should first uninstall the standard version of EGit using a command similar to:

```
<p2_director> -uninstallIU org.eclipse.egit.feature.group
```

3.2 Uninstall

NOTE: In Model RealTime version 12.0 **category identifiers begin with version qualifier**. An example of a command uninstalling all features for version **`v20231211_1631`** is:

```
<p2_director> -repository <model_rt_update_site> -uninstallIU “v20231211_1631.model-realtime.core, v20231211_1631.model-realtime.extra, v20231211_1631.model-realtime.integrations”
```

where the category identifiers are given after the `-uninstallIU` argument. See chapter 3.3 for how to extract the identifiers for available categories from the update site.

3.3 Extract identifiers for install categories

The command for extracting the identifiers of available categories in the Model RealTime update site is:

Windows:

```
<p2_director> -repository <model_rt_update_site> -list | findstr "model-realtime." | findstr /V "com.ibm.xtools." | findstr /V "com.hcl."
```

Linux:

```
<p2_director> -repository <model_rt_update_site> -list | grep "model-realtime." | grep -v "com.ibm.xtools." | grep -v "com.hcl."
```

The interesting lines in the result will look like the following (the qualifiers may be different):

```
v20231211_1631.model-realtime.core=1.0.0.827A-cJIx2BBcMAMMAeGx6GYeS
v20231211_1631.model-realtime.extra=1.0.0.9l7U-cJJXE6ED-z0Uz0z0VjmUylUkD
v20231211_1631.model-realtime.integrations=1.0.0.8_8E-cJKzU6A7QUcQgp0v6ZBbYOz-9iG
```

The category identifier is the name before the equal sign.

The update site also contains an Model RealTime specific version of EGit which has the same identifier as the standard version of EGit (`org.eclipse.egit.feature.group`).

4 Installing using the Eclipse Installer

The Eclipse Installer is a tool suitable for automating Eclipse-based installations. You can find more information about the Eclipse Installer at https://wiki.eclipse.org/Eclipse_Installer. There you also find instructions on how to download and install the Eclipse Installer.

NOTE: You must use version 1.4.0 or later of the Eclipse Installer. Installation errors will occur if you attempt to install Model RealTime with an older version of the installer.

4.1 Basic usage of the Eclipse Installer

Using the Eclipse Installer to install Model RealTime requires a setup file that specifies what should be installed. The minimal setup file for an Model RealTime 12.0 installation that installs all Model RealTime features, is the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<setup:Product
    xmi:version="2.0"
    xmlns:xmi="http://www.omg.org/XMI"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:setup="http://www.eclipse.org/oomph/setup/1.0"
    xmlns:setup.p2="http://www.eclipse.org/oomph/setup/p2/1.0"
    name="Model RealTime"
    label="Model RealTime 12.0">
  <version name="2023_06"
      label="2023_06"
      requiredJavaVersion="1.17">
    <setupTask
        xsi:type="setup.p2:P2Task"
        label="Model RealTime including Eclipse 2023-06">
      <requirement
          name="epp.package.cpp"
          versionRange="[4.28.0,4.29.0)"/>
      <requirement
          name="org.eclipse.platform.feature.group"
          versionRange="[4.28.0,4.29.0)"/>
      <requirement
          name="org.eclipse.rcp.feature.group"
          versionRange="[4.28.0,4.29.0)"/>
      <requirement
          name="org.eclipse.sdk"
          versionRange="[4.28.0,4.29.0)"/>
      <requirement name="org.eclipse.jgit.feature.group"/>
      <requirement name="org.eclipse.egit.feature.group"/>
      <requirement name="org.eclipse.jgit.pgm.feature.group"/>
      <requirement name="_VERSION_.model-realtime.core"/>
      <requirement name="_VERSION_.model-realtime.extra"/>
      <requirement name="_VERSION_.model-realtime.integrations"/>
      <repository
          url="http://download.eclipse.org/releases/2023-06"/>
      <repository
          url="url_to_modelrealtime_repository"/>
    </setupTask>
  </version>
  <description>Model RealTime Installation Script</description>
</setup:Product>
```

You can copy the text above and do the following replacements.

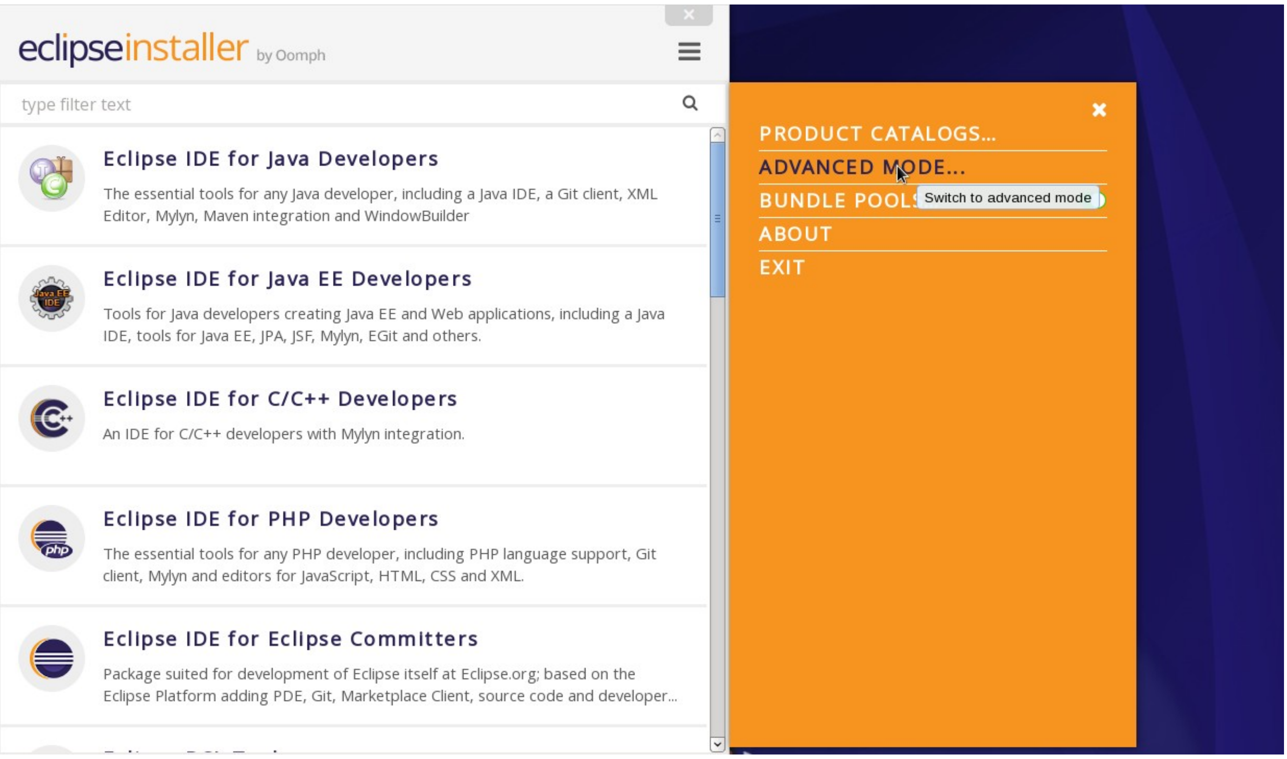
1. Replace ***url_to_modelrealtime_repository*** with your local Model RealTime 12.0 repository location and use a file with this contents as a starting point for your Eclipse installer setup script. Note that copying from PDF format may change the hyphens in the text ("-") and introduce "soft hyphens". So check that the "-" characters still remain after copying.

   Here is an example of what the repository URL for a local Model RealTime 12.0 repository (unzipped into a folder) can look like in the setup script:

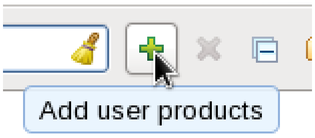
   ```
   <repository url="file:///C:/hcl-model-realtime-12.0.0-v20231207_0635_product"/>
   ```

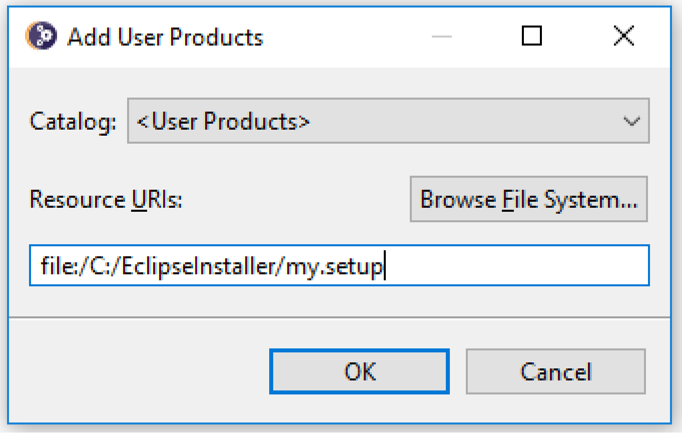
2. Replace ***_VERSION_*** with the actual version qualifier of the product you will be installing, for example v20231207_0635. See Section 3.3 on how to extract the identifiers from the zip file.

To install Model RealTime, start the Eclipse installer and switch to Advanced Mode.


eclipseinstaller by Oomph
type filter text
Eclipse IDE for Java Developers
The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Mylyn, Maven integration and WindowBuilder
Eclipse IDE for Java EE Developers
Tools for Java developers creating Java EE and Web applications, including a Java IDE, tools for Java EE, JPA, JSF, Mylyn, EGit and others.
Eclipse IDE for C/C++ Developers
An IDE for C/C++ developers with Mylyn integration.
Eclipse IDE for PHP Developers
The essential tools for any PHP developer, including PHP language support, Git client, Mylyn and editors for JavaScript, HTML, CSS and XML.
Eclipse IDE for Eclipse Committers
Package suited for development of Eclipse itself at Eclipse.org; based on the Eclipse Platform adding PDE, Git, Marketplace Client, source code and developer...
PRODUCT CATALOGS...
ADVANCED MODE...
BUNDLE POOLS
Switch to advanced mode
ABOUT
EXIT

Use the command **Add user products** to choose your setup file.





file:/C:/EclipseInstaller/my.setup

You can now install Model RealTime from the main Eclipse Installer page by selecting **Model RealTime 12.0** and pressing **Next**.

In the subsequent dialogs the default values work fine, even though you may want to modify the installation folder name. If you add additional Eclipse projects to your installation, remember to confirm that the target platform is set to 2023-06.

NOTE: Make sure the Eclipse installation path does not contain any space characters. There are components in Model RealTime which do not work correctly if they are located in a path that contains a space.

5 Shared Installations and Pre-Configured License Setup

If you would like to share one installation among a group of users, e.g. using a network installation, some special considerations concerning the license setup are needed. The main issue with this scenario is how to apply a valid license at the time of installation, to avoid that each user has to do this when they start Model RealTime.

The license information for Model RealTime is edited in the preference page *RealTime Development – License*. Contrary to most other preferences, this information is not stored in the workspace metadata but in the Model RealTime installation (more precisely in the file `<install-dir>/configuration/.settings/com.hcl.rtist.license.core.prefs`). This means that the person who creates a shared installation can enter the license information right after the installation is completed. All users of that installation will then have the license information already pre-configured and can immediately start to use Model RealTime.

5.1 Read-Only Installation

In a shared installation scenario it is common to have the installation folder to be read-only for regular users. In general this works for an Model RealTime installation but there are two constraints:

- The license information must be set-up before making the installation read-only, since it is stored within the installation folder.
- Features that provide libraries, such as the TargetRTS, Connexis and TCPServer features, cannot be installed into their usual locations in the **rsa_rt** folder if the Eclipse installation is read-only. However, the libraries are also installed into their corresponding plugins which in the case of a read-only Eclipse installation will be placed in the user configuration area. Another option could be to install these features on a different machine where the Eclipse installation is writable, and let all users access the libraries from there.

To avoid this problem and to configure the license setup for all users during the installation of Model RealTime, the following must be taken into account:

- Define a license policy file before installing Model RealTime

- Use the location of the license policy file during installation to avoid the manual license setup

In the following two sections we will look at the licence policy file, the related system properties and how to fix a pre-configured license setup for a shared installation scenario. We will also discuss some constraints when using a read only installation structure.

6 Post-Installation Setup

After installation you should do some post-installation setup tasks before starting to use Model RealTime. This is needed because the standard Eclipse settings with regards to memory etc. are usually not sufficient when working with Model RealTime.

6.1 Modifications in eclipse.ini file

Please refer to this article for information about setup tasks that are commonly needed for performance and other reasons.

NOTE: Starting from version 4.19 of Eclipse Standard Widget Toolkit (SWT) will report the following error for every non-disposed widget.

```
java.lang.Error: SWT Resource was not properly disposed
```

The errors will be visible in the Error Log view and workspace log file. To disable reporting of such errors a user should locate the following line in `eclipse.ini` file

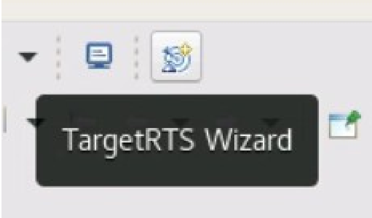
```
-Dorg.eclipse.swt.graphics.Resource.reportNonDisposed=true
```

and replace true with false, so it looks like this:

```
-Dorg.eclipse.swt.graphics.Resource.reportNonDisposed=false
```

6.2 Configuring GTK Theme

On Linux some GTK themes (for example Adwaita on RHEL8) can set tooltip backgrounds to be dark like in the image below.



This should be avoided, because it can cause issues with tooltips Model RealTime will be showing in diagrams. To address this a user can either look for a theme with a light tooltips background or override the theme’s settings by creating a file `~/.config/gtk-3.0/gtk.css` with the following contents:

```
tooltip { text-shadow: none; }
tooltip.background { background-color: rgba(255, 255, 255, 1.0);
border: 1px solid rgba(0, 0, 0, 1.0); }
tooltip decoration { background-color: transparent; }
tooltip * { background-color: transparent; color: black; }
```

After saving the file Model RealTime should be restarted for the changes to become effective.

7 Installing without Internet Access

As explained in the chapter Install it may be necessary during the installation of Model RealTime to download additional Eclipse plugins, that are needed by Model RealTime but are missing in your Eclipse installation. Usually this happens automatically (provided you have marked the option **Contact all update sites during install to find required software**). However, if you don't have access to the internet this obviously won't work. For example, this can happen if your machine is behind a firewall which blocks access to some internet sites such as eclipse.org.

If you get errors like "missing required plug-in" it's likely that Eclipse failed to download the necessary Eclipse plugins. If you cannot temporarily enable the internet access on the installation machine, you can workaround the problem in the following way: install Model RealTime on a different machine which has access to internet and then copy the installation folder to the target machine.