



Red Hat Developer Tools 1

Using Eclipse 4.15

Installing Eclipse 4.15 and the first steps with the application

Red Hat Developer Tools 1 Using Eclipse 4.15

Installing Eclipse 4.15 and the first steps with the application

Olga Tikhomirova
otikhomi@redhat.com

Supriya Takkhi

Peter Macko

Kevin Owen

Yana Hontyk

Legal Notice

Copyright © 2020 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

Information for users installing and starting to use Red Hat Developer Tools.

Table of Contents

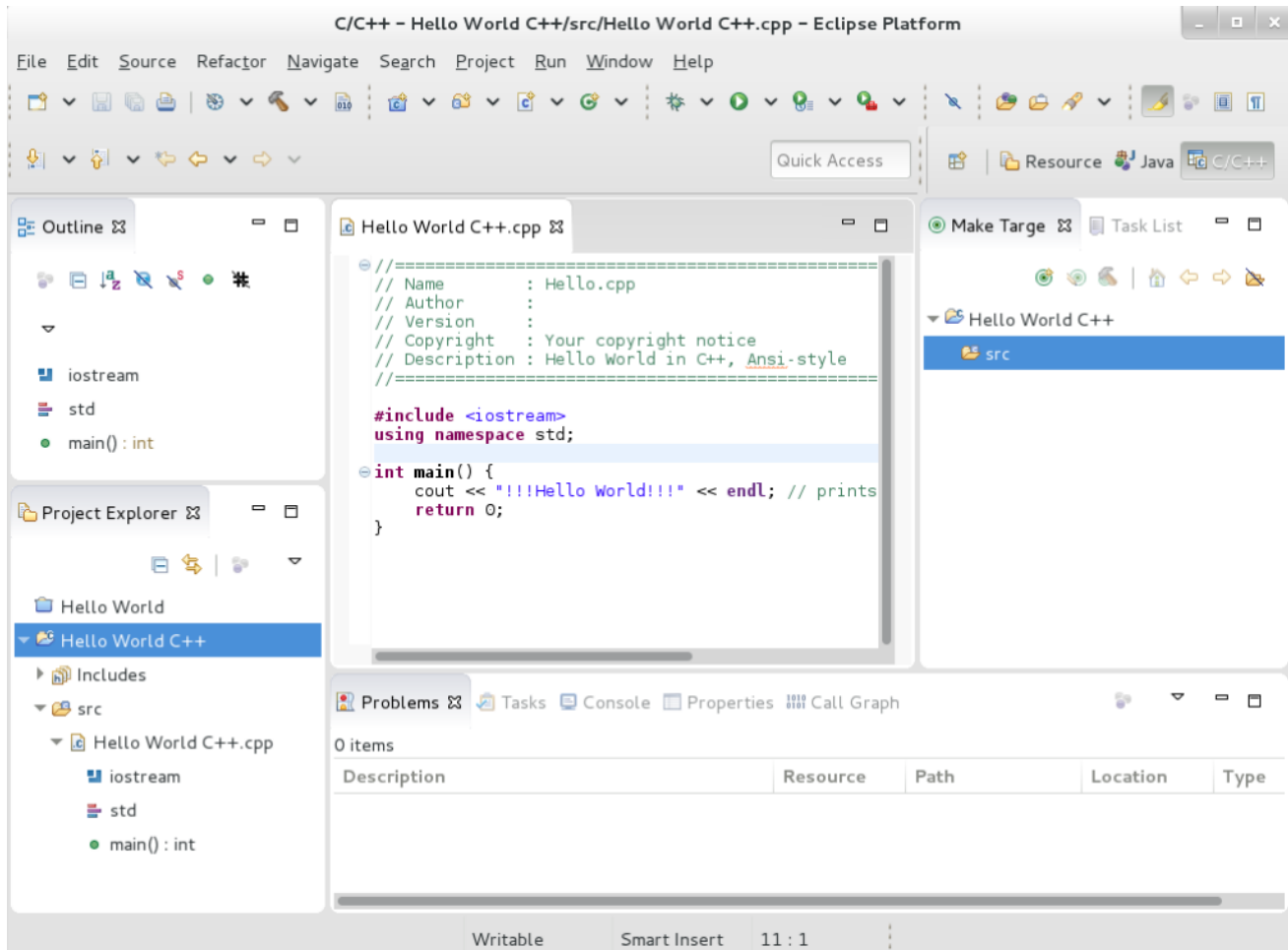
CHAPTER 1. ECLIPSE 4.15	3
1.1. GETTING ACCESS TO ECLIPSE ON RED HAT ENTERPRISE LINUX 7	5
Enabling the Red Hat Developer Tools debuginfo Repositories	6
Additional Resources	6
1.2. INSTALLING ECLIPSE	6
1.3. USING ECLIPSE	7
1.3.1. Using the Red Hat Developer Toolset Toolchain	7
1.3.2. Using the Red Hat Enterprise Linux Toolchain	7
1.4. ADDITIONAL RESOURCES	8
CHAPTER 2. CHANGES IN ECLIPSE 4.15	10
CHAPTER 3. KNOWN ISSUES IN ECLIPSE 4.15	12
3.1. ECLIPSE BECOMES UNRESPONSIVE ON STARTUP	12
3.2. RUNNING A WORKSPACE CREATED IN OLDER ECLIPSE VERSION CAUSES "INITIALIZING ECLIPSE ERROR REPORTING SYSTEM" ERROR	12
3.3. NULLPOINTEREXCEPTION WITH SOME PLUG-INS FROM THIRD-PARTY SOURCES	12
3.4. DEBUGINFO CONFLICTS	12
3.5. CONFLICT BETWEEN THE RHSCL RH-MAVEN33-SCLDEVEL AND RH-MAVEN35-SCLDEVEL PACKAGES	13
3.6. PYDEV USERS MAY EXPERIENCE ISSUES WITH THE PIP INTEGRATION	13
3.7. PYDEV SHORTCUT TO MANAGE PYTHON MODULES DOES NOT WORK	13
3.8. INCOMPATIBILITIES BETWEEN ECLIPSE SUBCLIPSE AND BASE RHEL SUBVERSION	13
3.9. TYCHO CONFLICTS	14

CHAPTER 1. ECLIPSE 4.15

Red Hat Developer Tools, available for Red Hat Enterprise Linux 7, includes Eclipse 4.15, which is based on the Eclipse Foundation's 2020-03 release train.

Eclipse is a powerful development environment that provides tools for each phase of the development process. It integrates a variety of disparate tools into a unified environment to create a rich development experience, provides a fully configurable user interface, and features a pluggable architecture that allows for extension in a variety of ways. For instance, the **Valgrind** plug-in allows programmers to perform memory profiling, that is otherwise performed on the command line through the **Eclipse** user interface.

Figure 1.1. Sample Eclipse Session



Eclipse provides a graphical development environment alternative to traditional interaction with command-line tools, and it is a welcome alternative for developers who do not want to use the command-line interface. The traditional, mostly command-line-based Linux tools (such as **gcc** or **gdb**) and **Eclipse** offer two distinct approaches to programming.

For an overview on how to develop applications for Red Hat JBoss Middleware or for support for OpenShift Tools, see [Red Hat Developer Studio](#).

Table 1.1. Eclipse Components Included inrh-eclipse package collection:

Package	Description

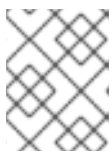
Package	Description
rh-eclipse-eclipse-cdt	The C/C++ Development Tooling (CDT) that provides features and plug-ins for development in these two languages.
rh-eclipse-eclipse-changelog	The ChangeLog plug-in that allows you to create and maintain changelog files.
rh-eclipse-eclipse-dtp	Eclipse Data Tools Platform Project
rh-eclipse-eclipse-egit	EGit, a team provider for Eclipse that provides features and plug-ins for interaction with Git repositories.
rh-eclipse-eclipse-emf	The Eclipse Modeling Framework (EMF) that allows you to build applications based on a structured data model.
rh-eclipse-eclipse-gcov	The GCov plug-in that integrates the GCov test coverage program with Eclipse .
rh-eclipse-eclipse-gef	The Graphical Editing Framework (GEF) that allows you to create a rich graphical editor from an existing application model.
rh-eclipse-eclipse-gprof	The Gprof plug-in that integrates the Gprof performance analysis utility with Eclipse .
rh-eclipse-eclipse-jdt	The Eclipse Java development tools (JDT) plug-in.
rh-eclipse-eclipse-jgit	JGit, a Java implementation of the Git revision control system.
rh-eclipse-eclipse-manpage	The Man Page plug-in that allows you to view manual pages in Eclipse .
rh-eclipse-eclipse-mpc	The Eclipse Marketplace Client.
rh-eclipse-eclipse-mylyn	Mylyn, a task management system for Eclipse .
rh-eclipse-eclipse-pde	The Plugin Development Environment for developing Eclipse plugins.
rh-eclipse-eclipse-perf	The Perf plug-in that integrates the perf tool with Eclipse .

Package	Description
rh-eclipse-eclipse-pydev	A full featured Python IDE for Eclipse .
rh-eclipse-eclipse-remote	The Remote Services plug-in that provides an extensible remote-services framework.
rh-eclipse-eclipse-rpm-editor	The Eclipse Spec File Editor that allows you to maintain RPM spec files.
rh-eclipse-eclipse-systemtap	The SystemTap plug-in that integrates SystemTap with Eclipse .
rh-eclipse-eclipse-subclipse	Subclipse, a team provider for Eclipse that provides features and plug-ins for interaction with Subversion repositories.
rh-eclipse-eclipse-valgrind	The Valgrind plug-in that integrates Valgrind with Eclipse .
rh-eclipse-eclipse-webtools	The Eclipse Webtools plug-ins.

1.1. GETTING ACCESS TO ECLIPSE ON RED HAT ENTERPRISE LINUX 7

Eclipse is an offering that is distributed as a part of the Red Hat Developer Tools content set, which is available to customers with deployments of Red Hat Enterprise Linux 7. In order to install Eclipse, enable the Red Hat Developer Tools and Red Hat Software Collections repositories by using the Red Hat Subscription Management.

1. Choose the system *variant*: **workstation** or **server** and modify the following commands:



NOTE

We recommend developers to use Red Hat Enterprise Linux Server for access to the widest range of development tools.

2. Enable the **rhel-7-variant-devtools-rpms** repository:

```
# subscription-manager repos --enable rhel-7-variant-devtools-rpms
```

3. Enable the **rhel-variant-rhsc1-7-rpms** repository:

```
# subscription-manager repos --enable rhel-variant-rhsc1-7-rpms
```

4. Enable the **rhel-7-variant-optional-rpms** repository:

```
# subscription-manager repos --enable rhel-7-variant-optional-rpms
```

Repositories are enabled. You can install Red Hat Eclipse as described in [Section 1.2, “Installing Eclipse”](#).

Enabling the Red Hat Developer Tools debuginfo Repositories



NOTE

This step is optional.

Red Hat Developer Tools also provides the **debuginfo** packages for all architecture-dependent RPMs included in the repositories. These packages are useful for core-file analysis and for debugging of Eclipse itself.

1. Enable the Red Hat Developer Tools **debuginfo** repositories, replace *variant* with the Red Hat Enterprise Linux system variant (**server** or **workstation**):

```
# subscription-manager repos --enable rhel-7-variant-devtools-debug-rpms
# subscription-manager repos --enable rhel-variant-rhscl-7-debug-rpms
```

For details about installing, understanding, and using the **debuginfo** packages, refer to [Debugging a Running Application](#).

Additional Resources

- For more information on how to register your system using Red Hat Subscription Management and associate it with subscriptions, see the [Red Hat Subscription Management](#) collection of guides.
- For detailed instructions on subscription to Red Hat Software Collections, see the *Red Hat Developer Toolset User Guide*, [Section 1.4. Getting Access to Red Hat Developer Toolset](#) .

1.2. INSTALLING ECLIPSE

The **Eclipse** development environment is provided as a collection of **RPM** packages.

For a list of available components, see [Table 1.1, "Eclipse Components Included in rh-eclipse package collection:"](#).



NOTE

Eclipse fully supports C, C++, and Java development, but does not provide support for the Fortran programming language.



NOTE

The **rh-eclipse** collection requires the **rh-maven35** collection.

Prerequisites

- Repositories enabled as per [Section 1.1, "Getting Access to Eclipse on Red Hat Enterprise Linux 7"](#).

Procedure

1. To install Eclipse, run the following command as the **root** user:

```
# yum install rh-eclipse
```

1.3. USING ECLIPSE

To start Eclipse:

1. Click **Applications** > **Programming** > **Red Hat Eclipse**, or type the following at a shell prompt:

```
$ scl enable rh-eclipse eclipse
```

2. During its startup, **Eclipse** prompts you to select a *workspace* that is a directory in which you want to store your projects. You can either use `~/workspace/`, which is the default option, or click **Browse** to browse your file system and select a custom directory. Additionally, you can select the **Use this as the default and do not ask again** check box to prevent **Eclipse** from displaying this dialog box the next time you run this development environment.
3. Click **OK** to confirm the selection and proceed with the startup.

1.3.1. Using the Red Hat Developer Toolset Toolchain

To use Eclipse with support for the **GNU Compiler Collection** and **binutils** from Red Hat Developer Toolset:

1. Ensure that **devtoolset-9-toolchain** is installed and run the application as described in [Section 1.3, “Using Eclipse”](#). The **rh-eclipse** collection uses the Red Hat Developer Toolset toolchain by default.

For detailed instructions on how to install the **devtoolset-9-toolchain** package on your system, see the *Red Hat Developer Toolset User Guide* [Section 1.5. Installing Red Hat Developer Toolset](#) .



IMPORTANT

Discard all previous build results for projects previously built with the Red Hat Enterprise Linux version of the **GNU Compiler Collection**. To do so, open the project in **Eclipse** and select **Project** > **Clean** from the menu.

1.3.2. Using the Red Hat Enterprise Linux Toolchain

To use Eclipse with support for the toolchain distributed with Red Hat Enterprise Linux, change the configuration of the project to use absolute paths to the Red Hat Enterprise Linux system versions of **gcc**, **g++**, and **as**.

To configure **Eclipse** to explicitly use the Red Hat Enterprise Linux system versions of the tools for the current project:

1. In the C/C++ perspective, click **Project** > **Properties** to open the project properties.
2. In the menu on the left-hand side of the dialog box, click **C/C++ Build** > **Settings**.
3. Click the **Tool Settings** tab.
4. If you are working on a C project:

- a. Select **GCC C Compiler** or **Cross GCC Compiler** and change the value of the **Command** field to:

```
/usr/bin/gcc
```

- b. Select **GCC C Linker** or **Cross GCC Linker** and change the value of the **Command** field to:

```
/usr/bin/gcc
```

- c. Select **GCC Assembler** or **Cross GCC Assembler** and change the value of the **Command** field to:

```
/usr/bin/as
```

5. If you are working on a C++ project:

- a. Select **GCC C Compiler*** or ***Cross G Compiler** and change the value of the **Command** field to:

```
/usr/bin/g++
```

- b. Select **GCC C Compiler** or **Cross GCC Compiler** and change the value of the **Command** field to:

```
/usr/bin/gcc
```

- c. Select **GCC C Linker*** or ***Cross G Linker** and change the value of the **Command** field to:

```
/usr/bin/g++
```

- d. Select **GCC Assembler** or **Cross GCC Assembler** and change the value of the **Command** field to:

```
/usr/bin/as
```

6. Click **OK** to save the configuration changes.

1.4. ADDITIONAL RESOURCES

A detailed description of **Eclipse** and all its features is beyond the scope of this document. For more information, see the following resources.

Installed Documentation

- **Eclipse** includes a built-in help system that provides extensive documentation for each integrated feature and tool.

See Also

- [Chapter 2, Changes in Eclipse 4.15](#) provides a list of selected features and improvements over the **Eclipse** development environment included in the previous release of Developer Tools.

- The [Red Hat Developer Toolset](#) chapter in the User Guide provides an overview of Red Hat Developer Toolset and more information on how to install it on your system.
- The [GNU Compiler Collection \(GCC\)](#) chapter in the User Guide provides information on how to compile programs written in C, C++, and Fortran on the command line.

CHAPTER 2. CHANGES IN ECLIPSE 4.15

Red Hat Developer Tools is distributed with Eclipse 4.15 and plug-ins from the 2020-03 release train that provide a number of bug fixes and feature enhancements.

The following is an abridged list of new features and compatibility changes in this release.

For details on how to use the new features, see the built-in Eclipse documentation.

Significant Package Updates

- **eclipse** 4.14 → 4.15: Eclipse IDE and JDT/PDE plug-ins have been updated to version 4.15. For a more complete list of changes, see the [Eclipse 4.15 – New and Noteworthy](#) page. The highlights include:
 - The Java Editor’s Content Assist now supports subword patterns.
 - Code completion in the Java Editor can now be run in a separate non-UI thread to prevent UI freezes.
 - A variety of new cleanup options and quick fixes have been added.
 - A quick fix has been added to wrap an **Optional** statement.
 - Resource leak analysis has been improved.
 - Debug expressions now support lambda expressions and method references.
 - The Console View can now interpret the ASCII control characters: `\f` - form feed and `\v` - vertical tab.
 - The Console View label now shows the termination time of a process in addition to the launch time.
 - Interactive performance has been improved.
- **eclipse-cdt** 9.10.0 → 9.11.0: The C/C++ Development Tooling has been updated to version 9.11.0. For details, see the upstream [CDT 9.11.0 release notes](#). Notable changes include:
 - Better support for C++14: binary integer literals and digit separators
 - New Code Analysis checkers: variable masking and magic numbers
- **eclipse-linuxtools** 7.5.0 → 7.6.0: The Linux Tools plug-ins have been updated to version 7.6.0 which improves charting functionality. For details, see the upstream [Linux Tools 7.6.0 release notes](#).
- **eclipse-pydev** 7.4.0 → 7.5.0: The PyDev plug-in has been updated to version 7.4.0 which introduces debugger and Cython improvements. For details, see the upstream [PyDev 7.5.0 release notes](#).
- **eclipse-egit/jgit** 5.6.0 → 5.7.0: The Git integration plug-ins have been updated to version 5.7.0. For details, see the upstream [EGit 5.7.0 release notes](#) and [JGit 5.7.0 release notes](#). Notable changes include:
 - A new feature has been added: you can now use commands on repository groups, for example, **pull**.

- Commits comparison has been improved: you can now use unified diffs and search for commits.
- **eclipse-webtools** 3.15 → 3.17: The Web Tools Platform has been updated to version 3.17. The update includes general code improvements and bug fixes. For details, see the upstream [Web Tools Platform 3.16 – New and Noteworthy](#) and [Web Tools Platform 3.17 – New and Noteworthy](#) pages.

Deprecations

- SWTBot removed

The SWTBot tool has been removed from the collection. You can install it from the upstream [SWTBot update site](#), follow the instructions there.

- TestNG removed

Support of the TestNG framework has been ended. TestNG is no longer available through the Marketplace Client (MPC), but you can install the TestNG plug-in via the [TestNG update site](#). However, because of the [NoClassDefFoundError bug](#) in TestNG, **NoClassDefFoundError** occurs while using TestNG P2 Feature, version 7.1.0. To work around this problem, install the later version of TestNG P2 Feature from the [TestNG P2 Feature update site](#).

CHAPTER 3. KNOWN ISSUES IN ECLIPSE 4.15

This section details the known issues in Eclipse 4.15.

3.1. ECLIPSE BECOMES UNRESPONSIVE ON STARTUP

Cause: Eclipse can become unresponsive on startup because of a thread deadlock.

Consequence: Only the Eclipse splash screen is shown without displaying the workspace selection dialog.

Workaround: Terminate the offending Eclipse process by finding its PID and sending the process a kill signal. Example:

```
$ jps
30066 Jps
31631 org.eclipse.equinox.launcher_1.5.300.v20190308-0658.jar
$ kill 31631
```

Result: When the offending process has been terminated, restart Eclipse.

3.2. RUNNING A WORKSPACE CREATED IN OLDER ECLIPSE VERSION CAUSES "INITIALIZING ECLIPSE ERROR REPORTING SYSTEM" ERROR

Workaround: Restart Eclipse with the **-clean** option to clear its dependency resolution cache:

```
$ scl enable rh-eclipse "eclipse -clean"
```

Result: Eclipse will start without this error message.

3.3. NULLPOINTEREXCEPTION WITH SOME PLUG-INS FROM THIRD-PARTY SOURCES

When you install a plug-in from a third-party update site, Eclipse sometimes fails to start with a **NullPointerException** in the workspace log file.

Workaround: Restart Eclipse with the **-clean** option to clear its dependency resolution cache:

```
$ scl enable rh-eclipse "eclipse -clean"
```

Result: Eclipse will start normally.

3.4. DEBUGINFO CONFLICTS

Cause: When a user attempts to install a **debuginfo** package for this collection, for example: **rh-eclipse-eclipse-cdt-debuginfo**, the contents may conflict with the same packages from the earlier collections.

Consequence: The installation of the **rh-eclipse debuginfo** package may fail if the same **debuginfo** package from the **rh-eclipse48** collection is installed.

Workaround: To use **debuginfo** for the **rh-eclipse** collection, first uninstall the **debuginfo** packages from the **rh-eclipse48** collection using this command:


```
$ yum remove rh-eclipse48*debuginfo
```

Result: The installation of the **debuginfo** packages from the **rh-eclipse** collection is successful.

3.5. CONFLICT BETWEEN THE RHSCCL RH-MAVEN33-SCLDEVEL AND RH-MAVEN35-SCLDEVEL PACKAGES

There is a conflict between the RHSCCL **rh-maven33-scldevel** and **rh-maven35-scldevel** packages. This affects the **rh-eclipse-scldevel** package (note that this package is not installed by default).

The **rh-eclipse-scldevel** and **rh-eclipse46-scldevel** packages cannot be present simultaneously.

3.6. PYDEV USERS MAY EXPERIENCE ISSUES WITH THE PIP INTEGRATION

Modules that are pre-installed by the Python SCL cannot be uninstalled. Attempts to do so will result in permission denied errors.

3.7. PYDEV SHORTCUT TO MANAGE PYTHON MODULES DOES NOT WORK

Using the **Ctrl+2** shortcut and selecting **pip** does not function properly.

Consequence: Attempts to use this shortcut result in an **SWTException** in the workspace log.

Workaround: Users can continue to manage their Python modules using the **PyDev > Interpreters** preference.

3.8. INCOMPATIBILITIES BETWEEN ECLIPSE SUBCLIPSE AND BASE RHEL SUBVERSION

Working copies of Subversion repositories created with Eclipse Subclipse are incompatible with the base RHEL version of Subversion. Using the **svn** command on such working copies may result in the following error:

```
$ svn up
svn: E155021: This client is too old to work with the working copy
```

Workaround: Use the pure Java implementation of Subversion used by Eclipse Subclipse on the command line:

```
# yum install rh-eclipse-svnkit-cli # Command line support for SVNKit
```

Now, use the **jsvn** command anywhere you would normally use the **svn** command:

```
$ jsvn up
Updating '':
At revision 16476.
```

3.9. TYCHO CONFLICTS

Cause: The **rh-eclipse-tycho** package conflicts with the same package from the earlier collections, for example: **rh-eclipse48-tycho**.

Consequence: The installation of the **rh-eclipse-tycho** package may fail when the **rh-eclipse48-tycho** package is already installed.

Workaround: Note that only users that want to build or rebuild Eclipse or its plug-ins need Tycho. If needed, uninstall the **rh-eclipse48-tycho** package before installing the **rh-eclipse-tycho** package using this command:

```
$ yum remove rh-eclipse48-tycho
```

Result: The installation of the **rh-eclipse-tycho** package is successful.