



Red Hat Developer Tools 2018.1

Using Eclipse

Installing Eclipse 4.7.2 and first steps with the application

Red Hat Developer Tools 2018.1 Using Eclipse

Installing Eclipse 4.7.2 and first steps with the application

Supriya Takkhi
sbharadw@redhat.com

Legal Notice

Copyright © 2018 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, 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 Software Collections 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

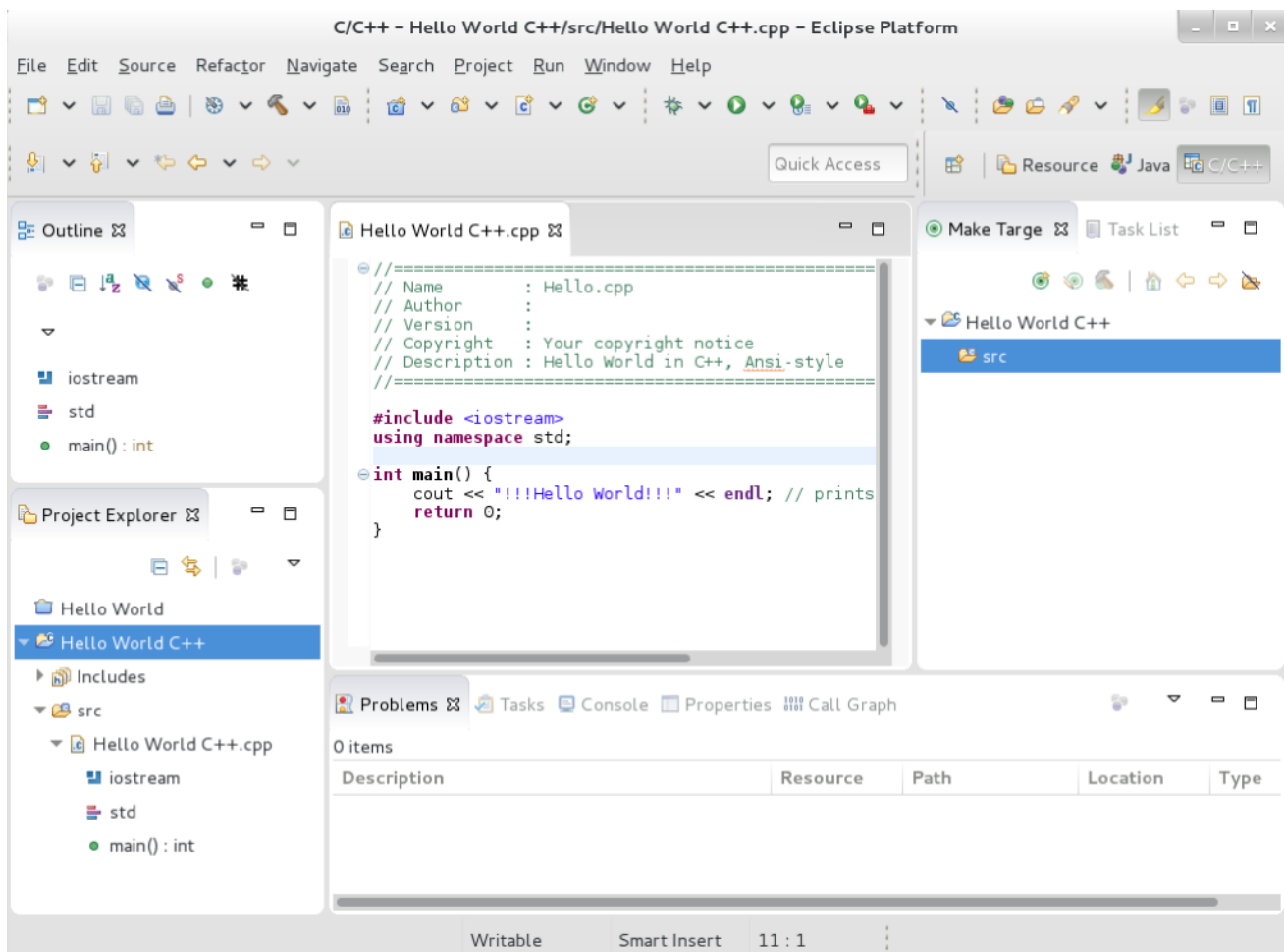
CHAPTER 1. ECLIPSE 4.7.2	3
1.1. ENABLING THE RED HAT DEVELOPER TOOLS REPOSITORIES	5
1.2. INSTALLING ECLIPSE	6
1.3. USING ECLIPSE	6
1.3.1. Using the Red Hat Developer Toolset Toolchain	7
1.3.2. Using the Red Hat Enterprise Linux Toolchain	7
1.4. ADDITIONAL RESOURCES	8
CHAPTER 2. CHANGES SINCE DEVELOPER TOOLS 2.1	9
CHAPTER 3. KNOWN ISSUES IN ECLIPSE 4.7.2	10
3.1. DEBUGINFO CONFLICTS	10
3.2. CONFLICT BETWEEN THE RHSCS RH-MAVEN33-SCLDEVEL AND RH-MAVEN35-SCLDEVEL PACKAGES	10
3.3. 'RUN AS' DOES NOT WORK RELIABLY ON C/C++ PROJECTS	10
3.4. INDEXING OF C/C++ PROJECTS BUILT IN A CONTAINER DOES NOT WORK WELL	10

CHAPTER 1. ECLIPSE 4.7.2

Red Hat Developer Tools, available for Red Hat Enterprise Linux 7, includes Eclipse 4.7.2, which is based on the Eclipse Foundation’s Oxygen release train. Note that `rh-eclipse47` requires the `rh-maven35` collection.

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 an 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 suite such as `gcc` or `gdb` and **Eclipse** offer two distinct approaches to programming.

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

Table 1.1. Eclipse Components Included in `rh-eclipse47`

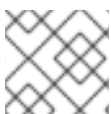
Package	Description
rh-eclipse47-eclipse-cdt	The C/C++ Development Tooling (CDT) that provides features and plug-ins for development in these two languages.
rh-eclipse47-eclipse-changelog	The ChangeLog plug-in that allows you to create and maintain changelog files.
rh-eclipse47-eclipse-dtp	Eclipse Data Tools Platform Project
rh-eclipse47-eclipse-egit	EGit, a team provider for Eclipse that provides features and plug-ins for interaction with Git repositories.
rh-eclipse47-eclipse-emf	The Eclipse Modeling Framework (EMF) that allows you to build applications based on a structured data model.
rh-eclipse47-eclipse-epp-logging	The Eclipse error reporting tool.
rh-eclipse47-eclipse-gcov	The GCov plug-in that integrates the GCov test coverage program with Eclipse.
rh-eclipse47-eclipse-gef	The Graphical Editing Framework (GEF) that allows you to create a rich graphical editor from an existing application model.
rh-eclipse47-eclipse-gprof	The Gprof plug-in that integrates the Gprof performance analysis utility with Eclipse.
rh-eclipse47-eclipse-jdt	The Eclipse Java development tools (JDT) plug-in.
rh-eclipse47-eclipse-jgit	JGit, a Java implementation of the Git revision control system.
rh-eclipse47-eclipse-manpage	The Man Page plug-in that allows you to view manual pages in Eclipse.
rh-eclipse47-eclipse-mpc	The Eclipse Marketplace Client.
rh-eclipse47-eclipse-mylyn	Mylyn, a task management system for Eclipse.
rh-eclipse47-eclipse-oprofile	The OProfile plug-in that integrates OProfile with Eclipse.
rh-eclipse47-eclipse-pde	The Plugin Development Environment for developing Eclipse plugins.

Package	Description
<code>rh-eclipse47-eclipse-perf</code>	The Perf plug-in that integrates the <code>perf</code> tool with Eclipse.
<code>rh-eclipse47-eclipse-ptp</code>	A subset of the PTP project that provides support for synchronized projects.
<code>rh-eclipse47-eclipse-pydev</code>	A full featured Python IDE for Eclipse.
<code>rh-eclipse47-eclipse-recommenders</code>	Code Recommenders
<code>rh-eclipse47-eclipse-remote</code>	The Remote Services plug-in that provides an extensible remote-services framework.
<code>rh-eclipse47-eclipse-rpm-editor</code>	The Eclipse Spec File Editor that allows you to maintain RPM spec files.
<code>rh-eclipse47-eclipse-rse</code>	The Remote System Explorer (RSE) framework that allows you to work with remote systems from Eclipse.
<code>rh-eclipse47-eclipse-systemtap</code>	The SystemTap plug-in that integrates SystemTap with Eclipse.
<code>rh-eclipse47-eclipse-valgrind</code>	The Valgrind plug-in that integrates Valgrind with Eclipse.
<code>rh-eclipse47-eclipse-webtools</code>	The Eclipse Webtools plug-ins.

1.1. ENABLING THE RED HAT DEVELOPER TOOLS REPOSITORIES

To install Eclipse you must first enable the Red Hat Developer Tools repositories.

- Enable the `rhel-7-server-devtools-rpms` repository on Red Hat Enterprise Linux Server or enable the `rhel-7-workstation-devtools-rpms` repository on Red Hat Enterprise Linux Workstation
- Enable the `rhel-server-rhsci-7-rpms` repository on Red Hat Enterprise Linux Server or enable the `rhel-workstation-rhsci-7-rpms` repository on Red Hat Enterprise Linux Workstation



NOTE

Red Hat Developer Tools is not supported on the **Client** or the **ComputeNode** variant.

To enable the Red Hat Developer Tools repositories:

1. Run the following commands as the root user:

```
# subscription-manager repos --enable rhel-7-server-devtools-rpms
# subscription-manager repos --enable rhel-server-rhsc1-7-rpms
```



NOTE

In the above command, depending on the variant of Red Hat Enterprise Linux that you are using, **Server** or **Workstation**, the commands will be:

- For the **Workstation** variant: **subscription-manager repos --enable rhel-7-workstation-devtools-rpms**
- For the **Server** variant: **subscription-manager repos --enable rhel-7-server-devtools-rpms**

For developers, we recommend using Red Hat Enterprise Linux Server for access to the widest range of development tools.

For more information on registering and attaching subscriptions, see [Using and Configuring Red Hat Subscription Management](#).

2. To add the Red Hat Developer Tools key to your system, run the following commands as the root user:

```
# cd /etc/pki/rpm-gpg
# wget -O RPM-GPG-KEY-redhat-devel
  https://www.redhat.com/security/data/a5787476.txt
# rpm --import RPM-GPG-KEY-redhat-devel
```

1.2. INSTALLING ECLIPSE

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

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

```
yum install rh-eclipse47
```

For a list of available components, see [Table 1.1, “Eclipse Components Included in rh-eclipse47”](#).



NOTE

rh-eclipse47 fully supports C, C++, and Java development, but does not provide support for the Fortran programming language.

1.3. USING ECLIPSE

To start **rh-eclipse47**:

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

```
scl enable rh-eclipse47 eclipse
```

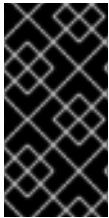
2. During its startup, **Eclipse** prompts you to select a *workspace* that is a directory in which you want to store your projects. You can either use `~/workspace/`, which is the default option, or click **Browse** to browse your file system and select a custom directory. Additionally, you can select 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 `rh-eclipse47` with support for the **GNU Compiler Collection** and **binutils** from Red Hat Developer Toolset:

1. Ensure that **devtoolset-Developer Toolset 7-toolchain** is installed and run the application as described in [Section 1.3, “Using Eclipse”](#). The `rh-eclipse47` Collection uses the Red Hat Developer Toolset toolchain by default.

For detailed instructions on how to install the **devtoolset-Developer Toolset 7-toolchain** package on your system, see [Installing Red Hat Developer Toolset](#).



IMPORTANT

If you are working on a project that you previously built with the Red Hat Enterprise Linux version of the **GNU Compiler Collection**, make sure that you discard all previous build results. 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 `rh-eclipse47` 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. This greatly decreases the initial time investment required for new developers to become fluent in its use. The use of this Help section is detailed in the *Red Hat Enterprise Linux Developer Guide* linked below.

See Also

- [Chapter 2, Changes Since Developer Tools 2.1](#) provides a comprehensive list of 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 SINCE DEVELOPER TOOLS 2.1

Red Hat Developer Tools is distributed with Eclipse 4.7.2 and other plugins from the Oxygen release train, which provide a number of bug fixes and feature enhancements over the Red Hat Developer Tools 2.1 version. Following is a comprehensive list of new features and compatibility changes in this release. For details on how to use these new features, see the built-in Eclipse documentation.

Dependencies

This version of the rh-eclipse47 collection no longer depends on the rh-java-common and rh-maven33 collections. It now depends only on the new rh-maven35 collection.

Significant Package Updates

- eclipse 4.7.1 → 4.7.2: Eclipse Platform was updated from 4.7.1 to 4.7.2. This mainly consisted of bug fixes and continued improvement of support for Java 9.
 - Upstream notes: <https://www.eclipse.org/eclipse/news/4.7.2/>
- eclipse-cdt 9.3.2 → 9.4: CDT was updated from 9.3.2 to 9.4 and includes the important new feature of supporting building C/C++ projects inside of docker containers.
 - Upstream notes: <https://wiki.eclipse.org/CDT/User/NewIn94>
- eclipse-dltk 5.8.1 → 5.8.2: DLTk was updated to bug fix release 5.8.2.
- eclipse-egit/jgit 4.8.0 → 4.10.0: Git support in Eclipse was updated from 4.8.0 to 4.10 and has focused on performance and usability improvements.
 - Upstream Notes:
 - https://wiki.eclipse.org/JGit/New_and_Noteworthy/4.10
 - https://wiki.eclipse.org/JGit/New_and_Noteworthy/4.9
 - https://wiki.eclipse.org/EGit/New_and_Noteworthy/4.10
 - https://wiki.eclipse.org/EGit/New_and_Noteworthy/4.9
- eclipse-linuxtools 6.1.1 → 6.2.0: Linxtools was updated to bug fix release 6.2.0, which included significant work in the Docker tooling plug-ins to enable CDT's "build in container" support.
 - Upstream notes: https://wiki.eclipse.org/Linux_Tools_Project/News/NewIn62
- eclipse-m2e-core-1.8.1 → 1.8.2: Eclipse Maven integration to update to bug fix release 1.8.2 for improved support for Java 9 and the included maven stack was also updated from Maven 3.3 to Maven 3.5.
- eclipse-mylyn 3.23.1 → 3.23.2: Mylyn was updated to bugfix release 3.23.2.
- eclipse-webtools 3.9.1 → 3.9.2: The Webtools plugins were updated to bug fix release 3.9.2 for improved support for Java 9.

CHAPTER 3. KNOWN ISSUES IN ECLIPSE 4.7.2

This section details the Known Issues in Eclipse 4.7.2.

3.1. DEBUGINFO CONFLICTS

Cause: When a user attempts to install a `debuginfo` package for this collection, for example: `rh-eclipse47-eclipse-cdt-debuginfo`, the contents may conflict with the same packages from the earlier collections.

Consequence: The installation of the `rh-eclipse47 debuginfo` package may fail if the same `debuginfo` package from the `rh-eclipse46` collection is installed.

Workaround: To use `debuginfo` for the `rh-eclipse47` collection, `debuginfo` packages from the `rh-eclipse46` collection should first be uninstalled using the command `yum remove rh-eclipse46*debuginfo`.

Result: The installation of the `debuginfo` packages from the `rh-eclipse47` collection is successful.

3.2. CONFLICT BETWEEN THE RHSCLRH-MAVEN33-SCLDEVEL AND RH-MAVEN35-SCLDEVEL PACKAGES

There is a conflict between the RHSCLRH-MAVEN33-SCLDEVEL and RH-MAVEN35-SCLDEVEL packages. This affects the `rh-eclipse47-scldevel` package (note that these are not installed by default):

- `rh-eclipse47-scldevel` cannot be directly updated, meaning, running the `yum update` command fails.
- `rh-eclipse47-scldevel` and `rh-eclipse46-scldevel` cannot be present together.

3.3. 'RUN AS' DOES NOT WORK RELIABLY ON C/C++ PROJECTS

If you alternate between `Run As → C/C++ Container Application` and `Run As → Local C/C++ Application`, for a C/C++ project, you may observe two issues:

- The program is not launched at all
- The program is launched as if you chose another option

3.4. INDEXING OF C/C++ PROJECTS BUILT IN A CONTAINER DOES NOT WORK WELL

While the build itself works, a user observes missing include files and unresolved symbols. There is no known workaround at this moment. Note that building in a container is a new feature that was not present in any of the previous DevTools releases.