Red Hat Developer Tools 1

Using Eclipse 4.18

Installing Eclipse 4.18 and the first steps with the application
Installing Eclipse 4.18 and the first steps with the application

Eva-Lotte Gebhardt
egebhard@redhat.com

Olga Tikhomirova
otikhomi@redhat.com

Peter Macko

Kevin Owen

Yana Hontyk
Abstract

Information for users installing and starting to use Red Hat Developer Tools.
Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright’s message.
CHAPTER 1. ECLIPSE 4.18

Red Hat Developer Tools on Red Hat Enterprise Linux 7 is an offering for developers on the RHEL platform that includes Eclipse 4.18, which is based on the Eclipse Foundation’s 2020-12 release train.

The Eclipse development environment provides tools for each phase of the development process. Eclipse 4.18 on RHEL 7 supports Java development.

To learn more about Eclipse, see the main Eclipse foundation page.

Sample Eclipse session

Eclipse provides a graphical development environment and is therefore an alternative to using the command-line interface.

For an overview of how to develop applications for Red Hat JBoss Middleware or for support of OpenShift Tools, see Red Hat Developer Studio.

1.1. ENABLING ACCESS TO ECLIPSE RPMS ON RED HAT ENTERPRISE LINUX 7

Eclipse is part of the Red Hat Developer Tools content set for RHEL 7. To install Eclipse, enable the Red Hat Developer Tools, Red Hat Software Collections, and Optional repositories using the Red Hat Subscription Management utility.

Prerequisites

- The host must be registered and attached to a subscription.
For more information on registering your system using Red Hat Subscription Management and associating it with subscriptions, see the Red Hat Subscription Management collection of guides.

**Procedure**

Choose the system variant, either workstation or server, to use in the following commands. Red Hat recommends to choose server for access to the widest range of development tools.

1. Enable the `rhel-7-variant-devtools-rpms` repository to access Red Hat Developer Tools:

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

2. Enable the `rhel-variant-rhscl-7-rpms` repository to access Red Hat Software Collections:

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

3. Enable the `rhel-7-variantoptional-rpms` repository to access additional components:

   ```bash
   # subscription-manager repos --enable rhel-7-variantoptional-rpms
   ```

**Optional: Enabling the Red Hat Developer Tools debuginfo repositories**

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

**Procedure**

- Enable the Red Hat Developer Tools debuginfo repositories and replace variant with the Red Hat Enterprise Linux system variant (server or workstation):

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

- Enable the Red Hat Software Collections debuginfo repository:

  ```bash
  # subscription-manager repos --enable rhel-__variant__-rhscl-7-debug-rpms
  ```

**Additional resources**

- For details on installing, understanding, and using the debuginfo packages, refer to Debugging a Running Application.

- For more information on registering your system using Red Hat Subscription Management and associating it with subscriptions, see the Red Hat Subscription Management collection of guides.

- For detailed instructions on managing a 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 following section describes how to install Eclipse.
NOTE

Eclipse is available only on the AMD64 and Intel 64 architecture.

Prerequisites

- On RHEL 7, the repositories must be enabled as per Section 1.1, "Enabling access to Eclipse RPMs on Red Hat Enterprise Linux 7".

Procedure

- On RHEL 7, run the following command:

  # yum install rh-eclipse

1.2.1. Installing additional Eclipse components

Eclipse 4.18 on RHEL 7 supports Java development. To install more components from the upstream repositories, for example to support the C and C++ languages, use the Install New Software wizard, Eclipse Marketplace Client, or the command-line interface.

NOTE

Installing additional Eclipse components is not possible without access to the internet.

1.2.1.1. Installing additional Eclipse components using the Install New Software wizard

Procedure

- To use the Install New Software wizard for the installation of additional components, in the main menu click Help > Install New Software and follow the instructions on the screen.

1.2.1.2. Installing additional Eclipse components using Eclipse Marketplace

To use the Marketplace Client for the installation of additional components, follow the instructions in Section 1.2.1.2.1, "Example: Installing C and C++ Development Tooling (CDT) using the Eclipse Marketplace Client".

1.2.1.2.1. Example: Installing C and C++ Development Tooling (CDT) using the Eclipse Marketplace Client

Procedure

1. From the main menu, select Help > Eclipse Marketplace.

2. In Eclipse Marketplace, use the Find field to search for the wanted component, in this case CDT, and press Go.
3. Click the Install button to start the installation and follow the instructions on the screen.

1.2.1.3. Installing additional Eclipse components using the command-line interface

Red Hat recommends using Eclipse Marketplace or the Install New Software wizard to install additional components to Eclipse. However, it is possible to install components from the command line using the p2 director application.

To use the command-line interface for the installation of additional components, follow the instructions in Section 1.2.1.3.1, “Example: Installing Eclipse C and C++ Development Tools using the command-line interface”.

1.2.1.3.1. Example: Installing Eclipse C and C++ Development Tools using the command-line interface

Prerequisites

- Eclipse is not running.

Procedure

1. In the command-line interface, run the following command:

```
$ scl enable rh-eclipse 'eclipse -noSplash -application org.eclipse.equinox.p2.director -repository https://download.eclipse.org/releases/2020-12 -i org.eclipse.cdt.feature.group'
```
2. Start Eclipse.

Eclipse C/C++ Development Tools is installed.

**WARNING**

Running the p2 director application as root causes significant problems for the RPM consistency. Never run the p2 director application as root.

Additional resources

- For a list of available components, see Section 1.4, "Eclipse Components".
- For further information on the p2 director application, see Installing software using the p2 director application in the online documentation or the built-in help system of Eclipse.

1.3. STARTING ECLIPSE

1.3.1. Starting Eclipse from the GUI

To start Eclipse from the GUI, complete the following steps:

- Click Applications > Programming > Red Hat Eclipse.

1.3.2. Starting Eclipse from the command-line interface

To start Eclipse from the command-line, type the following at a shell prompt:

- On RHEL 7:
  
  ```
  $ scl enable rh-eclipse eclipse
  ```

While starting, Eclipse prompts you to select a workspace directory for your projects. You can use `~/workspace/`, the default option, or click Browse and select a custom directory. You can also select Use this as the default and do not ask again to prevent Eclipse from displaying this dialog box again. Click OK to confirm the selection and proceed with the start.

1.4. ECLIPSE COMPONENTS

The Eclipse development environment is provided as a set of RPM packages. The set contains the following Eclipse components:

**Table 1.1. Eclipse Components on RHEL 7**

<table>
<thead>
<tr>
<th>Package</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rh-eclipse-eclipse-egit</td>
<td>EGit, a team provider for Eclipse, provides features and plug-ins for interaction with Git repositories.</td>
</tr>
</tbody>
</table>
The Eclipse Modeling Framework (EMF) enables you to build applications based on a structured data model.

The Graphical Editing Framework (GEF) enables you to create a rich graphical editor from an existing application model.

The Eclipse Java development tools (JDT) plug-in.

JGit, a Java implementation of the Git revision control system.

The Eclipse Marketplace Client.

The Plugin Development Environment for developing Eclipse plug-ins.

Subclipse, a team provider for Eclipse allows you to interact with Subversion repositories.

The Eclipse Webtools plug-ins.

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. It is accessible from Eclipse’s main menu: Help > Help Contents.

**Other resources**
- For a list of selected features and improvements in the latest version of the Eclipse development environment, see Section 1.5, “Changes in Eclipse 4.18”.

### 1.5. CHANGES IN ECLIPSE 4.18

Eclipse 4.18 ships with Red Hat Developer Tools and plug-ins from the 2020-12 release train that provide a number of bug fixes and feature enhancements.

This section lists notable new features and compatibility changes in this release.

**Significant package updates on RHEL 7**

<table>
<thead>
<tr>
<th>Package</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rh-eclipse-eclipse-emf</td>
<td>The Eclipse Modeling Framework (EMF) enables you to build applications based on a structured data model.</td>
</tr>
<tr>
<td>rh-eclipse-eclipse-gef</td>
<td>The Graphical Editing Framework (GEF) enables you to create a rich graphical editor from an existing application model.</td>
</tr>
<tr>
<td>rh-eclipse-eclipse-jdt</td>
<td>The Eclipse Java development tools (JDT) plug-in.</td>
</tr>
<tr>
<td>rh-eclipse-eclipse-jgit</td>
<td>JGit, a Java implementation of the Git revision control system.</td>
</tr>
<tr>
<td>rh-eclipse-eclipse-mpc</td>
<td>The Eclipse Marketplace Client.</td>
</tr>
<tr>
<td>rh-eclipse-eclipse-pde</td>
<td>The Plugin Development Environment for developing Eclipse plug-ins.</td>
</tr>
<tr>
<td>rh-eclipse-eclipse-subclipse</td>
<td>Subclipse, a team provider for Eclipse allows you to interact with Subversion repositories.</td>
</tr>
<tr>
<td>rh-eclipse-eclipse-webtools</td>
<td>The Eclipse Webtools plug-ins.</td>
</tr>
</tbody>
</table>
Eclipse IDE and JDT/PDE plug-ins have been updated to version 4.18. For a more complete list of changes, see the Eclipse 4.18 – New and Noteworthy page. Notable enhancements include:

- In the **Console** preference page it is now possible to select the new preference "Enable word wrap".

- In the **Appearance** preference page the new "System" theme is now available. It uses system colors to integrate smoothly into your operating system or operating system theme.

- Eclipse JDT has been updated to use JUnit 5.7.

- New clean up options and code formatting options have been added to Java Development Tools.

- In the **Arguments** tab for Java-based launch configurations (Java Application, JUnit, and others), you can now select the new checkbox to write arguments into an @argfile.

**eclipse-egit/jgit 5.9.0 → 5.10.0**

The Git integration plug-ins have been updated to version 5.10.0. For details, see the upstream EGit 5.10.0 release notes and JGit 5.10.0 release notes.

**eclipse-m2e 1.16.2 → 1.17.1**

The Maven integration plug-in has been updated to version 1.17.1.

**Deprecated functionality on RHEL 7**

Python development is no longer supported as part of Eclipse. It can be installed additionally from the Install New Software wizard or Eclipse Marketplace.

**Additional resources**

For details on how to use the new features, see Eclipse Installed documentation.

### 1.6. KNOWN ISSUES IN ECLIPSE 4.18

This section details the known issues in Eclipse 4.18.

**Known issues on RHEL 7**

**Initializing Eclipse Error Reporting System error**

This error occurs when running a workspace created in an older version of Eclipse.

To work around this problem, start Eclipse with the `-clean` option to clear its dependency resolution cache:

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

Eclipse will start without this error message.

**NullPointerExceptions**

NullPointerExceptions can occur when you install a plug-in from a third-party update site. In that case, Eclipse fails to start with a NullPointerException in the workspace log file.

To work around this problem, restart Eclipse with the `-clean` option to clear its dependency resolution cache:

On RHEL 7:
$ scl enable rh-eclipse "eclipse -clean"
Eclipse will start normally.

The **rh-eclipse-tycho** package conflicts with the same package from earlier collections
For example: **rh-eclipse48-tycho**:
As a result, the installation of the **rh-eclipse-tycho** package may fail when the **rh-eclipse48-tycho** package is already installed.
You only need Tycho if you 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
The installation of the **rh-eclipse-tycho** package will now succeed.

The **rh-eclipse-scldevel** package conflicts with packages from earlier collections
For example: **rh-maven36-scldevel**:
As a result, the installation of the **rh-maven36-scldevel** package may fail when the **rh-maven35-scldevel** package is already installed.
To solve this problem, uninstall the **rh-maven35-scldevel** package before installing the new version of **rh-eclipse-scldevel** using this command:

$ yum remove rh-maven35-scldevel
The installation of **rh-eclipse-scldevel** will now succeed.

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

To work around this problem, 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.

Lambda expression evaluation failed due to unexpected argument types
During compilation, some lambda expressions used in conditional breakpoints or Expression view are falsely assigned an Object variable type.

For example, the expression `lotteryNumbers.stream().anyMatch(a → a >= 42)` evaluates the following error message:

```
The operator >= is undefined for the argument type(s) Object, int
```