



Red Hat build of OpenJDK 11

Configuring Red Hat build of OpenJDK 11 on RHEL

Red Hat build of OpenJDK 11 Configuring Red Hat build of OpenJDK 11 on RHEL

Legal Notice

Copyright © 2024 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

Red Hat build of OpenJDK is a Red Hat offering on the Red Hat Enterprise Linux platform. The [Configuring Red Hat build of OpenJDK 11 on RHEL](#) guide provides an overview of this product and explains how to configure the software.

Table of Contents

PROVIDING FEEDBACK ON RED HAT BUILD OF OPENJDK DOCUMENTATION	3
MAKING OPEN SOURCE MORE INCLUSIVE	4
CHAPTER 1. INTERACTIVELY SELECTING A SYSTEM-WIDE RED HAT BUILD OF OPENJDK VERSION ON RHEL	5
CHAPTER 2. NON-INTERACTIVELY SELECTING A SYSTEM-WIDE RED HAT BUILD OF OPENJDK VERSION ON RHEL	7
CHAPTER 3. SELECTING AN INSTALLED RED HAT BUILD OF OPENJDK VERSION FOR A SPECIFIC APPLICATION	8
CHAPTER 4. SELECTING A SYSTEM-WIDE ARCHIVE RED HAT BUILD OF OPENJDK VERSION	9
CHAPTER 5. CONFIGURING THE JAVA_HOME ENVIRONMENT VARIABLE ON RHEL	10
CHAPTER 6. CONFIGURING THE HEAP SIZE FOR RED HAT BUILD OF OPENJDK APPLICATION ON RHEL ..	11

PROVIDING FEEDBACK ON RED HAT BUILD OF OPENJDK DOCUMENTATION

To report an error or to improve our documentation, log in to your Red Hat Jira account and submit an issue. If you do not have a Red Hat Jira account, then you will be prompted to create an account.

Procedure

1. Click the following link to [create a ticket](#).
2. Enter a brief description of the issue in the **Summary**.
3. Provide a detailed description of the issue or enhancement in the **Description**. Include a URL to where the issue occurs in the documentation.
4. Clicking **Submit** creates and routes the issue to the appropriate documentation team.

MAKING OPEN SOURCE MORE INCLUSIVE

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. INTERACTIVELY SELECTING A SYSTEM-WIDE RED HAT BUILD OF OPENJDK VERSION ON RHEL

If you have multiple versions of Red Hat build of OpenJDK installed on RHEL, you can interactively select the default Red Hat build of OpenJDK version to use system-wide.



NOTE

If you do not have root privileges, you can select a Red Hat build of OpenJDK version by configuring the [JAVA_HOME](#) environment variable.

Prerequisites

- You must have root privileges on the system.
- Multiple versions of Red Hat build of OpenJDK were installed using the **yum** package manager.

Procedure

1. View the Red Hat build of OpenJDK versions installed on the system.
\$ yum list installed "java"**

A list of installed Java packages appears.

```

Installed Packages
java-1.8.0-openjdk.x86_64           1:1.8.0.242.b08-0.el8_1          @rhel-8-
appstream-rpms
java-1.8.0-openjdk-headless.x86_64 1:1.8.0.242.b08-0.el8_1          @rhel-8-
appstream-rpms
java-11-openjdk.x86_64             1:11.0.9.10-0.el8_1             @rhel-8-
appstream-rpms
java-11-openjdk-headless.x86_64    1:11.0.9.10-0.el8_1             @rhel-8-
appstream-rpms
javapackages-filesystem.noarch     5.3.0-1.module+el8+2447+6f56d9a6 @rhel-
8-appstream-rpms

```

2. Display the Red Hat build of OpenJDK versions that can be used for a specific **java** command and select the one to use:

```

$ sudo alternatives --config java
There are 2 programs which provide 'java'.

Selection  Command
-----
* 1      java-1.8.0-openjdk.x86_64 (/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.232.b09-
0.el8_0.x86_64/jre/bin/java)
+ 2      java-11-openjdk.x86_64 (/usr/lib/jvm/java-11-openjdk-11.0.9.10-
0.el8_0.x86_64/bin/java)

```

Enter to keep the current selection[+], or type selection number: 1

- The current system-wide Red Hat build of OpenJDK version is marked with an asterisk.

- The current Red Hat build of OpenJDK version for the specified **java** command is marked with a plus sign.
3. Press **Enter** to keep the current selection or enter the **Selection** number of the Red Hat build of OpenJDK version you want to select followed by the **Enter** key.
The default Red Hat build of OpenJDK version for the system is the selected version.
 4. Verify that the chosen binary is selected.

```
$ java -version
openjdk version "11.0.9" 2020-10-15 LTS
OpenJDK Runtime Environment 18.9 (build 11.0.9+10-LTS)
OpenJDK 64-Bit Server VM 18.9 (build 11.0.9+10-LTS, mixed mode, sharing)
```



NOTE

This procedure configures the **java** command. Then **javac** command can be set up in a similar way, but it operates independently.

If you have Red Hat build of OpenJDK installed, **alternatives** provides more possible selections. In particular, the **javac** master alternative switches many binaries provided by the **-devel** sub-package.

Even if you have Red Hat build of OpenJDK installed, **java** (and other JRE masters) and **javac** (and other Red Hat build of OpenJDK masters) still operate separately, so you can have different selections for JRE and JDK. The **alternatives --config java** command affects the **jre** and its associated slaves.

If you want to change Red Hat build of OpenJDK, use the **javac alternatives** command. The **--config javac** utility configures the **SDK** and related slaves. To see all possible masters, use **alternatives --list** and check all of the **java,javac, jre,** and **sdk** masters.

CHAPTER 2. NON-INTERACTIVELY SELECTING A SYSTEM-WIDE RED HAT BUILD OF OPENJDK VERSION ON RHEL

If you have multiple versions of Red Hat build of OpenJDK installed on RHEL, you can select the default Red Hat build of OpenJDK version to use system-wide in a non-interactive way. This is useful for administrators who have root privileges on a Red Hat Enterprise Linux system and need to switch the default Red Hat build of OpenJDK on many systems in an automated way.



NOTE

If you do not have root privileges, you can select a Red Hat build of OpenJDK version by configuring the [JAVA_HOME](#) environment variable.

Prerequisites

- You must have root privileges on the system.
- Multiple versions of Red Hat build of OpenJDK were installed using the **yum** package manager.

Procedure

1. Select the major Red Hat build of OpenJDK version to switch to. For example, for Red Hat build of OpenJDK 11, use **java-11-openjdk**.

```
# PKG_NAME=java-11-openjdk
# JAVA_TO_SELECT=$(alternatives --display java | grep "family $PKG_NAME" | cut -d ' ' -f1)
# alternatives --set java $JAVA_TO_SELECT
```

2. Verify that the active Red Hat build of OpenJDK version is the one you specified.

```
$ java -version
openjdk version "11.0.9" 2020-10-15 LTS
OpenJDK Runtime Environment 18.9 (build 11.0.9+10-LTS)
OpenJDK 64-Bit Server VM 18.9 (build 11.0.9+10-LTS, mixed mode, sharing)
```

CHAPTER 3. SELECTING AN INSTALLED RED HAT BUILD OF OPENJDK VERSION FOR A SPECIFIC APPLICATION

Some applications require a specific Red Hat build of OpenJDK version to run. If multiple versions of Red Hat build of OpenJDK are installed on the system using the **yum** package manager or portable bundle, you can select a Red Hat build of OpenJDK version for each application where necessary by setting the value of the **JAVA_HOME** environment variable or using a wrapper script.

Prerequisites

- Multiple versions of Red Hat build of OpenJDK installed on the machine.
- Ensure that the application you want to run is installed.

Procedure

1. Set the **JAVA_HOME** environment variable. For example, if Red Hat build of OpenJDK 11 was installed using **yum**:

```
$ JAVA_HOME=/usr/lib/jvm/java-11-openjdk
```



NOTE

The symbolic link **java-11-openjdk** is controlled by the **alternatives** command.

2. Do one of the following:

- Launch the application using the default, system-wide configuration.

```
$ mvn --version
Apache Maven 3.5.4 (Red Hat 3.5.4-5)
Maven home: /usr/share/maven
Java version: 11.0.9, vendor: Oracle Corporation, runtime: /usr/lib/jvm/java-11-openjdk-11.0.9.10-0.el8_0.x86_64/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "4.18.0-144.el8.x86_64", arch: "amd64", family: "unix"
```

- Launch the application specifying the **JAVA_HOME** variable:

```
$ JAVA_HOME=/usr/lib/jvm/java-11-openjdk mvn --version
Apache Maven 3.5.4 (Red Hat 3.5.4-5)
Maven home: /usr/share/maven
Java version: 11.0.9, vendor: Oracle Corporation, runtime: /usr/lib/jvm/java-11-openjdk-11.0.9.10-0.el8_0.x86_64
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "4.18.0-144.el8.x86_64", arch: "amd64", family: "unix"
```

CHAPTER 4. SELECTING A SYSTEM-WIDE ARCHIVE RED HAT BUILD OF OPENJDK VERSION

If you have multiple versions of Red Hat build of OpenJDK installed with the archive on RHEL, you can select a specific Red Hat build of OpenJDK version to use system-wide.

Prerequisites

- Know the locations of the Red Hat build of OpenJDK versions installed using the archive.

Procedure

To specify the Red Hat build of OpenJDK version to use for a single session:

1. Configure **JAVA_HOME** with the path to the Red Hat build of OpenJDK version you want used system-wide.

```
$ export JAVA_HOME=/opt/jdk/jdk-11.0.9
```

2. Add **\$JAVA_HOME/bin** to the **PATH** environment variable.

```
$ export PATH="$JAVA_HOME/bin:$PATH"
```

To specify the Red Hat build of OpenJDK version to use permanently for a single user, add these commands into `~/.bashrc`:

```
export JAVA_HOME=/opt/jdk/jdk-11.0.9
export PATH="$JAVA_HOME/bin:$PATH"
```

To specify the Red Hat build of OpenJDK version to use permanently for all users, add these commands into `/etc/bashrc`:

```
export JAVA_HOME=/opt/jdk/jdk-11.0.9
export PATH="$JAVA_HOME/bin:$PATH"
```



NOTE

If you do not want to redefine **JAVA_HOME**, add only the **PATH** command to **bashrc**, specifying the path to the Java binary. For example, **export PATH="/opt/jdk/jdk-11.0.3/bin:\$PATH"**.

Additional resources

- Be aware of the exact meaning of **JAVA_HOME**. For more information, see [Changes/Decouple system java setting from java command setting](#).

CHAPTER 5. CONFIGURING THE JAVA_HOME ENVIRONMENT VARIABLE ON RHEL

Some applications require you to set the **JAVA_HOME** environment variable so that they can find the Red Hat build of OpenJDK installation.

Prerequisites

- You know where you installed Red Hat build of OpenJDK on your system. For example, **/opt/jdk/11**.

Procedure

1. Set the value of **JAVA_HOME**.

```
$ export JAVA_HOME=/opt/jdk/11
```

2. Verify that **JAVA_HOME** is set correctly.

```
$ printenv | grep JAVA_HOME
JAVA_HOME=/opt/jdk/11
```



NOTE

You can make the value of **JAVA_HOME** persistent by exporting the environment variable in **~/.bashrc** for single users or **/etc/bashrc** for system-wide settings. Persistent means that if you close your terminal or reboot your computer, you do not need to reset a value for the **JAVA_HOME** environment variable.

The following example demonstrates using a text editor to enter commands for exporting **JAVA_HOME** in **~/.bashrc** for a single user:

```
> vi ~/.bash_profile

export JAVA_HOME=/opt/jdk/11
export PATH="$JAVA_HOME/bin:$PATH"
```

Additional resources

- Be aware of the exact meaning of **JAVA_HOME**. For more information, see [Changes/Decouple system java setting from java command setting](#).

CHAPTER 6. CONFIGURING THE HEAP SIZE FOR RED HAT BUILD OF OPENJDK APPLICATION ON RHEL

You can configure Red Hat build of OpenJDK to use a customized heap size.

Procedure

- Add the maximum heap size option to the **java** command when running your application. For example, to set the maximum heap size to 100 megabytes, use the **-Xmx100m** option:

```
$ java -Xmx100m <your_application_name>
```

Additional resources

- For more information about the **Xmx** option, see **-Xmxsize** in the [Java documentation](#).

Revised on 2024-05-09 16:46:06 UTC