Red Hat build of OpenJDK 21

Configuring Red Hat build of OpenJDK 21 on RHEL
Configuring Red Hat build of OpenJDK 21 on RHEL
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 21 on RHEL guide provides an overview of this product and explains how to configure the software.
# Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE ......................................................... 3

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION ........................................ 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
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.
PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your feedback on our documentation. To provide feedback, you can highlight the text in a document and add comments.

This section explains how to submit feedback.

Prerequisites

- You are logged in to the Red Hat Customer Portal.
- In the Red Hat Customer Portal, view the document in Multi-page HTML format.

Procedure

To provide your feedback, perform the following steps:

1. Click the Feedback button in the top-right corner of the document to see existing feedback.

   **NOTE**

   The feedback feature is enabled only in the Multi-page HTML format.

2. Highlight the section of the document where you want to provide feedback.

3. Click the Add Feedback pop-up that appears near the highlighted text.
   A text box appears in the feedback section on the right side of the page.

4. Enter your feedback in the text box and click Submit.
   A documentation issue is created.

5. To view the issue, click the issue tracker link in the feedback view.
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 1:1.8.0.302.b08-0.el8_4 @rhel-8-appstream-rpms
   java-11-openjdk.x86_64 1:11.0.12.0.7-0.el8_4 @rhel-8-appstream-rpms
   java-11-openjdk-headless.x86_64 1:11.0.12.0.7-0.el8_4 @rhel-8-appstream-rpms
   java-17-openjdk.x86_64 1:17.0.0.0.35-4.el8 @rhel-8-appstream-rpms
   java-17-openjdk-headless.x86_64 1:17.0.0.0.35-4.el8 @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 3 programs which provide 'java'.

   Selection   Command
   ------------------------------------------------------
   1  java-11-openjdk.x86_64 (/usr/lib/jvm/java-11-openjdk-11.0.12.0.7-0.el8_4/bin/java)
   * 2  java-1.8.0-openjdk.x86_64 (/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.302.b08-0.el8_4.x86_64/jre/bin/java)
   + 3  java-17-openjdk.x86_64 (/usr/lib/jvm/java-17-openjdk-17.0.0.0.35-4.el8.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 "17" 2021-09-14
   OpenJDK Runtime Environment 21.9 (build 17+35)
   OpenJDK 64-Bit Server VM 21.9 (build 17+35, 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 21, use `java-17-openjdk`.

   ```
   # PKG_NAME=java-17-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 "17" 2021-09-14
   OpenJDK Runtime Environment 21.9 (build 17+35)
   OpenJDK 64-Bit Server VM 21.9 (build 17+35, 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 21 was installed using `yum`:

   ```bash
   $ JAVA_HOME=/usr/lib/jvm/java-17-openjdk
   ```

   **NOTE**
   The symbolic link `java-17-openjdk` is controlled by the `alternatives` command.

2. Do one of the following:

   - Launch the application using the default, system-wide configuration.
     ```bash
     $ 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_0.x86_64", arch: "amd64", family: "unix"
     ```

   - Launch the application specifying the `JAVA_HOME` variable:
     ```bash
     $ JAVA_HOME=/usr/lib/jvm/java-17-openjdk-17.0.0.0.35-4.el8.x86_64/ mvn --version
     Apache Maven 3.5.4 (Red Hat 3.5.4-5)
     Maven home: /usr/share/maven
     Java version: 17, vendor: Red Hat, Inc., runtime: /usr/lib/jvm/java-17-openjdk-17.0.0.0.35-4.el8.x86_64
     Default locale: en_US, platform encoding: UTF-8
     OS name: "linux", version: "4.18.0-305.19.1.el8_4.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/openjdk-17.0.0.0.35
   ```
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/openjdk-17.0.0.0.35
   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/openjdk-17.0.0.0.35
   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/openjdk-17.0.0.0.35/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 2023-11-14 18:18:46 UTC