



Red Hat build of OpenJDK 21

在 RHEL 上配置红帽构建的 OpenJDK 21

法律通告

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.

摘要

Red Hat build of OpenJDK 是 Red Hat Enterprise Linux 平台上的红帽产品。在 RHEL 中配置红帽构建的 OpenJDK 21 概述了此产品，并解释了如何配置软件。

目录

提供有关红帽构建的 OPENJDK 文档的反馈	3
使开源包含更多	4
第 1 章 在 RHEL 中以交互方式选择系统范围的红帽构建的 OPENJDK 版本	5
第 2 章 在 RHEL 上，以非交互方式选择系统范围的 RED HAT OPENJDK 版本	7
第 3 章 为特定应用程序选择已安装的 OPENJDK 版本	8
第 4 章 选择系统范围的归档红帽构建的 OPENJDK 版本	9
第 5 章 在 RHEL 上配置 JAVA_HOME 环境变量	10
第 6 章 在 RHEL 上为红帽构建的 OPENJDK 应用程序配置堆大小	11

提供有关红帽构建的 OPENJDK 文档的反馈

要报告错误或改进文档，请登录到 Red Hat JIRA 帐户并提交问题。如果您没有 Red Hat Jira 帐户，则会提示您创建一个帐户。

流程

1. 单击以下链接 [以创建 ticket](#)。
2. 在 **Summary** 中输入问题的简短描述。
3. 在 **Description** 中提供问题或功能增强的详细描述。包括一个指向文档中问题的 URL。
4. 点 **Submit** 创建问题，并将问题路由到适当的文档团队。

使开源包含更多

红帽致力于替换我们的代码、文档和 Web 属性中存在问题的语言。我们从这四个术语开始：master、slave、黑名单和白名单。由于此项工作十分艰巨，这些更改将在即将推出的几个发行版本中逐步实施。详情请查看 [CTO Chris Wright 的信息](#)。

第 1 章 在 RHEL 中以交互方式选择系统范围的红帽构建的 OPENJDK 版本

如果您在 RHEL 上安装了多个版本的 OpenJDK，您可以以互动方式选择默认的 Red Hat build of OpenJDK 版本来使用系统范围。



注意

如果您没有 root 特权，您可以通过配置 [JAVA_HOME](#) 环境变量来选择 OpenJDK 版本的红帽构建。

先决条件

- 必须具有系统上的 root 权限。
- 使用 **yum** 软件包管理器安装多个版本的 OpenJDK。

流程

1. 查看系统上安装 OpenJDK 版本的红帽构建。

```
$ yum list installed "java**"
```

此时会出现已安装的 Java 软件包列表。

```
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. 显示可用于特定 **java** 命令的红帽构建的 OpenJDK 版本，并选择要使用的 OpenJDK 版本：

```
$ 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.x86_64/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

- 当前系统范围的 Red Hat build of OpenJDK 版本标记为星号。
 - 指定 **java** 命令的当前红帽 OpenJDK 版本构建标有加号。
3. 按 **Enter** 键保留当前选择，或者输入您要选择的 Red Hat build of OpenJDK 版本的 **Selection** 号，后跟 **Enter** 键。
系统的默认 Red Hat build of OpenJDK 版本是所选版本。
 4. 验证是否选择了所选二进制文件。

```
$ 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)
```



注意

此流程配置 **java** 命令。然后，可以使用类似的方式设置 **javac** 命令，但它独立运行。

如果您安装了红帽构建的 OpenJDK，则 **替代方案** 会提供更多可能选择。特别是 **javac** master 备用切换由 **-devel** 子软件包提供的许多二进制文件。

即使您安装了红帽构建的 OpenJDK，**java**（及其他 JRE master）和 **javac**（及其他红帽构建的 OpenJDK 主控机）仍然独立运行，因此您可以对 JRE 和 JDK 有不同的选择。**alternatives --config java** 命令会影响 **jre** 及其关联的从设备。

如果要更改红帽构建的 OpenJDK，请使用 **javac alternatives** 命令。**--config javac** 实用程序配置 **SDK** 和相关的从设备。要查看所有可能的 master，请使用 **alternatives --list** 并检查所有 **java**、**java c**、**jre** 和 **sdk** master。

第 2 章 在 RHEL 上，以非交互方式选择系统范围的 RED HAT OPENJDK 版本

如果您在 RHEL 上安装了多个版本的 OpenJDK，您可以选择默认的 Red Hat build of OpenJDK 版本，来以非互动方式使用系统范围的系统。这对于在 Red Hat Enterprise Linux 系统上具有 root 权限的管理人员来说，需要以自动化的方式在很多系统上切换默认的 OpenJDK 构建。



注意

如果您没有 root 特权，您可以通过配置 [JAVA_HOME 环境变量](#) 选择 OpenJDK 版本的红帽构建。

先决条件

- 必须具有系统上的 root 权限。
- 使用 `yum` 软件包管理器安装多个版本的 OpenJDK。

流程

1. 选择要切换到的 OpenJDK 版本的主要红帽构建。例如，对于红帽构建的 OpenJDK 21，请使用 `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. 验证有效红帽构建的 OpenJDK 版本是您指定的版本。

```
$ 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)
```

第 3 章 为特定应用程序选择已安装的 OPENJDK 版本

有些应用程序需要特定的红帽构建的 OpenJDK 版本才能运行。如果使用 **yum** 软件包管理器或可移植捆绑包在系统中安装多个版本的 OpenJDK，您可以通过设置 **JAVA_HOME** 环境变量的值或使用 wrapper 脚本为每个应用程序选择 OpenJDK 版本。

先决条件

- 机器上安装的 Red Hat build of OpenJDK 的多个版本。
- 确保已安装您要运行的应用程序。

流程

1. 设置 **JAVA_HOME** 环境变量。例如，如果红帽构建的 OpenJDK 21 是使用 **yum** 安装的：


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



注意

符号链接 **java-17-openjdk** 由 **alternatives** 命令控制。

2. 执行以下操作之一：

- 使用默认的系统范围配置启动应用程序。

```
$ 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"
```

- 启动指定 **JAVA_HOME** 变量的应用程序：

```
$ 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"
```

第 4 章 选择系统范围的归档红帽构建的 OPENJDK 版本

如果您在 RHEL 上安装了多个版本的 OpenJDK，则可以选择一个特定的红帽构建的 OpenJDK 版本用于系统范围的使用。

先决条件

- 知道使用存档安装的红帽构建的 OpenJDK 版本的位置。

流程

指定用于单个会话的 OpenJDK 版本的红帽构建：

1. 使用您希望系统范围的 OpenJDK 版本的红帽构建路径配置 **JAVA_HOME**。
\$ export JAVA_HOME=/opt/jdk/openjdk-17.0.0.0.35
2. 将 **\$JAVA_HOME/bin** 添加到 **PATH** 环境变量中。
\$ export PATH="\$JAVA_HOME/bin:\$PATH"

要指定红帽为单个用户永久使用的 OpenJDK 版本构建，请将这些命令添加到 **~/.bashrc** 中：

```
export JAVA_HOME=/opt/jdk/openjdk-17.0.0.0.35
export PATH="$JAVA_HOME/bin:$PATH"
```

要指定红帽为所有用户永久使用的 OpenJDK 版本构建，请将这些命令添加到 **/etc/bashrc** 中：

```
export JAVA_HOME=/opt/jdk/openjdk-17.0.0.0.35
export PATH="$JAVA_HOME/bin:$PATH"
```



注意

如果您不想重新定义 **JAVA_HOME**，仅将 **PATH** 命令添加到 **bashrc** 中，指定到 Java 二进制文件的路径。例如，导出 **PATH="/opt/jdk/openjdk-17.0.0.0.35/bin:\$PATH"**。

其他资源

- 请注意 **JAVA_HOME** 的确切含义。如需更多信息，请参阅 [java 命令设置中的更改/覆盖系统 java 设置](#)。

第 5 章 在 RHEL 上配置 JAVA_HOME 环境变量

有些应用程序要求您设置 **JAVA_HOME** 环境变量，以便它们可以找到红帽构建的 OpenJDK 安装。

先决条件

- 您知道在系统上安装红帽构建的 OpenJDK 的位置。例如：`/opt/jdk/11`。

流程

1. 设置 **JAVA_HOME** 的值。

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

2. 验证 **JAVA_HOME** 是否已正确设置。

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



注意

您可以通过在 `~/.bashrc` 中导出单个用户或系统范围的 `/etc/bashrc` 中的环境变量来永久保留 **JAVA_HOME** 的值。持久性意味着，如果您关闭终端或重启计算机，则不需要为 **JAVA_HOME** 环境变量重置值。

以下示例演示了如何使用文本编辑器为单个用户在 `~/.bashrc` 中输入用于导出 **JAVA_HOME** 的命令：

```
> vi ~/.bash_profile

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

其他资源

- 请注意 **JAVA_HOME** 的确切含义。如需更多信息，请参阅 [java 命令设置中的更改/覆盖系统 java 设置](#)。

第 6 章 在 RHEL 上为红帽构建的 OPENJDK 应用程序配置堆大小

您可以将红帽构建的 OpenJDK 配置为使用自定义的堆大小。

流程

- 在运行应用程序时，将最大堆大小选项添加到 **java** 命令。例如，要将最大堆大小设置为 100MB，请使用 **-Xmx100m** 选项：

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

其他资源

- 有关 **Xmx** 选项的更多信息，请参阅 [Java 文档中的 -Xmxsize](#)。

更新于 2024-05-10