



## Red Hat build of OpenJDK 21

RHEL での Red Hat build of OpenJDK 21 の設定



# Red Hat build of OpenJDK 21 RHEL での Red Hat build of OpenJDK 21 の 設定

---

## 法律上の通知

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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> 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<sup>®</sup> 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.

## 概要

Red Hat build of OpenJDK は、Red Hat Enterprise Linux プラットフォーム上の Red Hat 製品です。RHEL での Red Hat build of OpenJDK 21 の設定 ガイドでは、この製品の概要と、ソフトウェアの設定方法を説明します。

---

## 目次

RED HAT BUILD OF OPENJDK ドキュメントへのフィードバック .....	3
多様性を受け入れるオープンソースの強化 .....	4
第1章 RHEL でシステム全体の RED HAT BUILD OF OPENJDK バージョンを対話的に選択 .....	5
第2章 RHEL でシステム全体の RED HAT BUILD OF OPENJDK バージョンを非対話的に選択 .....	7
第3章 特定のアプリケーション用にインストールされた RED HAT BUILD OF OPENJDK バージョンの選択 ...	8
第4章 システム全体のアーカイブ RED HAT BUILD OF OPENJDK バージョンを対話的に選択 .....	9
第5章 RHEL での JAVA_HOME 環境変数の設定 .....	10
第6章 RHEL での RED HAT BUILD OF OPENJDK アプリケーションのヒープサイズの設定 .....	11



## RED HAT BUILD OF OPENJDK ドキュメントへのフィードバック

エラーを報告したり、ドキュメントを改善したりするには、Red Hat Jira アカウントにログインし、課題を送信してください。Red Hat Jira アカウントをお持ちでない場合は、アカウントを作成するように求められます。

### 手順

1. 次のリンクをクリックして [チケットを作成します](#)。
2. **Summary** に課題の簡単な説明を入力します。
3. **Description** に課題や機能拡張の詳細な説明を入力します。問題があるドキュメントのセクションへの URL を含めてください。
4. **Submit** をクリックすると、課題が作成され、適切なドキュメントチームに転送されます。

## 多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、[Red Hat CTO である Chris Wright のメッセージ](#) を参照してください。



## 第1章 RHEL でシステム全体の RED HAT BUILD OF OPENJDK バージョンを対話的に選択

RHEL に Red Hat build of OpenJDK が複数インストールされている場合は、システム全体で使用するデフォルトの Red Hat build of OpenJDK バージョンを対話形式で選択できます。



### 注記

root 権限がない場合は、[JAVA\\_HOME 環境変数を設定](#)して Red Hat build of OpenJDK バージョンを選択できます。

### 前提条件

- システムに対する root 権限がある。
- **yum** パッケージマネージャーを使用して、複数のバージョンの Red Hat build of OpenJDK がインストールされている。

### 手順

1. システムにインストールされている Red Hat build of OpenJDK のバージョンを表示します。  
**\$ yum list installed "java\*\*"**

インストールされている Java パッケージのリストが表示されます。

```

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. 特定の **java** コマンドに使用できる Red Hat build of OpenJDK バージョンを表示し、使用するものを選択します。

```

$ 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

- 現在のシステム全体の Red Hat build of OpenJDK バージョンにはアスタリスクが付いています。
  - 指定した **java** コマンドの現在の Red Hat build of OpenJDK バージョンには、プラス記号が付いています。
3. **Enter** を押して現在の選択を保持するか、選択する Red Hat build of OpenJDK バージョンの **Selection** 番号を入力し、**Enter** キーを押します。  
システムのデフォルトの Red Hat build of OpenJDK バージョンは、選択したバージョンです。
  4. 選択したバイナリーが選択されていることを確認します。

```
$ 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)
```



### 注記

この手順では、**java** コマンドを設定します。次に、**javac** コマンドは同様の方法で設定できますが、独立して動作します。

Red Hat build of OpenJDK をインストールしている場合、**alternatives** はより多くの選択肢を提供します。特に、**javac** マスターの代替は、**-devel** サブパッケージが提供する多くのバイナリーを切り替えます。

Red Hat build of OpenJDK がインストールされていても **java** (およびその他の JRE マスター) や **javac** (およびその他の Red Hat build of OpenJDK マスター) は別々に動作するため、JRE と JDK で異なる選択を行うことができます。**alternatives --config java** コマンドは、**jre** とそれに関連付けられたスレーブに影響します。

Red Hat build of OpenJDK を変更する場合は、**javac alternatives** コマンドを使用します。**--config javac** ユーティリティーは、**SDK** および関連するスレーブを設定します。可能なすべてのマスターを表示するには、**alternatives --list** を使用して、**java**、**javac**、**jre**、**sdk** マスターのすべてを確認します。

## 第2章 RHEL でシステム全体の RED HAT BUILD OF OPENJDK バージョンを非対話的に選択

RHEL に Red Hat build of OpenJDK が複数インストールされている場合は、システム全体で使用するデフォルトの Red Hat build of OpenJDK バージョンを非対話形式で選択できます。これは、Red Hat Enterprise Linux システムで root 権限を持ち、多くのシステムのデフォルト Red Hat build of OpenJDK を自動的に切り替える必要がある管理者に役立ちます。



### 注記

root 権限がない場合は、[JAVA\\_HOME 環境変数を設定](#) して Red Hat build of OpenJDK バージョンを選択できます。

### 前提条件

- システムに対する root 権限がある。
- **yum** パッケージマネージャーを使用して、複数のバージョンの Red Hat build of OpenJDK がインストールされている。

### 手順

1. 切り替える Red Hat build of OpenJDK のメジャーバージョンを選択します。たとえば、Red Hat build of OpenJDK 21 の場合は **java-17-openjdk** を使用します。

```
# 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. アクティブな Red Hat build of OpenJDK バージョンが指定したバージョンであることを確認します。

```
$ 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 特定のアプリケーション用にインストールされた RED HAT BUILD OF OPENJDK バージョンの選択

アプリケーションによっては、実行に特定の Red Hat build of OpenJDK バージョンを必要とするものもあります。**yum** パッケージマネージャーまたはポータブルバンドルを使用して、複数のバージョンの Red Hat build of OpenJDK がシステムにインストールされている場合は、**JAVA\_HOME** 環境変数の値を設定するか、ラッパースクリプトを使用して、必要に応じて各アプリケーションの Red Hat build of OpenJDK バージョンを選択できます。

### 前提条件

- Red Hat build of OpenJDK の複数のバージョンがマシンにインストールされている。
- 実行するアプリケーションがインストールされている。

### 手順

1. **JAVA\_HOME** 環境変数を設定します。たとえば、**yum** を使用して Red Hat build of OpenJDK 21 がインストールされた場合は、以下を使用します。

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



#### 注記

シンボリックリンク **java-17-openjdk** は、**alternatives** コマンドにより制御されます。

2. 次のいずれかを行います。

- システム全体のデフォルト設定を使用してアプリケーションを起動します。

```
$ 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"
```

- **JAVA\_HOME** 変数を指定してアプリケーションを起動します。

```
$ 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 システム全体のアーカイブ RED HAT BUILD OF OPENJDK バージョンを対話的に選択

RHEL にアーカイブを使用して複数のバージョンの Red Hat build of OpenJDK がインストールされている場合は、システム全体で使用する特定の Red Hat build of OpenJDK バージョンを選択できます。

### 前提条件

- アーカイブを使用してインストールされた Red Hat build of OpenJDK のバージョンの場所を把握している。

### 手順

1つのセッションに使用する Red Hat build of OpenJDK バージョンを指定するには、以下のコマンドを実行します。

1. システム全体で使用する Red Hat build of OpenJDK バージョンへのパスで **JAVA\_HOME** を設定します。  
**\$ export JAVA\_HOME=/opt/jdk/openjdk-17.0.0.0.35**
2. **\$JAVA\_HOME/bin** を **PATH** 環境変数に追加します。  
**\$ export PATH="\$JAVA\_HOME/bin:\$PATH"**

1人のユーザーに永続的に使用する Red Hat build of OpenJDK バージョンを指定するには、以下のコマンドを **~/.bashrc** に追加します。

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

すべてのユーザーに永続的に使用する Red Hat build of OpenJDK バージョンを指定するには、以下のコマンドを **/etc/bashrc** に追加します。

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



### 注記

**JAVA\_HOME** を再定義しない場合は、Java バイナリーへのパスを指定して、PATH コマンドのみを **bashrc** に追加します。たとえば、**export PATH="/opt/jdk/openjdk-17.0.0.0.35/bin:\$PATH"** になります。

### 関連情報

- **JAVA\_HOME** の正確な意味に注意してください。詳細は、[システム設定を java コマンドの設定から変更/分離](#) を参照してください。

## 第5章 RHEL での JAVA\_HOME 環境変数の設定

一部のアプリケーションでは、Red Hat build of OpenJDK のインストールを検出できるように **JAVA\_HOME** 環境変数を設定する必要があります。

### 前提条件

- システムのどこに Red Hat build of OpenJDK をインストールしたかを理解している。たとえば、`/opt/jdk/11` です。

### 手順

1. **JAVA\_HOME** の値を設定します。

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

2. **JAVA\_HOME** が正しく設定されていることを確認します。

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



### 注記

単一ユーザーの場合は `~/.bashrc` の環境変数、システム全体の設定の場合は `/etc/bashrc` の環境変数をエクスポートして **JAVA\_HOME** の値を永続的にすることができます。永続的とは、端末を閉じたりコンピューターを再起動したりした場合に、**JAVA\_HOME** 環境変数の値をリセットする必要がないことを意味します。

次の例は、テキストエディターを使用して、シングルユーザーの `~/.bashrc` に **JAVA\_HOME** をエクスポートするためのコマンドを入力する方法を示しています。

```
> vi ~/.bash_profile  
  
export JAVA_HOME=/opt/jdk/11  
export PATH="$JAVA_HOME/bin:$PATH"
```

### 関連情報

- **JAVA\_HOME** の正確な意味に注意してください。詳細は、[システム設定を java コマンドの設定から変更/分離](#) を参照してください。

## 第6章 RHEL での RED HAT BUILD OF OPENJDK アプリケーションのヒープサイズの設定

カスタマイズされたヒープサイズを使用するように Red Hat build of OpenJDK を設定できます。

### 手順

- アプリケーションの実行時に **java** コマンドに最大ヒープサイズオプションを追加します。たとえば、最大ヒープサイズを 100 メガバイトに設定するには、**-Xmx100m** オプションを使用します。

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

### 関連情報

- Xmx** オプションの詳細は、[Java ドキュメント](#) の **-Xmxsize** を参照してください。

改訂日時: 2024-05-10