



Red Hat build of OpenJDK 21

RHEL での Red Hat build of OpenJDK 21 のインストールと使用

Red Hat build of OpenJDK 21 RHEL での Red Hat build of OpenJDK 21 のインストールと使用

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[®] 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.

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。Red Hat build of OpenJDK 21 のインストールと使用 ガイドでは、この製品の概要と、ソフトウェアをインストールして使用を開始する方法を説明します。

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 RED HAT BUILD OF OPENJDK 21の概要	5
第2章 RED HAT ENTERPRISE LINUX での RED HAT BUILD OF OPENJDK 21のインストール	6
2.1. YUM を使用して RHEL に JRE をインストール	6
2.2. アーカイブを使用した RHEL への JRE のインストール	7
2.3. YUM を使用した RHEL での RED HAT BUILD OF OPENJDK のインストールおよび使用	8
2.4. アーカイブを使用した RHEL での RED HAT BUILD OF OPENJDK のインストールおよび使用	8
2.5. YUM を使用した RHEL への RED HAT BUILD OF OPENJDK のメジャーバージョンの複数インストール	10
2.6. アーカイブを使用して RHEL に OPENJDK の複数のメジャーバージョンをインストール	11
2.7. YUM を使用して RHEL に OPENJDK の複数のマイナーバージョンをインストール	11
2.8. アーカイブを使用して RHEL に OPENJDK の複数のマイナーバージョンをインストール	12
第3章 RED HAT BUILD OF OPENJDK 21のデバッグシンボル	13
3.1. デバッグシンボルのインストール	13
3.2. デバッグシンボルのインストール場所の確認	13
3.3. デバッグシンボルの設定の確認	14
3.4. 致命的なエラーログファイルでのデバッグシンボルの設定	15
第4章 RED HAT ENTERPRISE LINUX での RED HAT BUILD OF OPENJDK 21の更新	17
4.1. YUM を使用した RHEL での RED HAT BUILD OF OPENJDK 21の更新	17
4.2. アーカイブを使用した RHEL での RED HAT BUILD OF OPENJDK 21の更新	17

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](#) をご覧ください。

第1章 RED HAT BUILD OF OPENJDK 21 の概要

OpenJDK (Open Java Development Kit) は、Java Platform Standard Edition (Java SE) のオープンソース実装です。Red Hat build of OpenJDK は、8u、11u、17u、21u の 4 つのバージョンで利用できます。

Red Hat build of OpenJDK 向けパッケージは、Red Hat Enterprise Linux および Microsoft Windows で利用でき、Red Hat Ecosystem Catalog の JDK および JRE として同梱されています。

第2章 RED HAT ENTERPRISE LINUX での RED HAT BUILD OF OPENJDK 21 のインストール

Red Hat build of OpenJDK は、モバイルアプリケーションからデスクトップアプリケーション、Web アプリケーション、エンタープライズシステムまで、プラットフォームに依存しない幅広いアプリケーションを開発および実行するための環境です。Red Hat は、Red Hat build of OpenJDK と呼ばれる Java Platform SE (Standard Edition) のオープンソース実装を提供します。

アプリケーションは、JDK (Java Development Kit) を使用して開発されます。アプリケーションは、JRE (Java ランタイム環境) および JDK に含まれる JVM (Java 仮想マシン) で実行されます。フットプリントが最小で、ユーザーインターフェイスに必要なライブラリーが含まれていないヘッドレスバージョンの Java もあります。ヘッドレスバージョンは、ヘッドレスサブパッケージにパッケージ化されています。



注記

JRE と JDK のどちらが必要かわからない場合は、JDK をインストールすることが推奨されます。

以下のセクションでは、Red Hat Enterprise Linux に Red Hat build of OpenJDK をインストールする手順を説明します。



注記

Red Hat build of OpenJDK の複数のメジャーバージョンをローカルシステムにインストールできます。あるメジャーバージョンから別のメジャーバージョンに切り替える必要がある場合は、コマンドラインインターフェイス (CLI) で以下のコマンドを実行し、画面のプロンプトに従います。

```
$ sudo update-alternatives --config 'java'
```

2.1. YUM を使用して RHEL に JRE をインストール

システムパッケージマネージャー (**yum**) を使用して、Red Hat build of OpenJDK Java Runtime Environment (JRE) をインストールできます。

前提条件

- root 権限があるユーザーとしてシステムにログインしている。
- ローカルシステムを Red Hat Subscription Manager アカウントに登録している。[Registering a system using Red Hat Subscription Manager](#) ユーザーガイドを参照してください。

手順

1. インストールするパッケージを指定して、**yum** コマンドを実行します。

```
$ sudo yum install java-21-openjdk
```

2. インストールが機能することを確認します。

```
$ java -version
```



注記

前のコマンドの出力で、システムで Red Hat build of OpenJDK の別のメジャーバージョンがチェックアウトされていることが分かった場合は、CLI で以下のコマンドを入力して、システムを Red Hat build of OpenJDK 21 を使用するように切り替えることができます。

```
$ sudo update-alternatives --config 'java'
```

2.2. アーカイブを使用した RHEL への JRE のインストール

アーカイブを使用して Red Hat build of OpenJDK Java Runtime Environment (JRE) をインストールできます。これは、Java 管理者が root 権限を持っていない場合に役立ちます。



注記

後続バージョンのアップグレードを容易にするために、JRE を含む親ディレクトリーを作成し、汎用パスを使用して最新の JRE へのシンボリックリンクを作成します。

手順

1. アーカイブファイルをダウンロードするディレクトリーを作成し、コマンドラインインターフェイス (CLI) でそのディレクトリーに移動します。以下に例を示します。

```
$ mkdir ~/jres
```

```
$ cd ~/jres
```

2. Red Hat カスタマーポータル [Software Downloads](#) ページに移動します。
3. **Version** ドロップダウンリストから Red Hat build of OpenJDK 21 の最新バージョンを選択し、Linux 用の JRE アーカイブをローカルシステムにダウンロードします。
4. アーカイブのコンテンツを任意のディレクトリーにデプロイメントします。以下に例を示します。

```
$ tar -xf archive_file_name.tar.gz -C ~/jres
```

5. アップグレードを容易にするために、JRE へのシンボリックリンクを使用して汎用パスを作成します。

```
$ ln -s ~/jres/archive_file_name ~/jres/java-21
```

6. **JAVA_HOME** 環境変数を設定します。

```
$ export JAVA_HOME=~/.jres/java-21
```

7. **JAVA_HOME** 環境変数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=~/.jres/java-21
```



注記

この方法でインストールした場合、Java は現在のユーザーのみが使用できません。

- 一般的な JRE パスの **bin** ディレクトリーを **PATH** 環境変数に追加します。

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

- 完全パスを指定せずに **java -version** が機能することを確認します。

```
$ java -version
```



注記

~/.bashrc に環境変数をエクスポートすることで、**JAVA_HOME** 環境変数が現在のユーザーに対して持続することを確認できます。

2.3. YUM を使用した RHEL での RED HAT BUILD OF OPENJDK のインストールおよび使用

システムパッケージマネージャー **yum** を使用して、Red Hat build of OpenJDK をインストールできません。

前提条件

- root 権限を持つユーザーとしてログインしている。
- ローカルシステムを Red Hat Subscription Manager アカウントに登録している。[Registering a system using Red Hat Subscription Manager](#) ユーザーガイドを参照してください。

手順

- インストールするパッケージを指定して、**yum** コマンドを実行します。

```
$ sudo yum install java-21-openjdk-devel
```

- インストールが機能することを確認します。

```
$ javac -version
```

```
javac 21.0.1
```

2.4. アーカイブを使用した RHEL での RED HAT BUILD OF OPENJDK のインストールおよび使用

Red Hat build of OpenJDK は、アーカイブでインストールできます。これは、Java 管理者が root 権限を持っていない場合に役立ちます。



注記

アップグレードを容易にするために、JRE を含む親ディレクトリーを作成し、汎用パスを使用して最新の JRE へのシンボリックリンクを作成します。

手順

1. アーカイブファイルをダウンロードするディレクトリーを作成し、コマンドラインインターフェイス (CLI) でそのディレクトリーに移動します。以下に例を示します。

```
$ mkdir ~/jdk
```

```
$ cd ~/jdk
```

2. Red Hat カスタマーポータル [Software Downloads](#) ページに移動します。
3. **Version** ドロップダウンリストから Red Hat build of OpenJDK 21 の最新バージョンを選択し、Linux 用の JDK アーカイブをローカルシステムにダウンロードします。
4. アーカイブのコンテンツを任意のディレクトリーにデプロイメントします。

```
$ tar -xf archive_file_name.tar.xz -C ~/jdk
```

5. アップグレードを容易にするために、JDK へのシンボリックリンクを使用して汎用パスを作成します。

```
$ ln -s ~/jdk/archive_file_name ~/jdk/java-21
```

6. **JAVA_HOME** 環境変数を設定します。

```
$ export JAVA_HOME=~/jdk/java-21
```

7. **JAVA_HOME** 環境変数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=~/jdk/java-21
```



注記

この方法でインストールした場合、Java は現在のユーザーのみが使用できません。

8. 一般的な JRE パスの **bin** ディレクトリーを **PATH** 環境変数に追加します。

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

9. 完全パスを指定せずに **java -version** が機能することを確認します。

```
$ java -version
```



注記

~/**.bashrc** に環境変数をエクスポートすることで、**JAVA_HOME** 環境変数が現在のユーザーに対して持続することを確認できます。

2.5. YUM を使用した RHEL への RED HAT BUILD OF OPENJDK のメジャーバージョンの複数インストール

システムパッケージマネージャー **yum** を使用して、Red Hat build of OpenJDK の複数バージョンをインストールできます。

前提条件

- インストールする Red Hat build of OpenJDK を提供するリポジトリへのアクセスを提供するアクティブなサブスクリプションを持つ Red Hat Subscription Manager (RHSM) アカウント。
- システムに対する root 権限がある。

手順

1. 以下の **yum** コマンドを実行してパッケージをインストールします。
Red Hat build of OpenJDK 21

```
$ sudo yum install java-21-openjdk
```

Red Hat build of OpenJDK 17

```
$ sudo yum install java-17-openjdk
```

Red Hat build of OpenJDK 11

```
$ sudo yum install java-11-openjdk
```

Red Hat build of OpenJDK 8

```
$ sudo yum install java-1.8.0-openjdk
```

2. インストール後に、利用可能な Java バージョンを確認します。

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

3. 現在の Java バージョンを確認します。

```
$ java -version
```



注記

Red Hat build of OpenJDK の複数のメジャーバージョンをローカルシステムにインストールできます。あるメジャーバージョンから別のメジャーバージョンに切り替える必要がある場合は、コマンドラインインターフェイス (CLI) で以下のコマンドを実行し、画面のプロンプトに従います。

```
$ sudo update-alternatives --config 'java'
```

関連情報

- **java --alternatives** を使用して、使用するデフォルトの Java バージョンを設定できます。詳細は、[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](#) を参照してください。

2.6. アーカイブを使用して RHEL に OPENJDK の複数のメジャーバージョンをインストール

Red Hat build of OpenJDK の複数のメジャーバージョンをインストールするには、**アーカイブ**を使用した RHEL への JRE のインストールと同じ手順を使用するか、複数のメジャーバージョンを使用して **アーカイブ**を使用して RHEL 8 に Red Hat build of OpenJDK をインストールできます。



注記

システムのデフォルトの Red Hat build of OpenJDK バージョンを設定する方法については、[システム全体の Java バージョンの選択](#) を参照してください。

関連情報

- JRE のインストール方法は、[アーカイブを使用した RHEL への JRE のインストール](#) を参照してください。
- JDK のインストール方法は、[アーカイブを使用した RHEL 8 への Red Hat build of OpenJDK のインストール](#) を参照してください。

2.7. YUM を使用して RHEL に OPENJDK の複数のマイナーバージョンをインストール

RHEL には、Red Hat build of OpenJDK の複数のマイナーバージョンをインストールできます。これは、インストールされているマイナーバージョンが更新されないようにすることで行われます。

前提条件

- [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](#) から、システム全体の Red Hat build of OpenJDK バージョンを選択している。

手順

- `/etc/yum.conf` ディレクトリーに `installonlypkgs` オプションを追加して、`yum` がインストール可能でも更新できない Red Hat build of OpenJDK パッケージを指定します。

```
installonlypkgs=java-<version>--openjdk,java-<version>--openjdk-  
headless,java-<version>--openjdk-devel
```

更新は、システムに古いバージョンを残したまま、新しいパッケージをインストールします。



注記

Red Hat build of OpenJDK のさまざまなマイナーバージョンは、`/usr/lib/jvm/<minor version>` ファイルにあります。

2.8. アーカイブを使用して RHEL に OPENJDK の複数のマイナーバージョンをインストール

複数のマイナーバージョンのインストールは、複数のマイナーバージョンを使用したアーカイブを使用した RHEL への JRE のインストール または アーカイブを使用した RHEL 8 への Red Hat build of OpenJDK のインストールと同じです。



注記

システムのデフォルトのマイナーバージョンを選択する方法は、[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](#) を参照してください。

関連情報

- JRE のインストール方法は、[アーカイブを使用した RHEL への JRE のインストール](#) を参照してください。
- JDK のインストール方法は、[アーカイブを使用した RHEL 8 への Red Hat build of OpenJDK のインストール](#) を参照してください。

第3章 RED HAT BUILD OF OPENJDK 21 のデバッグシンボル

Red Hat build of OpenJDK アプリケーションでクラッシュの調査に役立つシンボルのデバッグに役立ちます。

3.1. デバッグシンボルのインストール

この手順では、Red Hat build of OpenJDK のデバッグシンボルをインストールする方法を説明します。

前提条件

- ローカルの system に **gdb** パッケージをインストールしている。
 - CLI で **sudo yum install gdb** コマンドを実行して、ローカルシステムにこのパッケージをインストールできます。

手順

1. デバッグシンボルをインストールするには、以下のコマンドを入力します。

```
$ sudo debuginfo-install java-21-openjdk
$ sudo debuginfo-install java-21-openjdk-headless
```

これらのコマンドは、**java-21-openjdk-debuginfo**、**java-21-openjdk-headless-debuginfo**、および Red Hat build of OpenJDK 21 バイナリーのデバッグシンボルを提供する追加パッケージをインストールします。これらのパッケージは自己完全ではなく、実行可能なバイナリーは含まれません。



注記

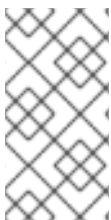
debuginfo-install は、**yum-utils** パッケージで提供されます。

2. デバッグシンボルがインストールされていることを確認するには、以下のコマンドを入力します。

```
$ gdb which java
Reading symbols from /usr/bin/java...Reading symbols from /usr/lib/debug/usr/lib/jvm/java-21-version/bin/java-21-version.x86_64.debug...done.
(gdb)
```

3.2. デバッグシンボルのインストール場所の確認

この手順では、デバッグシンボルの場所を見つける方法を説明します。



注記

debuginfo パッケージがインストールされていても、パッケージのインストール場所を取得できない場合は、正しいパッケージと java バージョンがインストールされているかどうかを確認します。バージョンを確認した後、再度デバッグシンボルの場所を確認してください。

前提条件

- ローカルの system に **gdb** パッケージをインストールしている。
 - CLI で **sudo yum install gdb** コマンドを実行して、ローカルシステムにこのパッケージをインストールできます。
 - デバッグシンボルパッケージをインストールしている。[デバッグシンボルのインストール](#)を参照してください。

手順

- デバッグシンボルの場所を見つけるには、**which java** コマンドで **gdb** を使用します。

```
$ gdb which java
```

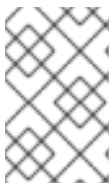
```
Reading symbols from /usr/bin/java...Reading symbols from /usr/lib/debug/usr/lib/jvm/java-21-  
openjdk-version/bin/java-version.x86_64.debug...done.  
(gdb)
```

- 以下のコマンドを使用して ***-debug** ディレクトリーを調べて、**java**、**javac**、および **javah** を含むライブラリーのデバッグバージョンをすべて表示します。

```
$ cd /usr/lib/debug/lib/jvm/java-21-openjdk-version
```

```
$ tree
```

```
OJDK 21 version:  
├── java-21-openjdk-version  
│   ├── bin  
│   │   ├── ...  
│   │   ├── java-java-version.x86_64.debug  
│   │   ├── javac-java-version.x86_64.debug  
│   │   └── javadoc-java-version.x86_64.debug  
│   │   └── ...  
│   └── lib  
│       ├── jexec-java-version.x86_64.debug  
│       ├── jli  
│       │   ├── libjli.so-java-version.x86_64.debug  
│       └── jspawnhelper-java-version.x86_64.debug  
│       └── ...
```



注記

javac および **javah** ツールは、**java-21-openjdk-devel** パッケージで提供されます。**\$ sudo debuginfo-install java-21-openjdk-devel** コマンドを使用してパッケージをインストールできます。

3.3. デバッグシンボルの設定の確認

デバッグシンボルの設定を確認および設定できます。

- 以下のコマンドを入力して、インストールされているパッケージのリストを取得します。

```
$ sudo yum list installed | grep 'java-21-openjdk-debuginfo'
```

- デバッグ情報パッケージがインストールされていない場合は、以下のコマンドを実行して、足りないパッケージをインストールします。

```
$ sudo yum debuginfo-install glibc-2.28-151.el8.x86_64 libgcc-8.4.1-1.el8.x86_64 libstdc++-8.4.1-1.el8.x86_64 sssd-client-2.4.0-9.el8.x86_64 zlib-1.2.11-17.el8.x86_64
```

- 特定のブレークポイントに到達する場合は、以下のコマンドを実行します。

```
$ gdb -ex 'handle SIGSEGV noprint nostop pass' -ex 'set breakpoint pending on' -ex 'break JavaCalls::call' -ex 'run' --args java ./HelloWorld
```

上記のコマンドは、以下のタスクを完了します。

- JVM はスタックオーバーフローチェックに SEGV を使用するため、SIGSEGV エラーを処理します。
- 保留中のブレークポイントを **yes** に設定します。
- **JavaCalls::call** 関数で break ステートメントを呼び出します。HotSpot(libjvm.so) でアプリケーションを起動する関数。

3.4. 致命的なエラーログファイルでのデバッグシンボルの設定

JVM クラッシュが原因で Java アプリケーションがダウンすると、致命的なエラーのログファイルが生成されます (例: **hs_error**、**java_error**)。これらのエラーログファイルは、アプリケーションの現在の作業ディレクトリに生成されます。クラッシュファイルには、スタックに関する情報が含まれます。

手順

1. **strip -g** コマンドを使用すると、デバッグシンボルをすべて削除できます。以下のコードは、デプロイメントされていない **hs_error** ファイルの例を示しています。

```
Native frames: (J=compiled Java code, j=interpreted, Vv=VM code, C=native code)
V [libjvm.so+0xb83d2a] Unsafe_SetLong+0xda
j sun.misc.Unsafe.putLong(Ljava/lang/Object;JJ)V+0
j Crash.main([Ljava/lang/String;)V+8
v ~StubRoutines::call_stub
V [libjvm.so+0x6c0e65] JavaCalls::call_helper(JavaValue*, methodHandle*,
JavaCallArguments*, Thread*)+0xc85
V [libjvm.so+0x73cc0d] jni_invoke_static(JNIEnv_*, JavaValue*, _jobject*, JNI_CallType,
_jmethodID*, JNI_ArgumentPusher*, Thread*) [clone .constprop.1]+0x31d
V [libjvm.so+0x73fd16] jni_CallStaticVoidMethod+0x186
C [libjli.so+0x48a2] JavaMain+0x472
C [libpthread.so.0+0x9432] start_thread+0xe2
```

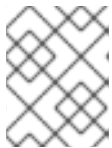
以下のコードは、ストライピング **hs_error** ファイルの例を示しています。

```
Stack: [0x00007ff7e1a44000,0x00007ff7e1b44000], sp=0x00007ff7e1b42850, free
space=1018k
Native frames: (J=compiled Java code, j=interpreted, Vv=VM code, C=native code)
V [libjvm.so+0xa7ecab]
```

```
j sun.misc.Unsafe.putAddress(JJ)V+0
j Crash.crash()V+5
j Crash.main([Ljava/lang/String;)V+0
v ~StubRoutines::call_stub
V [libjvm.so+0x67133a]
V [libjvm.so+0x682bca]
V [libjvm.so+0x6968b6]
C [libjli.so+0x3989]
C [libpthread.so.0+0x7dd5] start_thread+0xc5
```

- 以下のコマンドを入力して、同じバージョンのデバッグシンボルと致命的なエラーログファイルがあることを確認します。

```
$ java -version
```



注記

このチェックを完了するには、**sudo update-alternatives --config 'java'** を使用することもできます。

- nm** コマンドを使用して、**libjvm.so** に ELF データおよびテキストシンボルがあることを確認します。

```
$ nm /usr/lib/debug/usr/lib/jvm/java-21-
openjdk-version/lib/server/libjvm.so-version.x86_64.debug
```

関連情報

- クラッシュファイル **hs_error** は、デバッグシンボルがインストールされない状態で不完全です。詳細は、[Java application down due to JVM crash](#) を参照してください。

第4章 RED HAT ENTERPRISE LINUX での RED HAT BUILD OF OPENJDK 21 の更新

以下のセクションでは、Red Hat Enterprise Linux で Red Hat build of OpenJDK 21 を更新する手順を説明します。

4.1. YUM を使用した RHEL での RED HAT BUILD OF OPENJDK 21 の更新

インストールされている Red Hat build of OpenJDK パッケージは、**yum** システムパッケージマネージャーを使用して更新できます。

前提条件

- システムに対する root 権限がある。

手順

- 現在の Red Hat build of OpenJDK バージョンを確認します。

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

インストールされている Red Hat build of OpenJDK パッケージのリストが表示されます。

- 特定のパッケージを更新します。以下に例を示します。

```
$ sudo yum update java-21-openjdk
```

- 現在の Red Hat build of OpenJDK バージョンをチェックして、更新が機能していることを確認します。

```
$ java -version
```



注記

Red Hat build of OpenJDK の複数のメジャーバージョンをローカルシステムにインストールできます。あるメジャーバージョンから別のメジャーバージョンに切り替える必要がある場合は、コマンドラインインターフェイス (CLI) で以下のコマンドを実行し、画面のプロンプトに従います。

```
$ sudo update-alternatives --config 'java'
```

4.2. アーカイブを使用した RHEL での RED HAT BUILD OF OPENJDK 21 の更新

アーカイブを使用して Red Hat build of OpenJDK を更新できます。これは、Red Hat build of OpenJDK 管理者が root 権限を持たない場合に便利です。

前提条件

- JDK または JRE のインストールを指定する一般的なパスを把握している。たとえば、`~/jdk/java-21` など。

手順

1. JDK または JRE への汎用パスの既存のシンボリックリンクを削除します。
以下に例を示します。

```
$ unlink ~/jdk/java-21
```

2. インストール場所に最新バージョンの JDK または JRE をインストールします。

関連情報

- JRE のインストール方法は、[アーカイブを使用した RHEL への JRE のインストール](#) を参照してください。
- JDK のインストール方法は、[アーカイブを使用した RHEL 8 への Red Hat build of OpenJDK のインストール](#) を参照してください。

改訂日時: 2024-05-10