



## Red Hat build of OpenJDK 11

Red Hat build of OpenJDK 11.0.17 のリリース  
ノート



Red Hat build of OpenJDK 11 Red Hat build of OpenJDK 11.0.17 のリリース  
ノート

---

## 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 11.0.17 のリリースノート では、Red Hat build of OpenJDK 11 の新機能の概要と、潜在的な既知の問題と考えられる回避策のリストを提供します。

---

## 目次

はじめに .....	3
RED HAT BUILD OF OPENJDK ドキュメントへのフィードバック .....	4
多様性を受け入れるオープンソースの強化 .....	5
第1章 RED HAT BUILD OF OPENJDK のサポートポリシー .....	6
第2章 アップストリームの OPENJDK 11 との相違点 .....	7
第3章 RED HAT BUILD OF OPENJDK の機能 .....	8
Red Hat build of OpenJDK の新しい機能と機能拡張 .....	8
cpu.shares パラメーターが無効になっている .....	8
jdk.httpserver.maxConnections システムプロパティー .....	8
JFR を使用してオブジェクトのデシリアライゼーションを監視する .....	8
SHA-1 署名 JAR .....	9
HTTPURLConnection の keep-alive 動作を制御するためのシステムプロパティー .....	10
デフォルトの PKCS #12 MAC アルゴリズムを更新 .....	11
非推奨および削除された機能 .....	11
非推奨の Kerberos 暗号化タイプ .....	11
第4章 このリリースに関連するアドバイザリー .....	13



## はじめに

Open Java Development Kit (OpenJDK) は、Java Platform Standard Edition (Java SE) のオープンソース実装です。Red Hat build of OpenJDK には、8u、11u、17u の3つのバージョンがあります。

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。

## 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 RED HAT BUILD OF OPENJDK のサポートポリシー

Red Hat は、Red Hat build of OpenJDK の一部のメジャーバージョンを製品でサポートします。一貫性を保つために、これらのバージョンは、Oracle が Oracle JDK 向けに長期サポート (LTS) を指定しているバージョンと同じになります。

Red Hat build of OpenJDK のメジャーバージョンは、最初に導入された時点から少なくとも 6 年間サポートされます。詳細は、[OpenJDK のライフサイクルおよびサポートポリシー](#) を参照してください。



### 注記

RHEL 6 のライフサイクルは 2020 年 11 月に終了します。このため、Red Hat build of OpenJDK は、サポート対象の設定として RHEL 6 をサポートしていません。

## 第2章 アップストリームの OPENJDK 11 との相違点

Red Hat Enterprise Linux (RHEL) の Red Hat build of OpenJDK には、OpenJDK のアップストリーム ディストリビューションの構造上の変更が数多く含まれています。Red Hat build of OpenJDK の Microsoft Windows バージョンは、RHEL の更新にできる限り従います。

次のリストは、Red Hat build of OpenJDK 11 の最も注目すべき変更点を詳しく示しています。

- FIPS のサポート。Red Hat build of OpenJDK 11 は、RHEL が FIPS モードであるかどうかを自動的に検出し、Red Hat build of OpenJDK 11 がそのモードで動作するように自動的に設定します。この変更は、Microsoft Windows 向けの Red Hat build of OpenJDK ビルドには適用されません。
- 暗号化ポリシーのサポート。Red Hat build of OpenJDK 11 は、RHEL から有効な暗号化アルゴリズムとキーサイズの制約のリストを取得します。これらの設定コンポーネントは、トランスポート層セキュリティ (TLS) 暗号化プロトコル、証明書パス検証、および署名された JAR によって使用されます。さまざまなセキュリティプロファイルを設定して、安全性と互換性のバランスをとることができます。この変更は、Microsoft Windows 向けの Red Hat build of OpenJDK ビルドには適用されません。
- RHEL の Red Hat build of OpenJDK は、アーカイブ形式のサポート用の **zlib**、イメージのサポート用の **libjpeg-turbo**、**libpng**、**giflib** などのネイティブライブラリーと動的にリンクします。また、RHEL はフォントのレンダリングと管理のために、**Harfbuzz** および **Freetype** に対して動的にリンクします。
- **src.zip** ファイルには、Red Hat build of OpenJDK に同梱されるすべての JAR ライブラリーのソースが含まれています。
- RHEL の Red Hat build of OpenJDK は、タイムゾーン情報のソースとして、システム全体のタイムゾーンデータファイルを使用します。
- RHEL の Red Hat build of OpenJDK は、システム全体の CA 証明書を使用します。
- Microsoft Windows の Red Hat build of OpenJDK には、RHEL で利用可能な最新のタイムゾーンデータが含まれています。
- Microsoft Windows の Red Hat build of OpenJDK は、RHEL から入手可能な最新の CA 証明書を使用します。

### 関連情報

- システムが FIPS モードであるかどうかの検出の詳細は、Red Hat RHEL Planning Jira の [システム FIPS 検出の改善](#) の例を参照してください。
- 暗号化ポリシーの詳細については、[Using system-wide cryptographic policies](#) を参照してください。

## 第3章 RED HAT BUILD OF OPENJDK の機能

最新の Red Hat build of OpenJDK 11 リリースには、新機能が含まれる可能性があります。さらに、最新リリースは、以前の Red Hat build of OpenJDK 11 リリースに由来する機能を強化、非推奨、または削除する可能性があります。



### 注記

その他の変更点やセキュリティ修正については、[OpenJDK 11.0.17 Released](#) を参照してください。

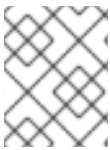
### Red Hat build of OpenJDK の新しい機能と機能拡張

Red Hat build of OpenJDK 11.0.17 リリースに含まれる新しい機能と機能拡張について理解するには、以下のリリースノートを参照してください。

#### cpu.shares パラメーターが無効になっている

Red Hat build of OpenJDK 11.0.17 リリースより前は、Red Hat build of OpenJDK は、**cgroups** と呼ばれる Linux コントロールグループに属する **cpu.shares** パラメーターの誤った解釈を使用していました。このパラメーターにより、Java 仮想マシン (JVM) が使用可能な CPU よりも少ない CPU を使用する可能性があり、コンテナ内で動作するときの JVM の CPU リソースとパフォーマンスに影響を与えます。

Red Hat build of OpenJDK 11.0.17 リリースでは、スレッドプールのスレッド数を決定するときに **cpu.shares** パラメーターを使用しないように JVM が設定されます。この設定を元に戻したい場合は、JVM の起動時に **-XX:+UseContainerCpuShares** 引数を渡します。



### 注記

**-XX:+UseContainerCpuShares** 引数は非推奨の機能であり、将来の Red Hat build of OpenJDK リリースで削除される可能性があります。

[JDK-8281181](#) (JDK バグシステム) を参照してください。

#### jdk.httpserver.maxConnections システムプロパティ

Red Hat build of OpenJDK 11.0.17 では、新しいシステムプロパティ **jdk.httpserver.maxConnections** が追加されています。これにより、**HttpServer** サービスに接続制限が指定されていないというセキュリティ上の問題が修正されます。この問題は、受け入れられた接続と確立された接続が無制限に開いたままになる可能性がありました。

**jdk.httpserver.maxConnections** システムプロパティを使用して、以下のように **HttpServer** サービスの動作を変更できます。

- **0** の値または **-1** などの負の値を設定して、サービスへの接続制限を指定します。
- **1** などの正の値を設定すると、サービスは、確立された接続の現在の数に対して、受け入れた接続をチェックします。サービスに確立された接続に達すると、サービスは受け入れた接続をすぐに終了します。

[JDK-8286918](#) (JDK Bug System) を参照してください。

#### JFR を使用してオブジェクトのデシリアライゼーションを監視する

JDK Flight Recorder (JFR) を使用して、オブジェクトのデシリアライズを監視できるようになりました。デフォルトでは、Red Hat build of OpenJDK 11.0.17 は JFR の **jdk.deserialization** イベント設定を無効にします。この機能を有効にするには、JFR 設定の **event-name** 要素を更新します。以下に例を示

します。

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration version="2.0" description="test">
  <event name="jdk.Deserialization">
    <setting name="enabled">true</setting>
    <setting name="stackTrace">false</setting>
  </event>
</configuration>
```

JFR を有効にし、デシリアライゼーションイベントを監視するように JFR を設定すると、監視対象のアプリケーションがオブジェクトをデシリアライズしようとするたびに、JFR によってイベントが作成されます。次に、JFR のシリアライゼーションフィルターメカニズムは、監視対象アプリケーションからデシリアライズされたオブジェクトを受け入れるか拒否するかを決定できます。

[JDK-8261160](#) (JDK Bug System) を参照してください。

## SHA-1 署名 JAR

Red Hat build of OpenJDK 11.0.17 リリースでは、**SHA-1** アルゴリズムで署名された JAR はデフォルトで制限され、署名されていないものとして扱われます。これらの制限は、次のアルゴリズムに適用されます。

- ダイジェスト、署名、およびオプションで JAR のタイムスタンプに使用されるアルゴリズム。
- コード署名者とタイムスタンプ機関の証明書チェーン内の証明書の署名アルゴリズムとダイジェストアルゴリズム、およびそれらの証明書が失効しているかどうかを確認するために使用される証明書失効リスト (CRL) またはオンライン証明書ステータスプロトコル (OCSP) 応答。

さらに、制限は署名済みの Java Cryptography Extension (JCE) プロバイダーにも適用されます。

以前にタイムスタンプが付けられた JAR の互換性リスクを軽減するために、この制限は、**SHA-1** アルゴリズムで署名され、**January 01, 2019** より前にタイムスタンプが付けられた JAR には適用されません。この例外は、将来の Red Hat build of OpenJDK リリースで削除される可能性があります。

JAR ファイルが制限の影響を受けるかどうかを判断するには、CLI で次のコマンドを発行します。

```
$ jarsigner -verify -verbose -certs
```

前のコマンドの出力から、**SHA1**、**SHA-1**、または **disabled** のインスタンスを検索します。さらに、JAR が署名なしとして扱われることを示す警告メッセージを検索します。以下に例を示します。

```
Signed by "CN="Signer""
Digest algorithm: SHA-1 (disabled)
Signature algorithm: SHA1withRSA (disabled), 2048-bit key
```

```
WARNING: The jar will be treated as unsigned, because it is signed with a weak algorithm that is now disabled by the security property:
```

```
jdk.jar.disabledAlgorithms=MD2, MD5, RSA keySize < 1024, DSA keySize < 1024, SHA1 denyAfter 2019-01-01
```

新しい制限の影響を受けるすべての JAR をより強力なアルゴリズムに置き換えるか、再署名することを検討してください。

JAR ファイルがこの制限の影響を受ける場合は、アルゴリズムを削除して、**SHA-256** などのより強力

なアルゴリズムでファイルに再署名できます。Red Hat build of OpenJDK 11.0.17 の **SHA-1** 署名付き JAR に対する制限を削除する必要があるため、セキュリティリスクを受け入れる場合は、以下のアクションを実行できます。

1. **java.security** 設定ファイルを変更します。または、このファイルを保存して、必要な設定で別のファイルを作成することもできます。
2. **SHA1 usage SignedJAR & denyAfter 2019 01 011** エントリーを **jdk.certpath.disabledAlgorithms** セキュリティープロパティーから削除します。
3. **jdk.jar.disabledAlgorithms** セキュリティープロパティーから **SHA1 denyAfter 2019-01-01** エントリーを削除します。

## 注記

**java.security** ファイルの **jdk.certpath.disabledAlgorithms** の値は、RHEL 8 および 9 のシステムセキュリティポリシーによって上書きされる場合があります。システムセキュリティポリシーで使用される値は、ファイル **/etc/crypto-policies/back-ends/java.config** で確認でき、**java.security** ファイルで **security.useSystemPropertiesFile** を **false** に設定するか、**-Djava.security.disableSystemPropertiesFile=true** を JVM 渡すことで無効にします。これらの値はこのリリースでは変更されていないため、値は Red Hat build of OpenJDK の以前のリリースと同じままです。

**java.security** ファイルの設定例については、[JBoss EAP for OpenShift の java.security プロパティーのオーバーライド](#) (Red Hat カスタマーポータル) を参照してください。

[JDK-8269039](#) (JDK バグシステム) を参照してください。

## HTTPURLConnection の keep-alive 動作を制御するためのシステムプロパティー

Red Hat build of OpenJDK 11.0.17 リリースには、**HTTPURLConnection** の **キープアライブ** 動作を制御するために使用できる次の新しいシステムプロパティーが含まれています。

- サーバーへの接続を制御する **http.keepAlive.time.server**。
- プロキシへの接続を制御する **http.keepAlive.time.proxy**。

Red Hat build of OpenJDK 11.0.17 リリースより前では、**keep-alive** 時間が指定されていないサーバーまたはプロキシが原因で、ハードコードされたデフォルト値によって定義された期間、アイドル接続が開いたままになる場合があります。

Red Hat build of OpenJDK 11.0.17 では、システムプロパティーを使用して **keep-alive** の時間のデフォルト値を変更できます。**keep-alive** プロパティーは、サーバーまたはプロキシのいずれかの HTTP **keep-alive** 時間を変更することでこの動作を制御します。これにより、Red Hat build of OpenJDK の HTTP プロトコルハンドラーは、指定された秒数が経過した後にアイドル状態の接続を閉じます。

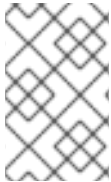
Red Hat build of OpenJDK 11.0.17 リリースより前では、次のユースケースで、**HTTPURLConnection** の特定の **keep-alive** 動作が発生していました。

- サーバーが **Connection:keep-alive** ヘッダーを指定し、サーバーの応答に **Keep-alive:timeout=N** が含まれている場合、クライアントの Red Hat build of OpenJDK **keep-alive** キャッシュは **N** 秒のタイムアウトを使用します (**N** は整数値)。
- サーバーが **Connection:keep-alive** ヘッダーを指定しているが、サーバーの応答に **Keep-alive:timeout=N** のエントリーが含まれていない場合、クライアントの Red Hat build of OpenJDK **keep-alive** キャッシュはプロキシに対して **60** 秒のタイムアウトを使用し、**5** サー

バーの秒。

- サーバーが **Connection:keep-alive** ヘッダーを指定しない場合、クライアントの Red Hat build of OpenJDK **keep-alive** キャッシュは、すべての接続に対して 5 秒のタイムアウトを使用しません。

Red Hat build of OpenJDK 11.0.17 リリースでは、前述の動作はそのままですが、2 番目と 3 番目に挙げた使用例におけるタイムアウトは、デフォルトの設定に依存するのではなく、**http.keepAlive.time.server** および **http.keepAlive.time.proxy** プロパティを使用して指定できるようになっています。



### 注記

**keep-alive** プロパティを設定し、サーバーが **Keep-Alive** 応答ヘッダーの **keep-alive** 時間を指定した場合、HTTP プロトコルハンドラーはサーバーによって指定された時間を使用します。この状況は、プロキシと同じです。

[JDK-8278067](#) (JDK バグシステム) を参照してください。

### デフォルトの PKCS #12 MAC アルゴリズムを更新

Red Hat build of OpenJDK 11.0.17 は、PKCS #12 キーストアのデフォルトのメッセージ認証コード (MAC) アルゴリズムを更新して、**SHA-1** 関数ではなく **SHA-256** 暗号化ハッシュ関数を使用します。**SHA-256** 関数は、データを保護するためのより強力な方法を提供します。

この更新は、**keystore.pkcs12.macAlgorithm** および **keystore.pkcs12.maclerationCount** システムプロパティで確認できます。

この更新された MAC アルゴリズムでキーストアを作成し、そのキーストアを Red Hat build of OpenJDK 11.0.12 より前のバージョンの Red Hat build of OpenJDK で使用しようとする  
と、**java.security.NoSuchAlgorithmException** メッセージが表示されます。

Red Hat build of OpenJDK 11.0.12 より前のバージョンの Red Hat build of OpenJDK で以前のキーストアを使用するには、**keystore.pkcs12.legacy** システムプロパティを **true** に設定して、MAC アルゴリズムを元に戻します。

[JDK-8267880](#) (JDK Bug System) を参照してください。

### 非推奨および削除された機能

次のリリースノートで、Red Hat build of OpenJDK 11.0.17 リリースで非推奨または削除された既存の機能を確認してください。

#### 非推奨の Kerberos 暗号化タイプ

Red Hat build of OpenJDK 11.0.17 は、**des3-hmac-sha1** および **rc4-hmac** Kerberos 暗号化タイプを廃止します。デフォルトでは、Red Hat build of OpenJDK 11.0.17 はこれらの暗号化タイプを無効にしますが、次のアクションを完了して有効にすることができます。

- **krb5.conf** 設定ファイルで、**allow\_weak\_crypto** タブを **true** に設定します。この設定により、**des-cbc-crc** や **des-cbc-md5** などの他の暗号化タイプも有効になります。



### 警告

この設定を適用する前に、Kerberos の認証メカニズムに弱い暗号化アルゴリズムを導入するなど、これらの弱い Kerberos 暗号化タイプをすべて有効にするリスクを考慮してください。

次の **krb5.conf** 設定ファイルの設定のいずれかに暗号化タイプを明示的にリストすることで、弱い暗号化タイプのサブセットを無効にすることができます。

- **default\_tkt\_enctypes**
- **default\_tgs\_enctypes**
- **permitted\_enctypes**

[JDK-8139348](#) (JDK Bug System) を参照してください。



## 第4章 このリリースに関連するアドバイザリー

次のアドバイザリーは、このリリースに含まれるバグ修正および CVE 修正に対して発行されています。

- [RHSA-2022:7008](#)
- [RHSA-2022:7009](#)
- [RHSA-2022:7010](#)
- [RHSA-2022:7011](#)
- [RHSA-2022:7012](#)
- [RHSA-2022:7013](#)
- [RHSA-2022:7052](#)
- [RHSA-2022:7054](#)

改訂日時: 2024-05-10