Red Hat build of OpenJDK 11

Red Hat build of OpenJDK 21.0.3 のリリースノート

Last Updated: 2024-05-10

Red Hat build of OpenJDK 11 Red Hat build of OpenJDK 21.0.3 のリリースノート

法律上の通知

概要

本書では、Red Hat build of OpenJDK 11 の新機能の概要と、考えられる既知の問題と、その回避策を説明します。

目次

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 は、Red Hat build of OpenJDK 8u と Red Hat build of OpenJDK 11u の 2 つのバージョンで利用できます。

Red Hat ビルドの 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。

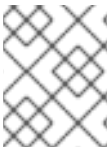
多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティー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 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ー (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ー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

3.1. 新機能および機能拡張

本項では、本リリースで導入された新機能を説明します。また、既存の機能の変更に関する情報も含まれます。

注記

その他の変更点やセキュリティー修正については、https://mail.openjdk.java.net/pipermail/jdk-updates-dev/2021-July/006954.html を参照してください。

3.1.1. PKCS12 キーストア生成のカスタマイズを追加

ユーザーが PKCS #12 キーストアの生成をカスタマイズできるようにする新しいシステムおよびセキュリティープロパティーが追加されました。これには、鍵の保護、証明書保護、および MacData のアルゴリズムとパラメーターが含まれます。**java.security** ファイルの "PKCS12 KeyStore properties" セクションで、これらのプロパティーの詳細な説明および可能な値を見つけます。

また、SunJCE プロバイダーに、以下の SHA-2 ベースの HmacPBE アルゴリズムのサポートを追加しました。

- HmacPBESHA224
- HmacPBESHA256
- HmacPBESHA384
- HmacPBESHA512
- HmacPBESHA512/224
- HmacPBESHA512/256

詳細は、JDK-8215293 を参照してください。

3.1.2. 1024 ビットキーのある root 証明書を削除

1024 ビット RSA 公開鍵を持つ以下のルート証明書が **cacerts** キーストアから削除されました。

- エイリアス名: thawtepremiumserverca [jdk]
  識別名: EMAILADDRESS=premium-server@thawte.com, CN=Thawte Premium Server CA, OU=Certification Services Division, O=Thawte Consulting cc, L=Cape Town, ST=Western Cape, C=ZA
- エイリアス名: verisignclass2g2ca [jdk]
  識別名: OU=VeriSign Trust Network, OU="(c) 1998 VeriSign, Inc. - For authorized use only", OU=Class 2 Public Primary Certification Authority - G2, O="VeriSign, Inc.", C=US
- エイリアス名: verisignclass3ca [jdk]
  識別名: OU=Class 3 Public Primary Certification Authority, O="VeriSign, Inc.", C=US
- エイリアス名: verisignclass3g2ca [jdk]

識別名: OU=VeriSign Trust Network, OU="(c) 1998 VeriSign, Inc. - For authorized use only", OU=Class 3 Public Primary Certification Authority - G2, O="VeriSign, Inc.", C=US

- エイリアス名: verisigntsaca [jdk]
  識別名: CN=Thawte Timestamping CA, OU=Thawte Certification, O=Thawte, L=Durbanville, ST=Western Cape, C=ZA

詳細は、JDK-8256902 を参照してください。

3.1.3. Telia company の Sonera Class2 CA 証明書を削除

以下のルート証明書は **cacerts** トラストストアから削除されました。

- エイリアス名: soneraclass2ca
  識別名: CN=Sonera Class2 CA, O=Sonera, C=FI

詳細は、JDK-8261361 を参照してください。

3.1.4. デフォルトの PKCS12 暗号化と MAC アルゴリズムのアップグレード

PKCS #12 キーストアで使用されるデフォルトの暗号化および MAC アルゴリズムを更新。AES-256 および SHA-256 に基づいた新しいアルゴリズムは、RC2、DESede、および SHA-1 をベースとする古いアルゴリズムよりも強力です。詳細は、**java.security** ファイルの **keystore.pkcs12** で始まるセキュリティープロパティーを参照してください。

互換性のために **keystore.pkcs12.legacy** という名前の新しいシステムプロパティーを定義しました。アルゴリズムを元に戻して、古い弱いアルゴリズムを使用します。このプロパティーには値が定義されていません。

詳細は、JDK-8242069 を参照してください。

3.1.5. TLS Application-Layer Protocol Negotiation (ALPN) 値のエンコーディングが改善されました。

SunJSSE プロバイダーは特定の TLS ALPN 値の読み取りまたは書き込みができません。これは、API インターフェイスとして String を選択することが原因で、U+00007F (7-bit ASCII) より大きい文字をマルチバイトアレイに変換する UTF-8 文字セットでは、文書化されていない内部使用を使用しています。

ALPN 値は、ピアが必要とするネットワークバイト表現を使用して表現されるようになりました。この場合、標準の 7 ビットの ASCII ベースの文字文字列の修正は必要ありません。ただし、SunJSSE は、8 ビット ISO_8859_1/LATIN-1 文字として文字列の文字をエンコード/デコードします。そのため、UTF-8 でエンコードした U+000007F 文字を使用するアプリケーションは、UTF-8 変換を実行するために変更する必要がある場合があります。または、Java セキュリティープロパティー **jdk.tls.alpnCharset** を "UTF-8" に設定して動作を元に戻すことができます。

詳細は、JDK-8257548 を参照してください。

3.1.6. certificate_authorities 拡張のサポートを追加

certificate_authorities 拡張は、TLS 1.3 で導入されたオプションの拡張機能です。これは、認証局 (CA)、エンドポイントサポートを示し、受信エンドポイントによって証明書の選択をガイドするために使用されます。

この Red Hat build of OpenJDK リリースは、クライアントとサーバーの両方で TLS 1.3 の **certificate_authorities** 拡張をサポートします。この拡張はクライアント証明書の選択には常に存在しますが、サーバー証明書の選択にはオプションになります。

アプリケーションは、**jdk.tls.client.enableCAExtension** システムプロパティーを **true** に設定すると、サーバー証明書の選択にこの拡張を有効にできます。プロパティーのデフォルト値は **false** です。

注記

クライアントが拡張のサイズ制限よりも多くの CA を信頼する場合 (2^16 バイト未満)、拡張は有効ではありません。また、一部のサーバー実装では、ハンドシェイクメッセージを 2^14 バイトを超えることができません。そのため、**jdk.tls.client.enableCAExtension** が **true** に設定され、クライアントがサーバー実装の制限よりも多くの CA を信頼すると相互運用性の問題が発生する可能性があります。

詳細は、JDK-8244460 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

以下のアドバイザリーは、本リリースに含まれるバグ修正および CVE の修正に発行されています。

- RHEA-2021:2761-04
- RHEA-2021:2762-03
- RHEA-2021:2753-02
- RHSA-2021:2784-02
- RHSA-2021:2783-02
- RHSA-2021:2782-02
- RHSA-2021:2781-02
- RHSA-2021:2780-01
- RHSA-2021:2779-01

改訂日時: 2024-05-10