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 Container 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。

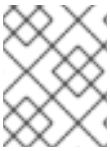
多様性を受け入れるオープンソースの強化

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/2020-October/004007.html を参照してください。

3.1.1. MS950 charset エンコーダーの変換テーブルを修正

一方向のバイトから文字へのマッピングの一部は、Unicode Consortium が指定する推奨マッピングに合わせて調整されています。

詳細は、JDK-8240196 を参照してください。

3.1.2. 外部 FIPS モジュールがセキュリティーモジュールデータベースに存在する場合に NSS を使用した SunPKCS11 の初期化を可能に

Security Modules Database (NSSDB) で FIPS 対応の外部モジュールが設定されている場合、NSS を使用したSunPKCS11 セキュリティープロバイダーの初期化が可能になりました。この変更以前に、SunPKCS11 プロバイダーは、そのようなライブラリーが FIPS モード以外で NSS 用に設定されている場合には、"FIPS flag set for non-internal module" というメッセージとともに RuntimeException を出力していました。

今回の変更により、OpenJDK は、システム全体の FIPS ポリシーが有効になっている場合に、GNU/Linux オペレーティングシステムの最近の NSS リリースで適切に動作するようになりました。

詳細は、JDK-8240191 を参照してください。

3.1.3. 英語と他のロケールとの間でローカライズされたタイムゾーンの名前が異なる

CLDR ロケールプロバイダーによって提供される英語のタイムゾーン名は、COMPAT プロバイダーから置換されるのではなく、CLDR 仕様に従って正しく合成されるようになりました。

たとえば、SHORT スタイル名は、LONG スタイル名の合成略語ではなくなり、代わりに GMT オフセット形式を生成します。

詳細は、JDK-8238914 を参照してください。

3.1.4. コンテナー内の OperatingSystemMXBean メソッドがコンテナー固有のデータを返す

コンテナーまたはその他の仮想化オペレーティング環境で実行する場合、次の **OperatingSystemMXBean** メソッドはコンテナー固有の情報を返します (利用可能な場合)。それ以外の場合は、次のホスト固有のデータを返します。

- getFreePhysicalMemorySize()

- getTotalPhysicalMemorySize()
- getFreeSwapSpaceSize()
- getTotalSwapSpaceSize()
- getSystemCpuLoad()

詳細は、JDK-8236876 を参照してください。

3.1.5. 追加された entrust ルート証明機関 - G4 証明書

entrust ルート証明書が cacerts トラストストアに追加されました。

- エイリアス名: entrustrootcag4
  Distinguished Name: CN=Entrust Root Certification Authority - G4, OU="(c) 2015 Entrust, Inc. - for authorized use only", OU=See www.entrust.net/legal-terms, O="Entrust, Inc.", C=US

詳細は、JDK-8250756 を参照してください。

3.1.6. 3 つの SSL Corporation ルート CA 証明書を追加

以下のルート証明書が、SSL Corporation の cacerts トラストストアに追加されました。

- エイリアス名: sslrootrsaca
  Distinguished Name: CN=SSL.com Root Certification Authority RSA, O=SSL Corporation, L=Houston, ST=Texas, C=US
- エイリアス名: sslrootevrsaca
  Distinguished Name: CN=SSL.com EV Root Certification Authority RSA R2, O=SSL Corporation, L=Houston, ST=Texas, C=US
- エイリアス名: sslrooteccca
  Distinguished Name: CN=SSL.com Root Certification Authority ECC, O=SSL Corporation, L=Houston, ST=Texas, C=US

詳細は、JDK-8250860 を参照してください。

3.1.7. 脆弱なアルゴリズムが使用されている場合に、そのアルゴリズムを制限する前にユーザーに警告するようにツールを更新

keytool および **jarsigner** ツールが更新されたため、脆弱な暗号化アルゴリズムが使用されていることをユーザーに警告してから無効にするようになりました。ツールは、SHA-1 ハッシュアルゴリズムと 1024 ビット RSA/DSA キーに対して警告を発行します。

詳細は、JDK-8244286 を参照してください。

3.1.8. TLS 署名方式を設定するための新しいシステムプロパティー

Red Hat build of OpenJDK で TLS 署名スキームをカスタマイズするために、2 つの新しいシステムプロパティーが追加されました。**jdk.tls.client.SignatureSchemes** が TLS クライアント側に追加され、**jdk.tls.server.SignatureSchemes** がサーバー側に追加されました。

各システムプロパティーには、TLS 接続に使用できる署名方式を指定する、サポート対象の署名方式名 (コンマ区切りリスト) が含まれています。

名前は、**Java Security Standard Algorithm Names Specification**の Signature Schemes セクションで説明されています。

詳細は、JDK-8242147 を参照してください。

3.1.9. krb5.conf での正規化のサポート

krb5.conf ファイル の canonicalize フラグが、JDK Kerberos 実装でサポートされるようになりました。**true** に設定すると、KDC サービス (AS プロトコル) への TGT 要求でクライアントによって RFC 6806 名前の正規化が要求されます。それ以外の場合、デフォルトでは要求されません。

新しいデフォルトの動作は、KDC サービスへの TGT 要求でクライアントが常に名前の正規化を要求していた以前のリリースとは異なります (**sun.security.krb5.disableReferrals** システムまたはセキュリティープロパティーで RFC 6806[1] のサポートが明示的に無効にされていない場合を想定)。

詳細は、JDK-8242059 を参照してください。

3.2. 非推奨の機能

3.2.1. デフォルトで無効になっている TLS、CertPath、および署名付き JAR の弱い名前付き曲線

弱い名前付き曲線は、次の **disabledAlgorithms** セキュリティープロパティーに追加することで、デフォルトで無効になります。

- jdk.tls.disabledAlgorithms
- jdk.certpath.disabledAlgorithms
- jdk.jar.disabledAlgorithms

Red Hat は、アップストリームが提供する曲線の多くを常に削除してきたため、このリリースで無効になっている唯一の曲線は次のとおりです。

- secp256k1

次の曲線は引き続き有効です。

- secp256r1
- secp384r1
- secp521r1
- X25519
- X448

多数の弱い名前付き曲線を無効にする必要がある場合、個々の名前付き曲線をそれぞれの **disabledAlgorithms** プロパティーに追加するのは非常に困難です。作業を減らすために、新しいセキュリティープロパティー **jdk.disabled.namedCurves** が実装され、すべての **disabledAlgorithms** プロパティーに共通の名前付き曲線を一覧表示できます。**disabledAlgorithms** プロパティーで新しいプロパティーを使用するには、完全なプロパティー名の前にキーワード **include** を付けます。ユーザーは、この新しいプロパティーとは別に、個別の名前付き曲線を **disabledAlgorithms** プロパティーに引き続き追加できます。**disabledAlgorithms** プロパティーに他のプロパティーを含めることはできません。

名前付き曲線を復元するには、特定のまたはすべての **disabledAlgorithms** セキュリティープロパティーから、**include jdk.disabled.namedCurves** を削除します。1 つまたは複数の曲線を復元するには、特定の名前付き曲線を **jdk.disabled.namedCurves** プロパティーから削除します。

詳細は、JDK-8236730 を参照してください。

3.2.2. tzdata2020b の一環で "US/Pacific-New" ゾーン名を削除

次の JDK の tzdata2020b への更新では、長い間使用されていなかったファイル pacificnew と systemv が削除されました。その結果、pacificnew データファイルで宣言された "US/Pacific-New" ゾーン名は使用できなくなりました。

更新に関する情報は、https://mm.icann.org/pipermail/tz-announce/2020-October/000059.html で確認できます。

詳細は、JDK-8254177 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。

- RHSA-2020:4316
- RHSA-2020:4305
- RHSA-2020:4306
- RHSA-2020:4307

改訂日時: 2024-05-10