Red Hat build of OpenJDK 11

Release notes for Red Hat build of OpenJDK 11.0.12

Last Updated: 2024-05-09

Red Hat build of OpenJDK 11 Release notes for Red Hat build of OpenJDK 11.0.12

Legal Notice

Abstract

This document provides an overview of new features in Red Hat build of OpenJDK 11, as well as a list of potential known issues and possible workarounds.

Table of Contents

PREFACE

Open Java Development Kit (OpenJDK) is a free and open source implementation of the Java Platform, Standard Edition (Java SE). The Red Hat build of OpenJDK is available in two versions, Red Hat build of OpenJDK 8u and Red Hat build of OpenJDK 11u.

Packages for the Red Hat build of OpenJDK are made available on Red Hat Enterprise Linux and Microsoft Windows and shipped as a JDK and JRE in the Red Hat Ecosystem Catalog.

PROVIDING FEEDBACK ON RED HAT BUILD OF OPENJDK DOCUMENTATION

To report an error or to improve our documentation, log in to your Red Hat Jira account and submit an issue. If you do not have a Red Hat Jira account, then you will be prompted to create an account.

Procedure

1. Click the following link to **create a ticket**.
2. Enter a brief description of the issue in the **Summary**.
3. Provide a detailed description of the issue or enhancement in the **Description**. Include a URL to where the issue occurs in the documentation.
4. Clicking **Submit** creates and routes the issue to the appropriate documentation team.

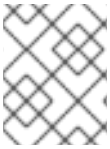
MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright's message.

CHAPTER 1. SUPPORT POLICY FOR RED HAT BUILD OF OPENJDK

Red Hat will support select major versions of Red Hat build of OpenJDK in its products. For consistency, these are the same versions that Oracle designates as long-term support (LTS) for the Oracle JDK.

A major version of Red Hat build of OpenJDK will be supported for a minimum of six years from the time that version is first introduced. For more information, see the OpenJDK Life Cycle and Support Policy .



NOTE

RHEL 6 reached the end of life in November 2020. Because of this, Red Hat build of OpenJDK is not supporting RHEL 6 as a supported configuration.

CHAPTER 2. DIFFERENCES FROM UPSTREAM OPENJDK 11

Red Hat build of OpenJDK in Red Hat Enterprise Linux (RHEL) contains a number of structural changes from the upstream distribution of OpenJDK. The Microsoft Windows version of Red Hat build of OpenJDK attempts to follow RHEL updates as closely as possible.

The following list details the most notable Red Hat build of OpenJDK 11 changes:

- FIPS support. Red Hat build of OpenJDK 11 automatically detects whether RHEL is in FIPS mode and automatically configures Red Hat build of OpenJDK 11 to operate in that mode. This change does not apply to Red Hat build of OpenJDK builds for Microsoft Windows.
- Cryptographic policy support. Red Hat build of OpenJDK 11 obtains the list of enabled cryptographic algorithms and key size constraints from RHEL. These configuration components are used by the Transport Layer Security (TLS) encryption protocol, the certificate path validation, and any signed JARs. You can set different security profiles to balance safety and compatibility. This change does not apply to Red Hat build of OpenJDK builds for Microsoft Windows.
- Red Hat build of OpenJDK on RHEL dynamically links against native libraries such as **zlib** for archive format support and **libjpeg-turbo**, **libpng**, and **giflib** for image support. RHEL also dynamically links against **Harfbuzz** and **Freetype** for font rendering and management.
- The **src.zip** file includes the source for all the JAR libraries shipped with Red Hat build of OpenJDK.
- Red Hat build of OpenJDK on RHEL uses system-wide timezone data files as a source for timezone information.
- Red Hat build of OpenJDK on RHEL uses system-wide CA certificates.
- Red Hat build of OpenJDK on Microsoft Windows includes the latest available timezone data from RHEL.
- Red Hat build of OpenJDK on Microsoft Windows uses the latest available CA certificate from RHEL.

Additional resources

- For more information about detecting if a system is in FIPS mode, see the Improve system FIPS detection example on the Red Hat RHEL Planning Jira.
- For more information about cryptographic policies, see Using system-wide cryptographic policies.

CHAPTER 3. RED HAT BUILD OF OPENJDK FEATURES

3.1. NEW FEATURES AND ENHANCEMENTS

This section describes the new features introduced in this release. It also contains information about changes in the existing features.

NOTE

For all the other changes and security fixes, see https://mail.openjdk.java.net/pipermail/jdk-updates-dev/2021-July/006954.html.

3.1.1. Added customize PKCS12 keystore generation

Added new system and security properties for enabling users to customize the generation of PKCS #12 keystores. This includes algorithms and parameters for key protection, certificate protection, and MacData. Find the detailed explanation and possible values for these properties in the "PKCS12 KeyStore properties" section of the **java.security** file.

Also, added support for the following SHA-2 based HmacPBE algorithms to the SunJCE provider:

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

For more information, see JDK-8215293.

3.1.2. Removed root certificates with 1024-bit keys

The following root certificates with weak 1024-bit RSA public keys have been removed from the **cacerts** keystore:

- Alias name: thawtepremiumserverca [jdk]
  Distinguished name: 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
- Alias name: verisignclass2g2ca [jdk]
  Distinguished name: 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
- Alias name: verisignclass3ca [jdk]
  Distinguished name: OU=Class 3 Public Primary Certification Authority, O="VeriSign, Inc.", C=US
- Alias name: verisignclass3g2ca [jdk]

Distinguished name: 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

- Alias name: verisigntsaca [jdk]
  Distinguished name: CN=Thawte Timestamping CA, OU=Thawte Certification, O=Thawte, L=Durbanville, ST=Western Cape, C=ZA

For more information, see JDK-8256902.

3.1.3. Removed Telia company's Sonera Class2 CA certificate

The following root certificate have been removed from the **cacerts** truststore:

- Alias Name: soneraclass2ca
  Distinguished Name: CN=Sonera Class2 CA, O=Sonera, C=FI

For more information, see JDK-8261361.

3.1.4. Upgraded the default PKCS12 encryption and MAC algorithms

Updated default encryption and MAC algorithms used in a PKCS #12 keystore. The new algorithms based on AES-256 and SHA-256 are stronger than the old algorithms that were based on RC2, DESede, and SHA-1. See the security properties starting with **keystore.pkcs12** in the **java.security** file for more details.

Defined a new system property named **keystore.pkcs12.legacy** for compatibility. It will revert the algorithms to use the older, weaker algorithms. There is no value defined for this property.

For more information, see JDK-8242069.

3.1.5. Improved encoding of TLS Application-Layer Protocol Negotiation (ALPN) values

SunJSSE providers cannot read or write certain TLS ALPN values. This is due to the choice of Strings as the API interface and the undocumented internal use of the UTF-8 character set that converts characters larger than U+00007F (7-bit ASCII) into multi-byte arrays.

ALPN values are now represented using the network byte representation expected by the peer, which should require no modification for standard 7-bit ASCII-based character Strings. However, SunJSSE now encodes/decodes string characters as 8-bit ISO_8859_1/LATIN-1 characters. his means the applications that are using characters above U+000007F encoded with UTF-8 may need to be modified to perform the UTF-8 conversion, or you can set the Java security property **jdk.tls.alpnCharset** to "UTF-8" to revert the behavior.

For more information, see JDK-8257548.

3.1.6. Added support for certificate_authorities extension

The **certificate_authorities** extension is an optional extension introduced in TLS 1.3. It indicates certificate authorities (CAs), the endpoint support and used by the receiving endpoint to guide certificate selection.

This Red Hat build of OpenJDK release supports the **certificate_authorities** extension for TLS 1.3 on both the client and the server sides. This extension is always present for client certificate selection, while it is optional for server certificate selection.

Applications can enable this extension for server certificate selection by setting the **jdk.tls.client.enableCAExtension** system property to **true**. The default value of the property is **false**.

NOTE

If the client trusts more CAs than the size limit of the extension (less than 2^16 bytes), the extension is not enabled. Also, some server implementations do not allow handshake messages to exceed 2^14 bytes. Consequently, there may be interoperability issues when **jdk.tls.client.enableCAExtension** is set to **true** and the client trusts more CAs than the server implementation limit.

For more information, see JDK-8244460.

CHAPTER 4. ADVISORIES RELATED TO THIS RELEASE

The following advisories have been issued to bugfixes and CVE fixes included in this release.

- 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

Revised on 2024-05-09 16:46:34 UTC