



OpenJDK 8

Release notes for OpenJDK 8.0.272

Legal Notice

Copyright © 2021 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.

Abstract

This document provides an overview of new features in OpenJDK 8, as well as a list of potential known issues and possible workarounds.

Table of Contents

PREFACE	3
MAKING OPEN SOURCE MORE INCLUSIVE	4
CHAPTER 1. SUPPORT POLICY	5
CHAPTER 2. DIFFERENCES FROM UPSTREAM OPENJDK 8	6
CHAPTER 3. OPENJDK FEATURES	7
3.1. NEW FEATURES AND ENHANCEMENTS	7
3.1.1. Allow SunPKCS11 initialization with NSS when external FIPS modules are present in the Security Modules Database	7
3.1.2. OperatingSystemMXBean methods inside a container return container specific data	7
3.1.3. Added entrust root certification authority - G4 certificate	7
3.1.4. Added 3 SSL corporation root CA certificate	8
3.1.5. SunPKCS11 provider upgraded with support for PKCS#11 v2.40	8
3.1.6. Support for canonicalize in krb5.conf	8
3.1.7. Updated xmldsig implementation to Apache Santuario 2.1.1	8
3.1.8. New OpenJDK-specific JDK 8 updates system property to fallback to legacy Base64 encoding format	8
3.1.9. Added additional Windows artifacts	9
3.2. DEPRECATED FEATURE	10
3.2.1. US/Pacific-New Zone name removed as part of tzdata2020b	10
CHAPTER 4. ADVISORIES RELATED TO THIS RELEASE	11

PREFACE

OpenJDK (Open Java Development Kit) 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, OpenJDK 8u and 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 Container Catalog.

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

Red Hat will support select major versions of OpenJDK in its products. For consistency, these versions will be the same ones that Oracle designates 'LTS' for the Oracle JDK.

A major version of OpenJDK will be supported for a minimum of six years from the time it is first introduced.

OpenJDK 8 is supported on Microsoft Windows and Red Hat Enterprise Linux until May 2026.



NOTE

RHEL 6 has reached the end of life in November 2020. Due to this, OpenJDK is not supporting RHEL 6 as a supporting configuration.

For more information, see the [OpenJDK Life Cycle and Support Policy](#) .

CHAPTER 2. DIFFERENCES FROM UPSTREAM OPENJDK 8

OpenJDK in Red Hat Enterprise Linux contains a number of structural changes to the upstream distribution of OpenJDK. The Windows version of OpenJDK tries to follow Red Hat Enterprise Linux as closely as possible.

The most notable changes are the following:

- On Red Hat Enterprise Linux, external native libraries are used for archive format support (**zlib**) and image formats (**libjpeg-turbo**, **libpng**, and **giflib**). Red Hat Enterprise Linux 8 uses additional font rendering (**harfbuzz**) external library.
On Microsoft Windows, these libraries are built from the sources of the corresponding Red Hat Enterprise Linux RPMs.
- On Red Hat Enterprise Linux, system-wide timezone data files are used as a source for timezone information.
On Microsoft Windows, the latest available timezone data from Red Hat Enterprise Linux is included.
- On Red Hat Enterprise Linux, system-wide CA certificates are used.
On Microsoft Windows, the latest available CA certificate from Red Hat Enterprise Linux is used.
- The Windows distribution includes the DejaVu set of TrueType Fonts imported from Red Hat Enterprise Linux.
- The **src.zip** file includes the source for all of the JAR libraries shipped with OpenJDK.

CHAPTER 3. 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/jdk8u-dev/2020-October/012817.html>

3.1.1. Allow SunPKCS11 initialization with NSS when external FIPS modules are present in the Security Modules Database

The SunPKCS11 security provider can now be initialized with NSS when FIPS-enabled external modules are configured in the Security Modules Database (NSSDB). Prior to this change, the SunPKCS11 provider would throw a `RuntimeException` with the message: "FIPS flag set for non-internal module" when such a library was configured for NSS in non-FIPS mode.

This change allows the JDK to work properly with recent NSS releases in GNU/Linux operating systems when the system-wide FIPS policy is turned on.

For more information, see [JDK-8240191](#).

3.1.2. `OperatingSystemMXBean` methods inside a container return container specific data

When executing in a container, or other virtualized operating environment, the following **`OperatingSystemMXBean`** methods return container-specific information, if available. Otherwise, they return host-specific data:

- `getFreePhysicalMemorySize()`
- `getTotalPhysicalMemorySize()`
- `getFreeSwapSpaceSize()`
- `getTotalSwapSpaceSize()`
- `getSystemCpuLoad()`

For more information, see [JDK-8236876](#).

3.1.3. Added entrust root certification authority - G4 certificate

The entrust root certificate has been added to the cacerts truststore:

- Alias Name: `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`

For more information, see [JDK-8250756](#).

3.1.4. Added 3 SSL corporation root CA certificate

The following root certificates have been added to the cacerts truststore for the SSL Corporation:

- Alias Name: sslrootsaca
Distinguished Name: CN=SSL.com Root Certification Authority RSA, O=SSL Corporation, L=Houston, ST=Texas, C=US
- Alias Name: sslrootevrsaca
Distinguished Name: CN=SSL.com EV Root Certification Authority RSA R2, O=SSL Corporation, L=Houston, ST=Texas, C=US
- Alias Name: sslrooteccca
Distinguished Name: CN=SSL.com Root Certification Authority ECC, O=SSL Corporation, L=Houston, ST=Texas, C=US

For more information, see [JDK-8250860](#).

3.1.5. SunPKCS11 provider upgraded with support for PKCS#11 v2.40

The SunPKCS11 provider has been updated with support for PKCS#11 v2.40. This version adds support for more algorithms such as the AES/GCM/NoPadding cipher, DSA signatures using SHA-2 family of message digests, and RSASSA-PSS signatures when the corresponding PKCS11 mechanisms are supported by the underlying PKCS11 library.

For more information, see [JDK-8221441](#).

3.1.6. Support for canonicalize in krb5.conf

The 'canonicalize' flag in the [krb5.conf](#) file is now supported by the JDK Kerberos implementation. When set to **true**, [RFC 6806](#) name canonicalization is requested by clients in TGT requests to KDC services (AS protocol). Otherwise, and by default, it is not requested.

The new default behavior is different from previous releases where name canonicalization was always requested by clients in TGT requests to KDC services (provided that support for [RFC 6806](#) was not explicitly disabled with the `sun.security.krb5.disableReferrals` system or security properties).

For more information, see [JDK-8242059](#).

3.1.7. Updated xmldsig implementation to Apache Santuario 2.1.1

The XMLDSig provider implementation in the `java.xml.crypto` module has been updated to version 2.1.1 of Apache Santuario.

New features include:

1. Support for the SHA-224 and SHA-3 DigestMethod algorithms specified in RFC 6931.
2. Support for the HMAC-SHA224, RSA-SHA224, ECDSA-SHA224, and RSASSA-PSS family of SignatureMethod algorithms specified in RFC 6931.

For more information, see [JDK-8202891](#).

3.1.8. New OpenJDK-specific JDK 8 updates system property to fallback to legacy Base64 encoding format

The upgrade to the Apache Santuario libraries (see above) introduced an issue where XML signature using Base64 encoding resulted in appending `` or `` to the encoded output. This behavioural change was made in the Apache Santuario codebase to comply with RFC 2045. The Santuario team has adopted a position of keeping their libraries compliant with RFC 2045.

Earlier versions of OpenJDK 8 using the legacy encoder returns encoded data in a format without `` or ``.

Therefore a new system property, specific to the OpenJDK 8 update stream, **`com.sun.org.apache.xml.internal.security.lineFeedOnly`** is made available to fall back to the legacy Base64 encoded format.

Users can set this flag in one of the following two ways:

- `-Dcom.sun.org.apache.xml.internal.security.lineFeedOnly=true`
- `System.setProperty("com.sun.org.apache.xml.internal.security.lineFeedOnly", "true")`

This new system property is disabled by default. It has no effect on default behaviour nor when **`com.sun.org.apache.xml.internal.security.ignoreLineBreaks`** property is set.

Later JDK family versions will only support the recommended property:

`com.sun.org.apache.xml.internal.security.ignoreLineBreaks`

For more information, see [JDK-8238185](#).

3.1.9. Added additional Windows artifacts

Currently, OpenJDK 8 support the following Windows artifacts:

- OpenJDK 1.8.0.265 JRE for Windows 64 Bit
- OpenJDK 1.8.0.265 JDK for Windows 64 Bit
- OpenJDK 1.8.0.265 JDK for Windows 64 Bit Installer
- OpenJDK 1.8.0.265 Debugging Symbols for Windows 64 Bit

In addition to the existing Windows artifacts, support for the following Windows artifacts is added in this release:

- OpenJDK 1.8.0.272 JRE for Windows 64 Bit (Alternative Toolchain)
- OpenJDK 1.8.0.272 JDK for Windows 64 Bit (Alternative Toolchain)
- OpenJDK 1.8.0.272 JDK for Windows 64 Bit Installer (Alternative Toolchain)
- OpenJDK 1.8.0.72 Debugging Symbols for Windows 64 Bit (Alternative Toolchain)



NOTE

Red Hat version of OpenJDK 8 for Windows is being built using Microsoft Visual Studio 2010 compiler toolchain. This version of compiler toolchain has reached its end of life on 14 July 2020 and is no longer supported by [Microsoft](#). With the release of 8u272 of OpenJDK 8, Red Hat provides two sets of OpenJDK 8 builds for Windows. The first set (with the same naming as before) is built with Visual Studio 2010 compiler toolchain and is expected to be the most compatible with existing Java applications. The second set is marked with "Alternative Toolchain" label and is built with Visual Studio 2017 toolchain. These "Alternative Toolchain" builds have the same level of support as the original ones, but may cause incompatibilities with existing Java applications, especially in the areas of JNI code and in interaction with Windows API.

For more information, see [JDK-8202076](#).

3.2. DEPRECATED FEATURE

3.2.1. US/Pacific-New Zone name removed as part of tzdata2020b

Following JDK's update to tzdata2020b, the long-obsolete files pacificnew and systemv have been removed. As a result, the "US/Pacific-New" zone name declared in the pacificnew data file is no longer available for use.

Information regarding the update can be viewed at <https://mm.icann.org/pipermail/tz-announce/2020-October/000059.html>

For more information, see [JDK-8254177](#).

CHAPTER 4. ADVISORIES RELATED TO THIS RELEASE

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

- [RHSA-2020:4348-02](#)
- [RHBA-2020:59597-02](#)
- [RHSA-2020:4350-02](#)
- [RHSA-2020:4347-02](#)
- [RHSA-2020:4352-02](#)
- [RHSA-2020:4349-02](#)