Red Hat build of OpenJDK 21

Release notes for Red Hat build of OpenJDK 21.0.1

Last Updated: 2023-12-14

Red Hat build of OpenJDK 21 Release notes for Red Hat build of OpenJDK 21.0.1

Legal Notice

Abstract

The Release notes for Red Hat build of OpenJDK 21.0.1 document provides an overview of new features in Red Hat build of OpenJDK 21 and a list of potential known issues and possible workarounds.

Table of Contents

PREFACE

Red Hat build of 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 four versions: 8u, 11u, 17u, and 21u.

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.

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 versions remain similar to Oracle JDK versions that are designated as long-term support (LTS).

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 21

Red Hat build of OpenJDK in Red Hat Enterprise Linux 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 Red Hat Enterprise Linux updates as closely as possible.

The following list details the most notable Red Hat build of OpenJDK 21 changes:

- FIPS support. Red Hat build of OpenJDK 21 automatically detects whether RHEL is in FIPS mode and automatically configures Red Hat build of OpenJDK 21 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 21 obtains the list of enabled cryptographic algorithms and key size constraints from the RHEL system configuration. 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.
- The **src.zip** file includes the source for all of 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 certificates from RHEL.

Additional resources

- See, Improve system FIPS detection (RHEL Planning Jira)
- See, Using system-wide cryptographic policies (RHEL documentation)

CHAPTER 3. DIFFERENCES FROM RED HAT BUILD OF OPENJDK 17

Red Hat build of OpenJDK 21 includes a number of enhancements that are not available in Red Hat build of OpenJDK 17, which is the previous long-term support (LTS) release.

For more information about the additional Java enhancement proposals (JEPs) that Red Hat build of OpenJDK 21 includes, see JEPs in JDK 21 integrated since JDK 17 .

Additional resources

- OpenJDK: JDK 18
- OpenJDK: JDK 19
- OpenJDK: JDK 20
- OpenJDK: JDK 21

CHAPTER 4. DEPRECATED AND UNSUPPORTED RED HAT BUILD OF OPENJDK CAPABILITIES

Ensure that you review the following deprecated and unsupported features before you install Red Hat build of OpenJDK 21:

JDK Mission Control in the Red Hat build of OpenJDK 21 for Windows packages

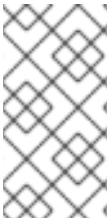
The Windows installer and the zip archive for Red Hat build of OpenJDK 21 no longer provide a distribution of JDK Mission Control (JMC). You can use the Red Hat build of Cryostat to manage JFR recordings for Java applications deployed on cloud platforms such as Red Hat OpenShift. For more information about the removal of JMC, see the Red Hat knowledge base article: Where is JDK Mission Control (JMC) in JDK 21?.

Deprecate Finalization for removal

This release deprecates Finalization for removal in a future release. For more information, see JEP 421: Deprecate Finalization for Removal.

Prepare to disallow the dynamic loading of agents

This release issues warnings when agents are loaded dynamically into a running JVM. The dynamic loading of agents will be disallowed by default in a future release. For more information, see JEP 451: Prepare to Disallow the Dynamic Loading of Agents.



NOTE

Red Hat does not provide builds of OpenJDK with 32-bit support. In OpenJDK 21, Windows 32-bit x86 support is also now deprecated upstream. This feature will be removed in a future release. For more information, see JEP 449: Deprecate the Windows 32-bit x86 Port for Removal.

CHAPTER 5. NEW FEATURES FOR THE RED HAT BUILD OF OPENJDK 21

The initial release of Red Hat build of OpenJDK 21 includes new features that enhance the use of your Java applications.

Red Hat build of OpenJDK 21 includes the following new features:

UTF-8 by default

For more information, see JEP 400: UTF-8 by Default .

Simple web server

For more information, see JEP 408: Simple Web Server .

Code snippets in Java API documentation

For more information, see JEP 413: Code Snippets in Java API Documentation .

Reimplement core reflection with method handles

For more information, see JEP 416: Reimplement Core Reflection with Method Handles .

Internet-address resolution SPI

For more information, see JEP 418: Internet-Address Resolution SPI .

Linux/RISC-V port

For more information, see JEP 422: Linux/RISC-V Port .

Scoped values *(Preview feature)*

For more information, see JEP 429: Scoped Values (Preview) .

String templates *(Preview feature)*

For more information, see JEP 430: String Templates (Preview) .

Sequenced collections

For more information, see JEP 431: Sequenced Collections .

Generational Z Garbage Collector (ZGC)

For more information, see JEP 439: Generational ZGC .

Record patterns

For more information, see JEP 440: Record Patterns .

Pattern matching for switch

For more information, see JEP 441: Pattern Matching for switch .

Foreign function and memory (FFM) API *(Third preview)*

For more information, see JEP 442: Foreign Function & Memory API (Third Preview) .

Unnamed patterns and variables *(Preview feature)*

For more information, see JEP 443: Unnamed Patterns and Variables (Preview) .

Virtual threads

For more information, see JEP 444: Virtual Threads .

Unnamed classes and instance main methods *(Preview feature)*

For more information, see JEP 445: Unnamed Classes and Instance Main Methods (Preview) .

Scoped values *(preview)*

For more information, see JEP 446: Scoped Values (Preview) .

Vector API (sixth incubator)

For more information, see JEP 448: Vector API (Sixth Incubator).

Key encapsulation mechanism API

For more information, see JEP 452: Key Encapsulation Mechanism API.

Structured concurrency *(Preview feature)*

For more information, see JEP 453: Structured Concurrency (Preview).

CHAPTER 6. RED HAT BUILD OF OPENJDK 21.0.1 BUG FIXES

Red Hat build of OpenJDK 21.0.1 includes the following bug fixes.

Fixed **Invalid CEN header** error on valid **.zip** files

Red Hat build of OpenJDK 21.0.0 introduced additional validation checks on the **ZIP64** fields of **.zip** files (JDK-8302483). However, these additional checks caused validation failures on some valid **.zip** files with the following error message: **Invalid CEN header (invalid zip64 extra data field size)**.

To fix this issue, Red Hat build of OpenJDK 21.0.1 supports zero-length headers and the additional padding that some **ZIP64** creation tools produce.

The following third-party tools have also released patches for better adherence to the .ZIP File Format Specification:

- Apache Commons Compress fix for Empty CEN Zip64 Extra Headers fixed in Commons Compress release 1.11
- Apache Ant fix for Empty CEN Zip64 Extra Headers fixed in Ant 1.10.14
- BND issue with writing invalid Extra Headers fixed in BND 5.3

NOTE

The **maven-bundle-plugin** 5.1.5 includes the BND 5.3 patch.

If these improved validation checks cause issues for deployed ZIP or JAR files, check how the file was created and whether patches are available from the generating software to resolve the issue. From Red Hat build of OpenJDK 21.0.0 onward, you can disable these checks by setting the **jdk.util.zip.disableZip64ExtraFieldValidation** system property to **true**.

See JDK-8313765 (JDK Bug System)

Fixed potential deadlock where JVM might hang when using Generational ZGC if a VM handshake stalls on memory

In Red Hat build of OpenJDK 21.0.0, the JVM might hang when all of the following situations occurred at the same time:

- The JVM was running out of heap memory.
- The garbage collector (GC) was just starting a relocation phase to reclaim memory.
- A JVM thread-local handshake was asking to relocate an object.

Red Hat build of OpenJDK 21.0.0 introduces a fix to avoid this potential deadlock.

See JDK-8311981 (JDK Bug System)

Fixed potential exception thrown by **java.util.regex.MatchResult** on Regex patterns containing lookaheads and lookbehinds

In Red Hat build of OpenJDK 21.0.0, the optimization of the **Matcher$ImmutableMatchResult** nested class (JDK-8132995) introduced an unintended regression when using instances returned by **java.util.regex.Matcher.toMatchResult()**. This regression occurred with **java.util.regex.Pattern** classes

containing lookahead and lookbehind assertions that, in turn, contained groups. If these groups were located outside the match, the **java.util.regex.MatchResult** threw a **StringIndexOutOfBoundsException** when accessing these groups.

Red Hat build of OpenJDK 21.0.1 resolves this issue by calculating a minimum start location as part of the match result and by using this minimum start location rather than the location of the first match when constructing **String** objects.

See JDK-8312976 (JDK Bug System)

Certigna root CA certificate added

In Red Hat build of OpenJDK 21.0.1, the **cacerts** truststore includes the Certigna root certificate:

- Name: Certigna (Dhimyotis)
- Alias name: certignarootca
- Distinguished name: CN=Certigna Root CA, OU=0002 48146308100036, O=Dhimyotis, C=FR

See JDK-8314960 (JDK Bug System).

Increased default value of jdk.jar.maxSignatureFileSize system property

Red Hat build of OpenJDK 21.0.0 introduced a **jdk.jar.maxSignatureFileSize** system property for configuring the maximum number of bytes that are allowed for the signature-related files in a Java archive (JAR) file (JDK-8300596). By default, the **jdk.jar.maxSignatureFileSize** property was set to 8000000 bytes (8 MB), which was too small for some JAR files.

Red Hat build of OpenJDK 21.0.1 increases the default value of the **jdk.jar.maxSignatureFileSize** property to 16000000 bytes (16 MB).

See JDK-8312489 (JDK Bug System).

CHAPTER 7. ADVISORIES RELATED TO THIS RELEASE

The following advisories are issued to document bug fixes and CVE fixes included in this release:

- RHSA-2023:6738
- RHSA-2023:6887
- RHEA-2023:6888
- RHSA-2023:6889
- RHBA-2023:6897
- RHBA-2023:6898

Revised on 2023-12-14 20:38:21 UTC