



Red Hat build of OpenJDK 17

Release notes for Red Hat build of OpenJDK 17.0.12

Red Hat build of OpenJDK 17 Release notes for Red Hat build of OpenJDK 17.0.12

Legal Notice

Copyright © 2024 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

The Release notes for Red Hat build of OpenJDK 17.0.12 document provides an overview of new features in Red Hat build of OpenJDK 17 and a list of potential known issues and possible workarounds.

Table of Contents

PREFACE	3
PROVIDING FEEDBACK ON RED HAT BUILD OF OPENJDK DOCUMENTATION	4
MAKING OPEN SOURCE MORE INCLUSIVE	5
CHAPTER 1. SUPPORT POLICY FOR RED HAT BUILD OF OPENJDK	6
CHAPTER 2. DIFFERENCES FROM UPSTREAM OPENJDK 17	7
CHAPTER 3. PLANNED CHANGES TO NAMING CONVENTION FOR WINDOWS BUILD ARTIFACTS	8
CHAPTER 4. RED HAT BUILD OF OPENJDK FEATURES	9
Red Hat build of OpenJDK enhancements	9
Fallback option for POST-only OCSP requests	9
DTLS 1.0 is disabled by default	9
RPATH preferred over RUNPATH for \$ORIGIN runtime search paths in internal JDK binaries	9
TrimNativeHeapInterval option available as a product switch	10
-XshowSettings launcher option includes a security category	10
GlobalSign R46 and E46 root certificates added	10
Fix for long garbage collection pauses due to imbalanced iteration during the Code Root Scan phase	11
Change in behavior for AWT headless mode detection on Windows	11
CHAPTER 5. ADVISORIES RELATED TO THIS RELEASE	12

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 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.

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 **Create** creates and routes the issue to the appropriate documentation team.

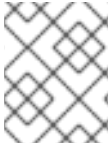
MAKING OPEN SOURCE MORE INCLUSIVE

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 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 17

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 17 changes:

- FIPS support. Red Hat build of OpenJDK 17 automatically detects whether RHEL is in FIPS mode and automatically configures Red Hat build of OpenJDK 17 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 17 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.
- 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. 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 certificate from RHEL.

Additional resources

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

CHAPTER 3. PLANNED CHANGES TO NAMING CONVENTION FOR WINDOWS BUILD ARTIFACTS

From October 2024 onward, Red Hat plans to introduce naming changes for some files that are distributed as part of Red Hat build of OpenJDK releases for Windows Server platforms.

These file naming changes will affect both the **.zip** archive and **.msi** installers that Red Hat provides for the JDK, JRE and **debuginfo** packages for Red Hat build of OpenJDK versions 8, 11, and 17.

The aim of this change is to adopt a common naming convention that is consistent across all versions of OpenJDK that Red Hat supports. Red Hat build of OpenJDK versions 8, 11, and 17 will be aligned with the naming convention that Red Hat has already adopted for Red Hat build of OpenJDK 21. This means that Red Hat build of OpenJDK 21 will not require any naming changes.

These planned changes do not affect the files for the Linux portable builds of any Red Hat build of OpenJDK version.

Red Hat build of OpenJDK 17.0.12 is the last release where Red Hat plans to use the old naming convention for Windows artifacts. The following list provides an example of how the planned naming changes will affect each file for future releases of Red Hat build of OpenJDK 17:

MSI installer for JDK package

- *Old file name:* **java-17-openjdk-`<version>`.win.x86_64.msi**
- *New file name:* **java-17-openjdk-`<version>`.win.jdk.x86_64.msi**

.zip archive for JDK package

- *Old file name:* **java-17-openjdk-`<version>`.win.x86_64.zip**
- *New file name:* **java-17-openjdk-`<version>`.win.jdk.x86_64.zip**

MSI installer for JRE package

- *Old file name:* **java-17-openjdk-`<version>`.jre.win.x86_64.msi**
- *New file name:* **java-17-openjdk-`<version>`.win.jre.x86_64.msi**

.zip archive for for JRE package

- *Old file name:* **java-17-openjdk-`<version>`.jre.win.x86_64.zip**
- *New file name:* **java-17-openjdk-`<version>`.win.jre.x86_64.zip**

.zip archive for debuginfo package

- *Old file name:* **java-17-openjdk-`<version>`.win.x86_64.debuginfo.zip**
- *New file name:* **java-17-openjdk-`<version>`.win.debuginfo.x86_64.zip**

CHAPTER 4. RED HAT BUILD OF OPENJDK FEATURES

The latest Red Hat build of OpenJDK 17 release might include new features. Additionally, the latest release might enhance, deprecate, or remove features that originated from previous Red Hat build of OpenJDK 17 releases.



NOTE

For all the other changes and security fixes, see [OpenJDK 17.0.12 Released](#).

Red Hat build of OpenJDK enhancements

Red Hat build of OpenJDK 17 provides enhancements to features originally created in previous releases of Red Hat build of OpenJDK.

Fallback option for **POST**-only OCSP requests

[JDK-8175903](#), which was introduced in Red Hat build of OpenJDK 17, added support for using the HTTP **GET** method for Online Certificate Status Protocol (OCSP) requests. This feature was enabled unconditionally for small requests.

The Internet Engineering Task Force (IETF) [RFC 5019](#) and [RFC 6960](#) explicitly allow and recommend the use of HTTP **GET** requests. However, some OCSP responders do not work well with these types of requests.

Red Hat build of OpenJDK 17.0.12 introduces a JDK system property, **com.sun.security.ocsp.useget**. By default, this property is set to **true**, which retains the current behavior of using **GET** requests for small requests. If this property is set to **false**, only HTTP **POST** requests are used, regardless of size.



NOTE

This fallback option for **POST**-only OCSP requests is a non-standard feature, which might be removed in a future release if the use of HTTP **GET** requests with OCSP responders no longer causes any issues.

See [JDK-8328638 \(JDK Bug System\)](#).

DTLS 1.0 is disabled by default

OpenJDK 9 introduced support for both version 1.0 and version 1.2 of the Datagram Transport Layer Security (DTLS) protocol ([JEP-219](#)). DTLSv1.0, which is based on TLS 1.1, is no longer recommended for use, because this protocol is considered weak and insecure by modern standards. In Red Hat build of OpenJDK 17.0.12, if you attempt to use DTLSv1.0, the JDK throws an **SSLHandshakeException** by default.

If you want to continue using DTLSv1.0, you can remove **DTLSv1.0** from the **jdk.tls.disabledAlgorithms** system property either by modifying the **java.security** configuration file or by using the **java.security.properties** system property.



NOTE

Continued use of DTLSv1.0 is not recommended and is at the user's own risk.

See [JDK-8256660 \(JDK Bug System\)](#).

RPATH preferred over RUNPATH for \$ORIGIN runtime search paths in internal JDK binaries

Native executables and libraries in the JDK use embedded runtime search paths (rpaths) to locate required internal JDK native libraries. On Linux systems, binaries can specify these search paths by using either **DT_RPATH** or **DT_RUNPATH**.

- If a binary specifies search paths by using **DT_RPATH**, these paths are searched *before* any paths that are specified in the **LD_LIBRARY_PATH** environment variable.
- If a binary specifies search paths by using **DT_RUNPATH**, these paths are searched only *after* paths that are specified in **LD_LIBRARY_PATH**. This means that the use of **DT_RUNPATH** can allow JDK internal libraries to be overridden by any libraries of the same name that are specified in **LD_LIBRARY_PATH**, which is undesirable from a security perspective.

In earlier releases, the type of runtime search path used was based on the default search path for the dynamic linker. In Red Hat build of OpenJDK 17.0.12, to ensure that **DT_RPATH** is used, the **--disable-new-dtags** option is explicitly passed to the linker.

See [JDK-8326891 \(JDK Bug System\)](#).

TrimNativeHeapInterval option available as a product switch

Red Hat build of OpenJDK 17.0.12 provides the **-XX:TrimNativeHeapInterval=ms** option as an official product switch. This enhancement enables the JVM to trim the native heap at specified intervals (in milliseconds) on supported platforms. Currently, the only supported platform for this enhancement is Linux with **glibc**.

You can disable trimming by setting **TrimNativeHeapInterval=0**. The trimming feature is disabled by default.

See [JDK-8325496 \(JDK Bug System\)](#).

-XshowSettings launcher option includes a security category

In Red Hat build of OpenJDK 17.0.12, the **-XshowSettings** launcher option includes a security category, which allows the following arguments to be passed:

Argument	Details
-XshowSettings:security or -XshowSettings:security:all	Show all security settings and continue.
-XshowSettings:security:properties	Show security properties and continue.
-XshowSettings:security:providers	Show static security provider settings and continue.
-XshowSettings:security:tls	Show TLS-related security settings and continue.

If third-party security providers are included in the application class path or module path, and configured in the **java.security** file, the output includes these third-party security providers.

See [JDK-8281658 \(JDK Bug System\)](#).

GlobalSign R46 and E46 root certificates added

In Red Hat build of OpenJDK 17.0.12, the **cacerts** truststore includes two GlobalSign TLS root certificates:

Certificate 1

- Name: GlobalSign
- Alias name: globalsignr46
- Distinguished name: CN=GlobalSign Root R46, O=GlobalSign nv-sa, C=BE

Certificate 2

- Name: GlobalSign
- Alias name: globalsigne46
- Distinguished name: CN=GlobalSign Root E46, O=GlobalSign nv-sa, C=BE

See [JDK-8316138 \(JDK Bug System\)](#).

Fix for long garbage collection pauses due to imbalanced iteration during the **Code Root Scan** phase

The **Code Root Scan** phase of garbage collection finds references to Java objects within compiled code. To speed up this process, a cache is maintained within each region of the compiled code that contains references into the Java heap.

On the assumption that the set of references was small, previous releases used a single thread per region to iterate through these references. This single-threaded approach introduced a scalability bottleneck, where performance could be reduced if a specific region contained a large number of references.

In Red Hat build of OpenJDK 17.0.12, multiple threads are used, which helps to remove any scalability bottleneck.

See [JDK-8315503 \(JDK Bug System\)](#).

Change in behavior for AWT headless mode detection on Windows

In earlier releases, unless the **java.awt.headless** system property was set to **true**, a call to **java.awt.GraphicsEnvironment.isHeadless()** returned **false** on Windows Server platforms.

From Red Hat build of OpenJDK 17.0.12 onward, unless the **java.awt.headless** property is explicitly set to **false** and if no valid monitor is detected on the current system at runtime, a call to **java.awt.GraphicsEnvironment.isHeadless()** returns **true** on Windows Server platforms. A valid monitor might not be detected, for example, if a session was initiated by a service or by PowerShell remoting.

This change in behavior means that applications running under these conditions, which previously expected to run in a headful context, might now encounter unexpected **HeadlessException** errors being thrown by Abstract Window Toolkit (AWT) operations.

You can reinstate the old behavior by setting the **java.awt.headless** property to **false**. However, if applications are running in headful mode and a valid display is not available, these applications are likely to continue experiencing unexpected issues.

See [JDK-8185862 \(JDK Bug System\)](#).

CHAPTER 5. ADVISORIES RELATED TO THIS RELEASE

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

- [RHSA-2024:4568](#)
- [RHSA-2024:4569](#)
- [RHSA-2024:4570](#)

Revised on 2024-07-22 15:54:53 UTC