



Red Hat build of OpenJDK 21

Release notes for Eclipse Temurin 21.0.3

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 Eclipse Temurin 21.0.3 provide an overview of new features in OpenJDK 21 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 ECLIPSE TEMURIN	6
CHAPTER 2. ECLIPSE TEMURIN FEATURES	7
New features and enhancements	7
Java compiler aligns with the Java Language Specification by rejecting the final keyword in record patterns	7
XML Security for Java updated to Apache Santuario 3.0.3	7
TrimNativeHeapInterval option available as a product switch	7
SystemTray.isSupported() method returns false on most Linux desktops	8
Certainly R1 and E1 root certificates added	8
Precise parallel scanning of large object arrays for young collection roots	8
Fixed potential JVM failures when using ZGC and a non-default ObjectAlignmentInBytes value	9

PREFACE

Open Java Development Kit (OpenJDK) is a free and open source implementation of the Java Platform, Standard Edition (Java SE). Eclipse Temurin is available in four LTS versions: OpenJDK 8u, OpenJDK 11u, OpenJDK 17u, and OpenJDK 21u.

Binary files for Eclipse Temurin are available for macOS, Microsoft Windows, and multiple Linux x86 Operating Systems including Red Hat Enterprise Linux and Ubuntu.

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.

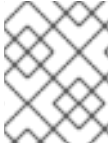
MAKING OPEN SOURCE MORE INCLUSIVE

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 ECLIPSE TEMURIN

Red Hat will support select major versions of Eclipse Temurin in its products. For consistency, these versions remain similar to Oracle JDK versions that Oracle designates as long-term support (LTS).

A major version of Eclipse Temurin will be supported for a minimum of six years from the time that version is first introduced. For more information, see the [Eclipse Temurin Life Cycle and Support Policy](#).



NOTE

RHEL 6 reached the end of life in November 2020. Because of this, Eclipse Temurin does not support RHEL 6 as a supported configuration.

CHAPTER 2. ECLIPSE TEMURIN FEATURES

Eclipse Temurin does not contain structural changes from the upstream distribution of OpenJDK.

For the list of changes and security fixes that the latest OpenJDK 21 release of Eclipse Temurin includes, see [OpenJDK 21.0.3 Released](#).

New features and enhancements

Review the following release notes to understand new features and feature enhancements included with the Eclipse Temurin 21.0.3 release:

Java compiler aligns with the Java Language Specification by rejecting the `final` keyword in record patterns

The Java 21 release enhanced the Java language with pattern matching for switch statements. However, OpenJDK 21 provides a **javac** compiler that allows use of the **final** keyword before a record pattern (for example, **case final R(...) ->**). This use of the **final** keyword violates the Java Language Specification.

In OpenJDK 21.0.3, the **javac** compiler is aligned with the Java Language Specification. This enhancement means that any program that includes the **final** keyword in a switch statement now fails to compile. In this situation, to ensure that the program compiles successfully, you must remove the **final** keyword.

See [JDK-8317300 \(JDK Bug System\)](#).

XML Security for Java updated to Apache Santuario 3.0.3

In OpenJDK 21.0.3, the XML signature implementation is based on Apache Santuario 3.0.3.

This enhancement introduces the following four SHA-3-based RSA-MGF1 **SignatureMethod** algorithms:

- **SHA3_224_RSA_MGF1**
- **SHA3_256_RSA_MGF1**
- **SHA3_384_RSA_MGF1**
- **SHA3_512_RSA_MGF1**

Because the **javax.xml.crypto.dsig.SignatureMethod** API cannot be modified in update releases to provide constant values for the new algorithms, use the following equivalent string literal values for these algorithms:

- **`http://www.w3.org/2007/05/xmldsig-more#sha3-224-rsa-MGF1`**
- **`http://www.w3.org/2007/05/xmldsig-more#sha3-256-rsa-MGF1`**
- **`http://www.w3.org/2007/05/xmldsig-more#sha3-384-rsa-MGF1`**
- **`http://www.w3.org/2007/05/xmldsig-more#sha3-512-rsa-MGF1`**

See [JDK-8319124 \(JDK Bug System\)](#).

TrimNativeHeapInterval option available as a product switch

OpenJDK 21.0.3 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 \(JDK Bug System\)](#).

SystemTray.isSupported() method returns false on most Linux desktops

In OpenJDK 21.0.3, the **java.awt.SystemTray.isSupported()** method returns **false** on systems that do not support the **SystemTray** API correctly. This enhancement is in accordance with the **SystemTray** API specification.

The **SystemTray** API is used to interact with the taskbar in the system desktop to provide notifications. **SystemTray** might also include an icon representing an application. Due to an underlying platform issue, GNOME desktop support for taskbar icons has not worked correctly for several years. This platform issue affects the JDK's ability to provide **SystemTray** support on GNOME desktops. This issue typically affects systems that use GNOME Shell 44 or earlier.



NOTE

Because the lack of correct **SystemTray** support is a long-standing issue on some systems, this API enhancement to return **false** on affected systems is likely to have a minimal impact on users.

See [JDK-8322750 \(JDK Bug System\)](#).

Certainly R1 and E1 root certificates added

In OpenJDK 21.0.3, the **cacerts** truststore includes two Certainly root certificates:

Certificate 1

- Name: Certainly
- Alias name: certainlyrootr1
- Distinguished name: CN=Certainly Root R1, O=Certainly, C=US

Certificate 2

- Name: Certainly
- Alias name: certainlyrootel
- Distinguished name: CN=Certainly Root E1, O=Certainly, C=US

See [JDK-8321408 \(JDK Bug System\)](#).

Precise parallel scanning of large object arrays for young collection roots

During the collection of young generation objects, the Parallel Garbage Collector (GC) partitions old generation objects into 64 kB stripes to scan for references to the young generation. These stripes are assigned to worker threads to perform scanning in parallel.

Previous releases of OpenJDK 21 did not constrain the worker threads to their own stripe. This lack of constraint imposed limits on parallelism. For example, if a large object started in a stripe that was allocated to a specific worker thread, this thread could end up scanning an object with thousands of references across multiple stripes.

OpenJDK 21.0.3 constrains each worker thread to its allocated stripe. Each thread now processes only the interesting parts of large object arrays. When large object arrays are present, pauses for the Parallel GC are now similar to pauses for the Garbage-First (G1) collector. In some cases, this enhancement helps to reduce the length of pauses by 75–80%. For example, a 100-millisecond pause in earlier releases could be reduced to a 20-millisecond pause in this release.

See [JDK-8310031 \(JDK Bug System\)](#).

Fixed potential JVM failures when using ZGC and a non-default `ObjectAlignmentInBytes` value

In earlier releases of OpenJDK 21, if you ran the JVM with the **`-XX:+UseZGC`** option and a non-default value for **`-XX:ObjectAlignmentInBytes`**, the JVM could fail or malfunction. This issue occurred because **`ZBarrierSet::clone_obj_array`** ignored padding at the end of an object array.

OpenJDK 21.0.3 resolves this issue to ensure that you can successfully run the JVM when using the Z Garbage Collector (ZGC) and non-default values for Java object alignment when running the JVM.

See [JDK-8325074 \(JDK Bug System\)](#).

Revised on 2024-04-26 14:04:24 UTC