



## Red Hat build of OpenJDK 21

红帽构建的 OpenJDK 21.0.3 发行注记





## 法律通告

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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> 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<sup>®</sup> 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.

## 摘要

红帽构建的 OpenJDK 21.0.3 发行注记 文档概述了红帽构建的 OpenJDK 21 中的新功能以及潜在的已知问题列表和可能的临时解决方案。

---

## 目录

前言 .....	3
提供有关红帽构建的 OPENJDK 文档的反馈 .....	4
使开源包含更多 .....	5
第 1 章 红帽构建的 OPENJDK 支持政策 .....	6
第 2 章 与上游 OPENJDK 21 的不同 .....	7
第 3 章 RED HAT BUILD OF OPENJDK 功能 .....	8
Red Hat build of OpenJDK 的改进 .....	8
第 4 章 与本发行版本相关的公告 .....	11



## 前言

Open Java Development Kit (OpenJDK)是 Java Platform, Standard Edition (Java SE)的一个免费的开源实现。红帽构建的 OpenJDK 有四个版本：8u、11u、17u 和 21u。

红帽构建的 OpenJDK 软件包在 Red Hat Enterprise Linux 和 Microsoft Windows 上提供，并作为红帽生态系统目录中的 JDK 和 JRE 提供。

## 提供有关红帽构建的 OPENJDK 文档的反馈

要报告错误或改进文档，请登录到 Red Hat JIRA 帐户并提交问题。如果您没有 Red Hat Jira 帐户，则会提示您创建一个帐户。

### 流程

1. 单击以下链接 [以创建 ticket](#)。
2. 在 **Summary** 中输入问题的简短描述。
3. 在 **Description** 中提供问题或功能增强的详细描述。包括一个指向文档中问题的 URL。
4. 点 **Submit** 创建问题，并将问题路由到适当的文档团队。

---

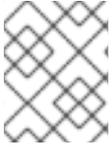
## 使开源包含更多

红帽致力于替换我们的代码、文档和 Web 属性中存在问题的语言。我们从这四个术语开始：master、slave、黑名单和白名单。由于此项工作十分艰巨，这些更改将在即将推出的几个发行版本中逐步实施。详情请查看 [CTO Chris Wright 的信息](#)。

## 第 1 章 红帽构建的 OPENJDK 支持政策

红帽在其产品中支持选择版本的 OpenJDK 的主版本。为了实现一致性，这些版本与指定为长期支持 (LTS) 的 Oracle JDK 版本类似。

自首次引入该版本起，红帽构建的 OpenJDK 主版本将最少提供六年的支持。如需更多信息，请参阅 [OpenJDK 生命周期和支持政策](#)



### 注意

RHEL 6 于 2020 年 11 月结束其生命周期。因此，红帽构建的 OpenJDK 不支持 RHEL 6 作为支持的配置。

## 第 2 章 与上游 OPENJDK 21 的不同

Red Hat 在 Red Hat Enterprise Linux 中构建 OpenJDK 包含了很多来自 OpenJDK 上游发行版的结构更改。红帽构建的 Microsoft Windows 版本尝试尽快遵循 Red Hat Enterprise Linux 更新。

以下列表详细介绍了最显著的 OpenJDK 21 更改：

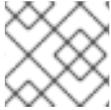
- FIPS 支持。Red Hat build of OpenJDK 21 会自动检测 RHEL 是否处于 FIPS 模式，并自动配置红帽构建的 OpenJDK 21 以在该模式下运行。此更改不适用于适用于 Microsoft Windows 的红帽构建的 OpenJDK 构建。
- 加密策略支持。红帽构建的 OpenJDK 21 从 RHEL 系统配置获取启用的加密算法和密钥大小限制列表。这些配置组件由传输层安全(TLS)加密协议、证书路径验证和任何签名的 JAR 使用。您可以设置不同的安全配置集来平衡安全性和兼容性。此更改不适用于适用于 Microsoft Windows 的红帽构建的 OpenJDK 构建。
- **src.zip** 文件包含红帽构建的 OpenJDK 附带的所有 JAR 库的源。
- RHEL 上的红帽 OpenJDK 构建使用系统范围的时区数据文件作为时区信息的来源。
- RHEL 上的红帽构建的 OpenJDK 使用系统范围的 CA 证书。
- Microsoft Windows 上的红帽构建的 OpenJDK 包括 RHEL 的最新可用时区数据。
- Microsoft Windows 上的红帽构建的 OpenJDK 使用 RHEL 的最新可用 CA 证书。

### 其他资源

- 请参阅[增强系统 FIPS 检测\(RHEL 规划 JIRA\)](#)
- 请参阅[使用系统范围的加密策略\(RHEL 文档\)](#)

## 第 3 章 RED HAT BUILD OF OPENJDK 功能

最新的红帽构建的 OpenJDK 21 发行版本可能包括新功能。另外，最新版本可能会增强、弃用或删除来自以前红帽构建的 OpenJDK 21 版本的功能。



### 注意

有关所有其他更改和安全修复，请参阅 [OpenJDK 21.0.3 发行版本](#)。

### Red Hat build of OpenJDK 的改进

红帽构建的 OpenJDK 21 提供对最初在红帽构建的 OpenJDK 版本中创建的功能的改进。

#### Java 编译器通过拒绝记录模式中的最终关键字来与 Java 语言规格一致

Java 21 版本增强了 Java 语言，其特征与交换机语句的模式匹配。但是，红帽构建的 OpenJDK 21 提供了一个 `javac` 编译器，允许在记录模式前使用最终关键字（例如，最终 `R (...)->`）。这对 `final` 关键字的使用违反了 Java 语言规范。

在红帽构建的 OpenJDK 21.0.3 中，`javac` 编译器与 Java 语言规格一致。此增强意味着在 `switch` 语句中包含最终关键字的任何程序现在都无法编译。在这种情况下，为了确保程序成功编译，您必须删除 `final` 关键字。

请参阅 [JDK-8317300 \(JDK Bug System\)](#)。

#### Java 更新到 Apache Santuario 3.0.3 的 XML 安全性。

在红帽构建的 OpenJDK 21.0.3 中，XML 签名实现基于 Apache Santuario 3.0.3。

此增强包括以下四个基于 SHA3 的 RSA-MGF1 `SignatureMethod` 算法：

- `SHA3_224_RSA_MGF1`
- `SHA3_256_RSA_MGF1`
- `SHA3_384_RSA_MGF1`
- `SHA3_512_RSA_MGF1`

由于在更新版本中无法修改 `javax.xml.crypto.dsig.SignatureMethod` API，以便为新算法提供恒定值，因此请为这些算法使用以下等效的字符串字面值：

- `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`

请参阅 [JDK-8319124 \(JDK Bug System\)](#)。

#### `TrimNativeHeapInterval` 选项可作为产品交换机使用

红帽构建的 OpenJDK 21.0.3 提供 `-XX:TrimNativeHeapInterval=ms` 作为官方产品交换机。此功能增强使 JVM 可以在受支持的平台上以指定间隔（以毫秒为单位）修剪原生堆。目前，只有这个增强支持的平台是带有 `glibc` 的 Linux。

您可以通过设置 `TrimNativeHeapInterval=0` 来禁用修剪。修剪功能默认为禁用。

请参阅 [JDK-8325496 \(JDK Bug System\)](#)。

在大多数 Linux 桌面上，`SystemTray.isSupported ()` 方法返回 `false`

在红帽构建的 OpenJDK 21.0.3 中，`java.awt.SystemTray.isSupported ()` 方法在不支持 `SystemTray` API 的系统上返回 `false`。此功能增强取决于 `SystemTray` API 规格。

`SystemTray` API 用于与系统桌面中的任务栏交互，以提供通知。`SystemTray` 可能还包括代表应用程序的图标。由于底层平台问题，对任务栏图标的 GNOME 桌面支持在数年内无法正常工作。此平台问题会影响 JDK 在 GNOME 桌面上提供 `SystemTray` 支持的能力。此问题通常会影响到使用 GNOME Shell 44 或更早版本的系统。



### 注意

因为缺少正确的 `SystemTray` 支持是某些系统上的长期问题，所以这个 API 增强在受影响系统中返回 `false` 可能会对用户造成最小的影响。

请参阅 [JDK-8322750 \(JDK Bug System\)](#)。

**意外的 R1 和 E1 根证书**

在红帽构建的 OpenJDK 21.0.3 中，`cacerts` truststore 包括两个 Certainly root 证书：

证书 1

- 名称：Certainly
- 别名名称：clearrootr1
- 区分名称：CN=Certainly Root R1, O=Certainly, C=US

证书 2

- 名称：Certainly
- 别名名称：clearrootel
- 区分名称：CN=Certainly Root E1, O=Certainly, C=US

请参阅 [JDK-8321408 \(JDK Bug System\)](#)。

**为 sng 集合根对大型对象阵列进行精确扫描**

在生成对象的集合过程中，Parallel Garbage Collector (GC) 将旧生成对象分区到 64 kB 条带，以扫描对年轻代的引用。这些条带被分配给 worker 线程，以便并行执行扫描。

以前的红帽构建的 OpenJDK 21 版本不会将 worker 线程限制为自己的条带。这缺少对 parallelism 的限制。例如，如果在分配给特定 worker 线程的条带中启动大型对象，则此线程最终可能会扫描多个条带上数千个引用的对象。

红帽构建的 OpenJDK 21.0.3 将每个 worker 线程限制为其分配的条带。现在，每个线程只处理大型对象阵列的有趣的部分。当大型对象阵列存在时，对 Parallel GC 暂停现在类似于 Garbage-First (G1) 收集器的暂停。在某些情况下，这个增强有助于将暂停的长度减少 75-80%。例如，之前版本的一个 100millisecond 暂停可能会降低到这个版本中的 20millisecond pause。

请参阅 [JDK-8310031 \(JDK Bug System\)](#)。

**修复了使用 ZGC 和非默认 `ObjectAlignmentInBytes` 值时潜在的 JVM 失败**

在早期版本的 OpenJDK 21 中，如果您使用 **-XX:+UseZGC** 选项运行 JVM，并为 **-XX:ObjectAlignmentInBytes** 运行 JVM，则 JVM 可能会失败或出现故障。出现这个问题的原因是，**ZBarrierSet::clone\_obj\_array** 会忽略对象数组末尾的 padding。

红帽构建的 OpenJDK 21.0.3 解决了这个问题，在使用 Z Garbage Collector (ZGC) 和非默认值时，您可以成功运行 JVM。

请参阅 [JDK-8325074 \(JDK Bug System\)](#)。

## 第 4 章 与本发行版本相关的公告

以下公告包括了记录程序错误修复和 CVE 修复：

- [RHSA-2024:1826](#)
- [RHSA-2024:1827](#)
- [RHSA-2024:1828](#)

更新于 2024-05-10