OpenJDK 17

Release notes for OpenJDK 17.0.3
Abstract

This document provides an overview of new features in OpenJDK 17, and a list of potential known issues and possible workarounds.
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREFACE</td>
<td>3</td>
</tr>
<tr>
<td>MAKING OPEN SOURCE MORE INCLUSIVE</td>
<td>4</td>
</tr>
<tr>
<td>PROVIDING FEEDBACK ON RED HAT DOCUMENTATION</td>
<td>5</td>
</tr>
<tr>
<td>CHAPTER 1. SUPPORT POLICY FOR OPENJDK</td>
<td>6</td>
</tr>
<tr>
<td>CHAPTER 2. DIFFERENCES FROM UPSTREAM OPENJDK 17</td>
<td>7</td>
</tr>
<tr>
<td>CHAPTER 3. OPENJDK FEATURES</td>
<td>8</td>
</tr>
<tr>
<td>3.1. OPENJDK ENHANCEMENTS</td>
<td>8</td>
</tr>
<tr>
<td>OCSP response verification with RSASSA-PSS</td>
<td>8</td>
</tr>
<tr>
<td>CHAPTER 4. KNOWN ISSUES</td>
<td>9</td>
</tr>
<tr>
<td>XPath expressions</td>
<td>9</td>
</tr>
<tr>
<td>Alternative Data Streams (ADS) behavior on OpenJDK for Microsoft Windows</td>
<td>9</td>
</tr>
<tr>
<td>CHAPTER 5. ADVISORIES RELATED TO THIS RELEASE</td>
<td>10</td>
</tr>
</tbody>
</table>
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 three versions: OpenJDK 8u, OpenJDK 11u, and OpenJDK 17u.

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.
PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your feedback on our documentation. To provide feedback, you can highlight the text in a document and add comments.

This section explains how to submit feedback.

Prerequisites

- You are logged in to the Red Hat Customer Portal.
- In the Red Hat Customer Portal, view the document in Multi-page HTML format.

Procedure

To provide your feedback, perform the following steps:

1. Click the Feedback button in the top-right corner of the document to see existing feedback.

   NOTE
   
   The feedback feature is enabled only in the Multi-page HTML format.

2. Highlight the section of the document where you want to provide feedback.

3. Click the Add Feedback pop-up that appears near the highlighted text.

   A text box appears in the feedback section on the right side of the page.

4. Enter your feedback in the text box and click Submit.

   A documentation issue is created.

5. To view the issue, click the issue tracker link in the feedback view.
CHAPTER 1. SUPPORT POLICY FOR OPENJDK

Red Hat supports select major versions of OpenJDK in its products. For consistency, these versions remain similar to Oracle JDK versions that are designated as long-term support (LTS).

Red Hat supports a major version of OpenJDK for a minimum of six years from the time Red Hat first introduces OpenJDK.

OpenJDK 17 is supported on Microsoft Windows and Red Hat Enterprise Linux until November 2027.

**NOTE**

RHEL 6 has reached the end of life in November 2020. OpenJDK 17 is not supported on RHEL 6.

**Additional resources**

For more information, see the [OpenJDK Life Cycle and Support Policy](#).
CHAPTER 2. DIFFERENCES FROM UPSTREAM OPENJDK 17

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

The following list details the most notable Red Hat OpenJDK 17 changes:

- **FIPS support.** Red Hat OpenJDK 17 automatically detects whether RHEL is in FIPS mode and automatically configures OpenJDK 17 to operate in that mode. This change does not apply to OpenJDK builds for Microsoft Windows.

- **Cryptographic policy support.** Red Hat 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 OpenJDK builds for Microsoft Windows.

- **Red Hat 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 OpenJDK builds for Microsoft Windows.**

- **The src.zip file includes the source for all of the JAR libraries shipped with OpenJDK.**

- **Red Hat OpenJDK on RHEL uses system-wide timezone data files as a source for timezone information.**

- **Red Hat OpenJDK on RHEL uses system-wide CA certificates.**

- **Red Hat OpenJDK on Microsoft Windows includes the latest available timezone data from RHEL.**

- **Red Hat OpenJDK on Microsoft Windows uses the latest available CA certificate from RHEL.**

**Additional resources**

- **For more information about detecting if a system is in FIPS mode, see the Improve system FIPS detection example on the Red Hat RHEL Planning Jira.**

- **For more information about cryptographic policies, see Using system-wide cryptographic policies.**
CHAPTER 3. OPENJDK FEATURES

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

NOTE
For all the other changes and security fixes, see OpenJDK 17.0.3 Released.

3.1. OPENJDK ENHANCEMENTS

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

OCSP response verification with RSASSA-PSS
Before the OpenJDK 17.0.3 release, verification of the Online Certificate Status Protocol (OCSP) response by the RSASSA-PSS algorithm failed. This issue occurred because the HTTP GET helper method did not set extra parameters for an RSASSA-PSS signature.

The OpenJDK 17.0.3 resolves this issue, so that an OCSP response that is sent by the HTTP GET method can be verified by the RSASSA-PSS algorithm.

See JDK-8179503.
CHAPTER 4. KNOWN ISSUES

OpenJDK 17 might include known issues. Solutions might exist for some of these known issues.

XPath expressions
OpenJDK 17 specifies new default limits for XPath expressions. If you find that your code is exceeding these limits, you can raise them by passing the following parameters to your OpenJDK deployment or by setting the JAVA_TOOL_OPTIONS environment variable to use them:

- -Djdk.xml.xpathExprGrpLimit, which defaults to 10
- -Djdk.xml.xpathExprOpLimit, which defaults to 100
- -Djdk.xml.xpathTotalOpLimit, which defaults to 10,000

These default limits restrict the number of expression groups, the number of operators in a single expression, and the total number of operators.

Alternative Data Streams (ADS) behavior on OpenJDK for Microsoft Windows

Description
The OpenJDK 17 for Microsoft Windows release sets the default value for the jdk.io.File.enableADS to false, which can cause some Java applications to fail because they cannot recognize certain characters in Microsoft Windows hidden file paths. This is a known regression in behavior and Red Hat intends to fix this issue at a later release of OpenJDK 17. If your application fails because of this configuration, you might receive a java.io.FileNotFoundException: Invalid file path exception message. The output from this message is dependent on the string value that is specified in the FileOutputStream class in your OpenJDK 17 for Microsoft Windows build. The following example demonstrates a value of NUL: that is specified as the string value for the FileOutputStream class:

```java
import java.io.FileOutputStream;

public class NulFile {
    public static void main(String[] args) throws Exception {
        new FileOutputStream("NUL: ");
    }
}
```

Workaround
You can workaround this issue by choosing one of the following options:

- Pass the -Djdk.io.File.enableADS=true parameter to your OpenJDK deployment.
- Set the JAVA_TOOL_OPTIONS environment variable to JAVA_TOOL_OPTIONS=-Djdk.io.File.enableADS=true.

Either option permits your Java applications to write data into alternative data streams (ADS) for your OpenJDK 17 for Microsoft Windows build.

See JDK-8285500
CHAPTER 5. ADVISORIES RELATED TO THIS RELEASE

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

- RHSA-2022:1436
- RHSA-2022:1437
- RHSA-2022:1445
- RHBA-2022:1447

Revised on 2022-05-05 14:51:54 UTC