Red Hat build of OpenJDK 8

Eclipse Temurin 8.0.352 release notes

Last Updated: 2024-05-10

Legal Notice

Abstract

Review the release notes to understand new features and feature enhancements that have been included with the latest build of OpenJDK 8 that is provided by Eclipse Temurin.

Table of Contents

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 three LTS versions: OpenJDK 8u, OpenJDK 11u, and OpenJDK 17u.

Packages for Eclipse Temurin are available for Microsoft Windows and on 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 are the same versions that Oracle designates as long-term support (LTS) for the Oracle JDK.

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 included in the latest OpenJDK 8 release of Eclipse Temurin, see OpenJDK 8u352 Released.

New features and enhancements

Review the following release notes to understand new features and feature enhancements that the Eclipse Temurin 8.0.352 release provides:

Reference object changes and configurations

From OpenJDK 8.0.352 onward, you can no longer clone **Reference** objects.

If you attempt to clone a reference object, the **java.lang.ref.Reference::clone** method throws a **CloneNotSupportedException** message.

If you want to copy an existing **Reference** object, you must use the constructor of the appropriate **Reference** subclass to create a **Reference** object. This ensures the new **Reference** object contains referent and reference queues that are identical to the target **Reference** object.

For the OpenJDK 8.0.352 release, the **java.lang.ref.Reference.enqueue** method changes behavior. When application code calls the **java.lang.ref.Reference.enqueue** method, this method clears the **Referent** before it adds the object to the registered queue. After the **Reference** object is enqueued, code that expects the return value of **java.lang.ref.Reference.get()** to be **non-null** might throw a **NullPointerException**.

The OpenJDK 8.0.352 release changes the behavior of **PhantomReference** objects, so that they are cleared before being enqueued in any associated queues. This is the same as the existing behaviour for **SoftReference** and **WeakReference** objects.

Links

- See JDK-8201793 (JDK Bug System).
- See JDK-8175797 (JDK Bug System).
- See JDK-8071507 (JDK Bug System).
- For more information about these **Reference** object configurations, see OpenJDK 8 Maintenance Release 4 (Red Hat Customer Portal)

Enablement of TLSv1.3 for client roles

OpenJDK 8.0.352 enables TLSv1.3 protocol support for client roles, by default.

From the OpenJDK 8.0.272 release, TLSv1.3 protocol support for server roles was already enabled.

If you create a TLS client role in OpenJDK 8.0.352 while keeping the default protocol setting, and TLSv1.3 is used in the connection established with the TLS server, compatibility issues might affect your application. The following list details common compatibility issues:

- TLSv1.3 uses a half-duplex-close policy whereas TLSv1.2 uses a full-duplex-close policy. You can use the **jdk.tls.acknowledgeCloseNotify** system property to configure TLSv1.3 to use a full-duplex-close policy. For more information about this configuration, see JDK-8208526.

- TLSv1.3 does not support certain algorithms in the **signature_algorithms_cert** extension. For example, if you only allow Digital Signature Algorithm (DSA) for signature verification in your configurations, you will experience incompatibility issues when using the TLSv1.3 protocol.
- A client that uses DSA certificates for client authentication causes compatibility issues with TLSv1.3.
- TLSv1.3 contains different cipher suites than earlier TLS protocol versions. For an application with hard-coded unsupported cipher suites, compatibility issues might exist.
- TLSV1.3 session resumption and key update behaviors differ from earlier TLS protocol versions. An application that relies on handshake details from these protocols might experience compatibility issues.

If you need to disable TLSv1.3 protocol support for your client role, complete one of the following actions:

- Obtain a TLSv1.2 context with **SSLContext.getInstance("TLSv1.2")**.
- Set the **jdk.tls.client.protocols** system property to TLSv1.2. For example, **-Djdk.tls.client.protocols="TLSv1.2"**.
- Set an earlier TLS protocol for the OpenJDK **javax.net.ssl** API, as demonstrated with the following examples:

```
sslSocket.setEnabledProtocols(new String[] {"TLSv1.2"});
```

```
sslEngine.setEnabledProtocols(new String[] {"TLSv1.2"});
```

```
SSLParameters params = sslSocket.getSSLParameters();
params.setProtocols(new String[] {"TLSv1.2"});
slsSocket.setSSLParameters(params);
```

Links

- See JDK-8208526 (JDK Bug System).

jdk.httpserver.maxConnections system property

OpenJDK 8.0.352 adds a new system property, **jdk.httpserver.maxConnections**, that fixes a security issue where no connection limits were specified for the **HttpServer** service, which can cause accepted connections and established connections to remain open indefinitely.

You can use the **jdk.httpserver.maxConnections** system property to change the **HttpServer** service's behavior in the following ways:

- Set a value of **0** or a negative value, such as **-1**, to specify no connection limit for the service.
- Set a positive value, such as **1**, to cause the service to check any accepted connection against the current count of established connections. If the maximum number of established connections for the service is reached, the service immediately closes the accepted connection.

Support for Microsoft Visual Studio 2017

From the OpenJDK 8.0.352 release onward, the Windows JDK and JRE 1.8.0 releases are compiled with the Visual Studio 2017 toolchain, because this toolchain is currently supported by Microsoft.

NOTE

The Red Hat Customer Portal no longer uses the *Alternative toolchain* label to mark binaries that were compiled with the Visual Studio 2017 toolchain.

For customers that rely on the Microsoft Visual Studio 2010 toolchain for compiling binaries, which Red Hat labels as the *legacy toolchain*, Red Hat continues to support these binaries. On the **Software Details** page, on the Red Hat Customer Portal, a file compiled with this toolchain contains a **vs10** entry in its file name. For example, **openjdk-1.8.0.345/java-1.8.0-openjdk-1.8.0.352-2.b08.redhat.windows.vs10.x86_64.zip**.

IMPORTANT

Microsoft no longer supports the Visual Studio 2010 toolchain, so Red Hat can only provide limited support for any products related to this toolchain.

Revised on 2024-05-10 09:06:49 UTC