



# Best Practices for Moving to OpenJDK



This whitepaper explains what Zulu is and how to migrate applications from the Oracle JDK to the Zulu builds of OpenJDK.

### Introduction

How the Java Development Kit (JDK) is delivered is changing, and these changes have impacted a large number of Java users. Since JDK 11 (released in September 2018) it has no longer been possible to use the Oracle JDK in production without a commercial support agreement. To provide an alternative Java runtime with economically priced support and continued access to bug fixes and security patches, Azul Systems have created the Zulu OpenJDK distribution. This whitepaper explains what Zulu is and how to migrate applications from the Oracle JDK to the Zulu builds of OpenJDK. We have also appended a series of links from the Oracle and Azul Systems websites to help you to understand your Java support alternatives.

### The OpenJDK

In 2006 Sun Microsystems, the then custodian of Java, announced that it was open-sourcing its implementation of the JDK. To do this, it created the OpenJDK project, hosted at <http://openjdk.java.net>

Initially, just the JVM source code was released, but this was followed early in 2007 by that of the core libraries. Some parts remained closed-source until April 2009 with build 53 of OpenJDK 7, when a complete JDK could be built using only open-source code.

Since Java SE 7, the OpenJDK has been the reference implementation (RI) of the Java SE standard, as defined

by the relevant Java Specification Request (JSR). A project was also created for OpenJDK 6 to allow an opensource implementation of Java SE 6 to be built. Andrew Brygin of Azul Systems is the project lead for OpenJDK 6.

OpenJDK is licensed under the GNU Public License (GPL) version 2 with classpath exception (CPE). The CPE prevents the copyleft (or viral) nature of the GPL from affecting applications running on OpenJDK.

### Changes to JDK Binary Distribution and Updates

In September 2017, Oracle announced a series of changes to the way the JDK would be developed, distributed and updated. These changes can be summarised as follows:

- The release of the JDK has changed to a time-based model with two releases every year, in March and September.
- Oracle has started to produce a second binary distribution of the JDK, built using only the OpenJDK source code. This is released under the GPLv2 with classpath exception license, which is less restrictive than the Oracle Binary Code License (BCL) for Java SE used for the Oracle JDK binary.
- From JDK 11, there are no functional differences between the Oracle JDK binary and one built purely from OpenJDK source. To achieve this, Oracle has contributed some previously closed-source features to the OpenJDK (e.g., Mission Control and Flight Recorder). Oracle has also removed some features from the Oracle JDK binary such as Java Web Start and JavaFX.
- From JDK 11, the Oracle JDK license has changed to the Oracle Technology Network License Agreement (OTNLA), which is no longer completely free. Under this license, the Oracle JDK can be used freely for development and testing but requires a commercial support agreement for use in a production environment.
- Oracle OpenJDK binaries will only be updated for six months until the next JDK is released. Long-term support from Oracle will only be available for the Oracle JDK binary under the commercial support agreement.
- Free public updates for JDK 8 ended in January 2019 with update 211. Users can continue to use this version freely since it is still covered by the Oracle BCL. Since April 2019, updates have still been available from the java.oracle.com website but these are released under the OTNLA. Use of these updates also requires a commercial support agreement from Oracle.

### TABLE OF CONTENTS

Introduction	2	Encryption	3	Migrating Applications to use the Zulu JDK	5
The OpenJDK	2	Class Libraries	4	Getting support, timely security updates and bugfixes for Zulu	5
Changes to JDK Binary Distribution and Updates	2	Other Features	4	The Zulu Commercial Compatibility Kit	5
The Zulu Build of OpenJDK	3	Developer Tools	4	Conclusions	5
Differences Between OpenJDK and the Oracle JDK	3	Java Runtime Deployment Strategy	4	Additional Resources	5
Java Virtual Machine (JVM)	3	Migrating to Zulu Enterprise	5		
		Downloading and Installing Zulu	5		

**The Azul Zulu Build of OpenJDK**

The OpenJDK is simply source code; in order to use this as a Java runtime, it is necessary to build a binary distribution. Azul’s binary distribution of the OpenJDK is called Zulu.

To create Zulu, the standard build procedure, available as part of the OpenJDK, is used. To ensure conformance with the Java SE specification, the full suite of Technology Compatibility Kit (TCK) tests are run on the resulting Zulu binaries.

To ensure that the JDK is as stable and secure as possible, regular updates are provided for the OpenJDK, which contain bug fixes and security patches.

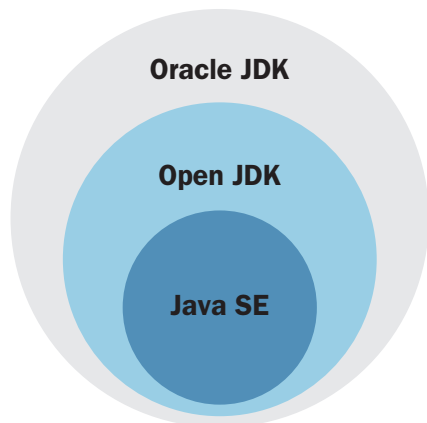
Until JDK 9, these updates were provided for the current release and the previous release during an overlap period. This overlap has lasted between 13 and 37 months since the release of JDK 6. OpenJDK 6, 7, 8 and 11 have all reached the end of availability of public updates. Azul includes backported updates from the current JDK in Zulu versions of all these JDKs. This enables customers to ensure the stability and security of older Java versions still in production use.

Azul’s engineering team have many staff-years of experience working on the JDK; many of them having worked for Sun Microsystems and Oracle.

**Differences Between OpenJDK and the Oracle JDK**

As the OpenJDK is the reference implementation of the Java SE standard, Oracle (and others) use this as the basis for their JDK binary distributions. However, prior to JDK 11, the Oracle JDK was a super-set of the OpenJDK, including a number of additional features. Some of these are open source; some are closed source.

The diagram below shows the relationship between the Java SE standard (as defined by the relevant Java Specification Request (JSR) of the Java Community Process (JCP)), the OpenJDK and the Oracle JDK (pre-JDK 11).



The following sections outline the differences between the source code of the OpenJDK and what is included in the Oracle JDK (prior to JDK 11). Azul makes specific changes to minimize these differences, and these are also noted.

**The following are not included in the OpenJDK:**

**Fonts:** Both the Oracle and Zulu JDKs provide the same set of commercially licensed fonts. For more details of fonts for Zulu see the later section on the Commercial Compatibility Kit.

**SNMP:** The Oracle JDK provides closed-source support. There is no equivalent in the Zulu JDK. This has been removed from the Oracle JDK as of JDK 11.

**Sound Drivers:** The Oracle JDK provides this for Windows only. There is no equivalent in the Zulu JDK.

**The following differ between OpenJDK and the Oracle JDK:**

**Color Matching:** The Oracle JDK replaces this feature with a closed-source version. The Zulu JDK uses the version from OpenJDK, which is comparable but not identical.

**Font Rendering:** The Oracle JDK uses a closed-source font renderer (T2K). Zulu (and OpenJDK) use an open-source renderer (FreeType), which is comparable but not identical.

**Java 2D Graphics Rasterizer:** Prior to JDK 9, the OpenJDK used the open-source Pisces renderer, which was replaced with a closed-source version, Ductus, in the Oracle JDK. In JDK 9, Pisces was replaced by the Marlin renderer, which gives performance comparable to Ductus. The Zulu JDK has used Marlin since JDK 8.

**Encryption**

The Oracle JDK and OpenJDK both support the same encryption and cryptography capabilities with identical code.

Since January 2018, (the release of JDK 8 update 161) all OpenJDK-based distributions (Zulu and Oracle) switched to using an “unlimited” strength cryptographic policy. Prior to this release, the default was “strong” (128 bit), and it was necessary to use additional packages to upgrade this to unlimited. For the Oracle JDK, there was the Java Cryptographic Extensions (JCE) package, and for Zulu, there was the Cryptographic Extension Kit (CEK).

Zulu and OpenJDK have generally followed the same cryptographic changes roadmap that Oracle has, which can be found here:

<https://www.java.com/en/jre-jdk-cryptoroadmap.html>



### Class Libraries

**JavaFX:** This was included in the Oracle JDK. There is a separate open-source project called OpenJFX. Azul can provide support for OpenJFX, through the ZuluFX product. The OpenJFX project can be found at <http://openjdk.java.net/projects/openjfx>

### Other Features

**Browser plugin:** This is included in the Oracle JDK but not the Zulu JDK. This was deprecated from the Oracle JDK in JDK 9 and will be removed from JDK 11.

**Java Web Start:** This is included in the Oracle JDK but not the Zulu JDK. This was deprecated from the Oracle JDK in JDK 9 and was removed from JDK 11. Azul offers an open-source alternative, IcedTEA web. This may require some work to migrate to and may not offer all the features provided by the Oracle version. Azul offers this without support.



### Developer Tools

**Mission Control:** In JDK 9 and earlier, this was a commercial product that could be used freely in development and testing but required a commercial contract with Oracle for use in production. Oracle open-sourced this tool, and from JDK 10 it is part of the OpenJDK. Azul provides this as a separate binary that can be used with any JVM that supports Flight Recorder functionality.

**Flight Recorder:** Similar to Mission Control, in JDK 10 and earlier, this tool was licensed separately. Oracle open-sourced this tool and, from JDK 11, it is part of OpenJDK. Azul has backported the code for Flight Recorder to OpenJDK 8, meaning that Mission Control can be used to analyse running Zulu 8 JVMs or recordings.

**VisualVM:** This was included in the Oracle JDK until JDK 8 but not the Zulu JDK. It was removed from JDK 9 but is still available as an open-source project from <https://visualvm.github.io>.

**Java DB:** This is a binary distribution of the Apache Derby database with no changes to the source code included with the Oracle JDK. This was included with the Oracle JDK until JDK 8 but not Zulu. It was removed from JDK 9 but is still available as an open-source project from <https://db.apache.org/derby/>



### Java Runtime Deployment Strategy

Users need to evaluate the options available to them when deciding on a strategy for existing and future deployment of JVM-based applications.

#### There are several choices:

**1:** Continue to use the existing runtime, such as JDK 8, without further updates for security patches or bug fixes. This is the option with the most significant level of risk, as systems will potentially be open to security flaws or instability. For specific use-cases, like PCI compliance, this may also expose the application to compliance regulation issues.

**2:** Migrate applications to the latest version of the JDK using free OpenJDK binaries and repeat this every six months, as new JDK versions are released. This choice still has considerable risks associated with it. Oracle has stated that new releases of the JDK may contain changes that break backward compatibility (until JDK 9, a key feature of the Java platform has been backwards compatibility). The effort required to test and certify applications and libraries on a new version of the JDK every six months should not be underestimated.

**3:** Continue to use the Oracle JDK binaries and pay for commercial support from Oracle. This eliminates the risks of the first two options but comes with a significant cost. Oracle has announced a new, subscription-based pricing scheme for JDK support. The pricing of this is based on a processor count, which is neither purely the number of cores in a processor, nor the number of sockets in a server. Calculating the number of processors can be complex, but for x86-based hardware, each pair of cores in a socket counts as a single processor. A typical server that has two sockets with hex-core processors will be priced as six processors. The cost of a Java SE subscription is \$15 per processor, per month giving an annual cost for the example server of \$1,050. In an estate with 1000 such servers, this will cost over a million dollars a year.

**4:** Use a JDK from an alternative supplier with lower cost. This is where Azul can play a crucial role with the Zulu Enterprise binary distribution of OpenJDK. Azul only counts servers, so the example in option three above would cost one-tenth of the Oracle price for premium support or a twelfth of the Oracle cost for standard support. Using Zulu Enterprise is obviously a very cost effective option.

### Migrating to Azul Zulu Enterprise

This section describes how you can migrate your applications to use Zulu Enterprise.

### Downloading and Installing Azul Zulu

Zulu OpenJDK is released under the GPLv2 with CPE license.

Zulu Binaries are available for the Windows, Linux, Solaris and MacOS X platforms and can be downloaded from: [azul.com/downloads/zulu](http://azul.com/downloads/zulu)

A comprehensive installation guide is also available at this address. This covers the full range of formats provided, prerequisites for installation as well as instructions for deployment to popular cloud providers like Microsoft Azure and Amazon Web Services. The guide also includes details of postinstallation tasks including how to switch the default Java runtime between different installed versions.

### Migrating Applications to use the Azul Zulu JDK

The list of differences between the OpenJDK and Oracle JDK in the earlier sections should be used to identify if there are any features used by an application that are not supported in the OpenJDK. Where alternatives exist, such as Apache Derby, an appropriate binary should be located and downloaded.

### The Azul Zulu Commercial Compatibility Kit

As mentioned in the section Differences Between OpenJDK and the Oracle JDK, specific commercial fonts are provided in the Oracle JDK that are not part of the OpenJDK. The Zulu Commercial Compatibility Kit (CCK) provides an additional range of Lucida fonts to give the same resources as the Oracle JDK.

The Zulu CCK requires acceptance of the associated terms of use. The installer will also ensure that the version of the Zulu CCK matches the installed version of Zulu, so it is essential to download the correct version. Further details and downloads can be found here: [azul.com/products/zulu-and-zulu-enterprise/cck-downloads](http://azul.com/products/zulu-and-zulu-enterprise/cck-downloads)

### Reality of Migrating To Azul Zulu Enterprise

One question that will occur to potential users of Zulu Enterprise is, how difficult is it to switch my JDK?

The approach that should be taken is to use the same process that is used in the past to deploy an update to the Oracle JDK. The directory structure and delivery format of Zulu Enterprise is identical.

Azul has numerous large enterprise customers, many with considerable data centre estates with thousands of servers running Java. Despite using Java to run an extremely wide variety of applications, no customer has reported any issues with the compatibility between the Oracle JDK and Zulu Enterprise. This has made the

migration smooth and seamless, with no additional administration costs incurred.

### Conclusions

Zulu is a build of the OpenJDK source code that fully conforms to the Java SE specification. Any application that relies only on standard Java components and APIs can switch to using Zulu simply by changing the path of where the Java executable is located.

With skilled engineers undertaking the technically complex tasks of backporting and integrating updates into older versions of the OpenJDK customers can continue to have the latest security and stability fixes while using older versions of Java.

The Zulu build of OpenJDK provides a cost-effective alternative to having to migrate applications to newer versions of Java.

If you would like to speak to an existing Zulu Enterprise user, please contact Azul Sales at [info@azul.com](mailto:info@azul.com) and we will respond within one business day.

### Additional Resources

#### From Oracle:

##### Java SE Support Roadmap:

[oracle.com/technetwork/java/java-se-support-roadmap.html](http://oracle.com/technetwork/java/java-se-support-roadmap.html)

##### Java SE Subscriptions FAQ:

[oracle.com/technetwork/java/javaseproducts/overview/javasesubscriptionfaq-4891443.html](http://oracle.com/technetwork/java/javaseproducts/overview/javasesubscriptionfaq-4891443.html)

##### Java SE Support Subscriptions Overview:

[oracle.com/java/java-se-subscription.html](http://oracle.com/java/java-se-subscription.html)

#### From Azul Systems:

##### Java SE Overview:

[azul.com/products/zulu-enterprise/key-features-benefits/](http://azul.com/products/zulu-enterprise/key-features-benefits/)

##### Azul Zulu Enterprise Support Overview:

[azul.com/products/zulu-enterprise/](http://azul.com/products/zulu-enterprise/)

##### Azul Product Support Lifecycle:

[azul.com/products/azul\\_support\\_roadmap/](http://azul.com/products/azul_support_roadmap/)

##### Key benefits of Zulu Enterprise:

[azul.com/products/zulu-enterprise/key-features-benefits/](http://azul.com/products/zulu-enterprise/key-features-benefits/)

## Connect with an Azul Expert

Azul Systems, Inc.  
385 Moffett Park Drive, Suite 115  
Sunnyvale, CA 94089 USA  
+1.650.230.6500  
[www.azul.com](http://www.azul.com)  
[info@azul.com](mailto:info@azul.com)