Transitioning from Oracle JDK to Azul Zulu Builds of OpenJDK

scc | azul

Organizations who are interested in exploring trusted alternatives to Oracle Java SE need reliable information on how to migrate their applications from the Oracle JDK distribution of OpenJDk to other distributions like Azul Zulu Builds of OpenJDK, which offer equivalent or better expert engineering support.

Azul has put together this guide based on migrating customers from the Oracle JDK to Azul Zulu Builds of Open JDK. While the time required to migrate depends on many factors, including whether applications are using discontinued technologies like Applets or Webstart, Azul can boast a 100% success rate.

- One of our banking customers converted 2500 applications over a weekend.
- One of the largest entertainment companies in the world transitioned thousands of applications within a couple of months.

Azul Zulu Builds are often described as drop-in replacements for the Oracle JDK — both Oracle JDK and the Azul Zulu Builds are derived from the OpenJDK repository. Core functionality like the JVM, libraries, etc. are completely interchangeable. No modifications to source code are required to swap one for the other nor is a recompilation of application code necessary.

In addition, each Azul Zulu binary passes all the tests of the Technology Compatibility Kit (TCK) provided as part of the relevant Java Specification Request (JSR). There are over 150,000 tests that ensure a binary conforms to the defined specification and provides a high level of confidence of functional equivalence between tested JDKs.

Azul suggests a 3-phase approach to transitioning applications to Azul Platform Core:

1. Planning
2. Implementation
3. Testing

1. Planning

Identify machines/instances needing to transition and the major JDK versions in use.

You can use an inventory report from a SAM (Software Asset Management) tool as a starting point. If you do not have a SAM, please see the Appendix regarding how to determine which JDK is installed on different operating systems.

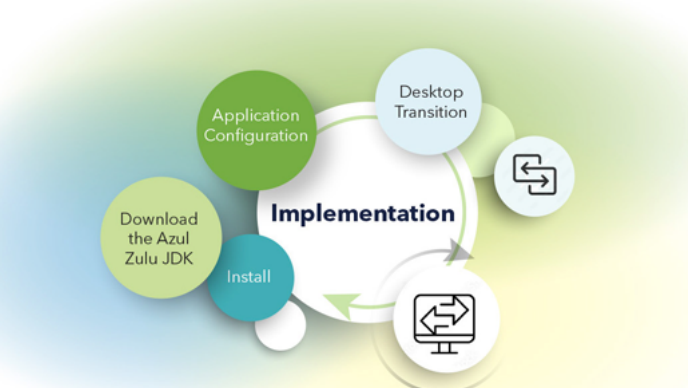
Align on a current JDK migration and update strategy with application managers. It can be helpful to classify different JDK environments according to urgency and complexity as well as based on existing release planning and policies.

Decide upon a centralized or decentralized approach. You may also decide to combine a migration with a transition to more recent versions of Java in a two-step process. Or you may decide to require use of only certain versions of the JDK, or to require use of stabilized builds only.

Find out more information about Azul's stabilized builds, or Critical Patch Updates:

https://www.azul.com/wp-content/uploads/tb-security-onlyupdates.pdf.

2. Implementation



Quick and swift migration strategy

All Azul JDKs and Oracle JDKs are built from the same source code from the OpenJDK Project. Core functionality like the JVM (Java Virtual Machine), libraries, etc. are completely interchangeable. No modifications to source code are required to swap one for the other nor is a recompilation of application code necessary. Migration experts, who are certified in the Azul migration methodology, can manage your migration for you.

Install the Azul Zulu JDK

The installation process depends on which format of distribution is being used.

Zip file Use a utility or tool to unzip the archive dependent on which operating system is being used. This is a manual installation method so the Azul JDK can be installed in a directory of your choosing.

Compressed tar file (.tar.gz). Use the UNIX command tar -xvf <zulu-package>.tar.gz. This is also a manual installation method so any directory can be chosen.

Windows MSI file. From the Windows command line, run msiexec /i <zulu-package>.msi /qn. This will install Zulu into the C:\Program Files\Zulu\<zu-lu-jdk>\ directory.

Linux RPM file: Install from a command prompt using the command:

- On Red Hat Enterprise Linux: yum install <zulu-package>.rpm
- On SUSE Linux Enterprise Server: zipper install <zulu-package>.rpm Linux DEB file:

On Ubuntu or Debian use the command: apt install <zulu-package.deb>.

MacOS DMG file. This can be installed graphically from the desktop or the command line using the command: hdiutil mount <zulu_package>.dmg
Full instructions for installation can be found in the Azul documentation at https://docs.azul.com/zulu/zuludocs/ZuluUserGuide/Title.htm.

See Appendix for details of using repositories.

Additional Fonts - Oracle JDK prior to JDK 11 included additional Lucida fonts. To provide compatibility with the Oracle JDK, Azul provides the Azul Commercial Compatibility Kit (CCK). For desktop machines, the CCK should be installed if graphical applications are being used. Details of the available CCK files and installation instructions can be found at https://www.azul.com/products/components/commercial-compatibility-kit/.

Desktop Transition – if you use desktop applications with Browser Plug-ins (Applet) and Java Web Start functionality, please see the section Desktop Machine Transition at the end of this document. These are not included in the Azul Zulu OpenJDK binary distributions.

Application Configuration - Having installed the Azul Zulu JDK on a machine, it may be necessary to reconfigure applications to use the new JDK.

Each application will vary in how it determines where the Java executable is located.

Here are some common scenarios.

1. The PATH environment variable.

This is set differently depending on which operating system is being used. The PATH environment variable should be modified to include the bin directory of the Azul Zulu JDK installation as the first place where an executable called java is located.

2. The JAVA_HOME environment variable.

Similar to PATH, this is also set differently depending on the operating system in use. JAVA_HOME indicates where the JDK is installed and should be set accordingly for the Azul Zulu JDK. Note that it is the installation directory so, unlike changing PATH, should not reference the bin sub-directory. JAVA_HOME is used by some applications (e.g. some application servers) but is not used universally by all applications.

3. The Tomcat server.

The default configuration file should be modified to reflect the location of the Azul Zulu JDK installation.

3. Testing

Functionally, there are no differences between the Oracle JDK and Azul Zulu OpenJDK (other than those already noted for desktop machines). This means that there will be no differences executing your Java application using either JDK.

However, it is recommended to run your standard tests for applications being used to ensure that none of the changes between JDK updates has affected application behavior. This is a good practice even for exact like for like version releases.

In conclusion, Azul Zulu Builds of OpenJDK are a direct replacement for the Oracle JDK (with the exception of the discontinued desktop features such as Applets and Web Start).

Once Java versions are identified, transition consists of installation of the latest versions of the Azul Zulu JDK and minor changes to application configuration to reflect the new location of the JDK.

Appendix

How to determine what JDK version is deployed on various Operating Systems:

Ubuntu/Debian Linux

Assuming that the JDK has been installed using the package manager, installed versions can be listed using the command:

$ dpkg -l | egrep [jJ][rR][eE]\|[jJ][dD][kK]
This will produce output similar to this example from a machine running Ubuntu

ii default-jre-headless 2:1.11-68ubuntu1~18.04.1 amd64 Standard Java or Java compatible Runtime (headless)
ii jdk1.8 1.8.0202-1 amd64 Java Platform Standard Edition Development Kit ii openjdk-11-jre-headless: amd64 11.0.7 +10-2ubun-tu2~18.04 amd64

OpenJDK Java runtime, using Hotspot JIT (headless)
ii zulu-8 8.48.0.51-1 amd64 Azul Systems Zulu JDK 8.48.0.51 (8u262-b19)

This shows four Java runtimes are installed:

1. Default Ubuntu 18.04 Java Runtime Environment (JRE)
2. Oracle JDK 8 update 202 (Although Oracle is not stated explicitly, it is implied by the use of the Java(TM) trademark).
3. OpenJDK 11 update 7 from the Ubuntu Linux distribution
4. Azul Zulu OpenJDK 8 update 262. An alternative approach is to search the whole filesystem, which will also find any JDKs that have been installed manually (such as by unpacking a zip file). Use this command to search for files named java that are executable.

find / -perm u+x -type f -name java

NOTE: To enable a full scan of the file system, this should be run as the root user.

The paths listed will often indicate the exact version of the JDK. If, however, for example, the directory is /opt/jdk8, the exact version installed will need to be determined by running the java command:

$ /opt/jdk8/bin/java -version

This will produce output similar to this:

java version “1.8.0_202” Java(TM) SE Runtime Environment (build 1.8.0_202- b08) Java HotSpot(TM) 64-Bit Server VM (build 25.202-b08, mixed mode)

Again, this is the Oracle JDK 8 update 202. Oracle/Red Hat Enterprise Linux/SUSE Linux Enterprise Server Use the command (as root) # rpm -qa --queryformat “%{NAME} %{VERSION} %{VENDOR} \n” | egrep [jJ][rR][eE]\|[jJ][dD][kK]

This will produce output something like:

jdk1.8 1.8.0_202 Oracle Corporation

Alternatively, the same full file system search method can be used as described for Ubuntu/Debian.

Windows

From a command prompt run the command wmic, which starts an interactive shell.

Type the command: product get name
In this output, you will see something like this
Java 8 Update 202 (64-bit)
Java SE Development Kit 8 Update 202 (64-bit)

MacOS X

To list JDKs installed using the packaging system, use the command:
$ pkgutil --pkgs | egrep jre\|jdk

This will produce output like this
com.oracle.jre com.oracle.jdk-14.0.2 com.oracle.jdk8u202

The com.oracle.jre package does not provide a version number. This can be obtained using the command:
$ pkgutil –pkg-info com.oracle.jre

Which will produce output like this, showing it is Java SE 10.0.2
package-id: com.oracle.jre
version: 10.0.2.0.13 volume: /
location: Library/Internet Plug-Ins/
JavaAppletPlugin.plugin install-time: 1538383780

MacOS is a UNIX-based operating system, so the full file system search method described for Ubuntu/ Debian Linux can also be used.

Solaris

To list JDKs installed using the packaging system, use these two commands:

$ pkg list | egrep [jJ][rR][eE]\|[jJ][dD][kK]
$ pkginfo | egrep [jJ][rR][eE]\|[jJ][dD][kK]

This will produce output like this:

runtime/java/jre-8
1.8.0.181.12 and

system SUNWj8cfg
JDK 8.0 Host Config(1.8.0_202)

system SUNWj8dev
JDK 8.0 Dev. Tools (1.8.0_202)

system SUNWj8jmp
JDK 8.0 Man Pages: Japan (1.8.0_202)

system SUNWj8man
JDK 8.0 Man Pages (1.8.0_202)

system SUNWj8rt
JDK 8.0 64-bit Runtime Env. (1.8.0_202)

Solaris is a UNIX-based operating system, so the full _le system search method described for Ubuntu/ Debian Linux can also be used.

Desktop Machine Transition

Prior to the release of JDK 11, Oracle JDK contained a number of commercial features that are not part of the OpenJDK source code. Two of these specically apply to the deployment of applications on desktop systems.

1. The Browser Plugin: This is used to enable Applets to be used through a web browser. There is no open source alternative to this, and the Azul Zulu JDK does not include equivalent functionality. Most browser providers no longer support plugins, and Oracle ended support for the Browser Plugin in March 2019 (even for those with a commercial support contract). In these situations, the option is to continue using your existing JDK and accept the potential security risks of not being able to address known vulnerabilities.

2. Java Web Start: This deployment technology provides for applications to automatically update themselves when the user runs them. Although the Java Network Launch Protocol (JNLP), which is part of Web Start, has a JSR, no reference implementation was provided for this. Azul can provide builds of IcedTea-Web, which is an open-source alternative to Java Web Start. This is not a drop-in replacement, so some additional transition effort is needed – Azul has worked with other customers on this topic. IcedTeaWeb transition is beyond the scope of this document. Download at https://www.azul.com/ products/components/icedtea-web/